New Address: (202) 696-4318
Dr. Susan Chipman
Personnel & Training Research
Code 442PT
Office of Naval Research
800 N. Quincy Street
Arlington, VA 22217

The Seventh Annual Conference of The Cognitive Science Society

Irvine, California
1985



Program of the

Seventh Annual Conference of the Cognitive Science Society

15-17 August 1985
Irvine, California

The Seventh Annual Conference of The Cognitive Science Society

Richard H. Granger, Jr.
Program Chair

Kurt Eiselt
Assistant to the Program Chair

Jennifer Holbrook
Submissions Coordinator

Nomi Feldman
Conference Coordinator

Laura Yoklavich
Conference Assistant

Reviewers

Yigal Arens
William H. Batchelder
Myron L. Braunstein
Jaime G. Carbonell
Diana Deutsch
Michael G. Dyer
Arthur C. Graesser
Steven Hampson
Patrick J. Hayes
Graeme Hirst
Donald D. Hoffman
Keith J. Holyoak
Mary-Louise E. Kean
Dennis F. Kibler
Walter Kintsch
Janet L. Kolodner
Howard S. Kurtzman
David LaBerge
Patrick W. Langley
Michael Lebowitz
Wendy G. Lehnert
Edward H. Matthei
James L. McGaugh
James R. Meehan
Donald A. Norman
Terry Parsons
Michael I. Posner
Brian J. Reiser
Christopher K. Riesbeck
Paul S. Rosenbloom
Mallory Selfridge
David E. Smith
Michael K. Tanenhaus
Kurt VanLehn
Norman M. Weinberger
Kenneth N. Wexler
John I. Yellott, Jr.

Officers of the Cognitive Science Society

John Anderson	Carnegie-Mellon	(1983-1988)
Daniel G. Bobrow (Chair 84-85)	Xerox PARC	(1979-1987)
Gordon H. Bower	Stanford	(1980-1986)
Joan Bresnan	Stanford & Xerox PARC	(1983-1988)
Ann Brown	Illinois	(1984-1989)
Allan Collins	Bolt Beranek & Newman	(1979-1987)
Patrick J. Hayes	Schlumberger PA Resrch	(1984-1989)
Walter Kintsch (Past Chair 83-84)	Colorado	(1979-1985)
Ulric Neisser	Emory	(1985-1990)
Donald A. Norman	UC San Diego	(1979-1990)
Zenon Pylyshyn (Chair Elect 85-86)	Ontario	(1979-1985)
Edward E. Smith	Bolt Beranek & Newman	(1980-1986)

This conference was supported by grants from the Alfred P. Sloan Foundation and the Life Sciences Directorate and Psychological Sciences Division of the Office of Naval Research.

Thursday 15 August 1985

7:30am-8:45am Breakfast [Mesa Court Cafeteria]

8:45am-10:15am **Invited Address**
Shimon Ullman, Massachusetts Institute of Technology
[Fine Arts Village Theater]

10:15am-10:30am Break

10:30am-12:10pm **Paper Session I [Fine Arts Village Theater]**

10:30am-10:55am
Symmetry Detection and the Perceived Orientation of Simple Plane Polygons 1
Paul Kube, University of California, Berkeley

10:55am-11:20am
Variations on Parts and Wholes: Information Precedence vs. Global Precedence 11
Marc M. Sebrechts, Wesleyan University
John J. Fragala, Wesleyan University

11:20am-11:45am
The Neural Locus of Mental Image Generation: Converging Evidence from Brain-Damaged and Normal Subjects 19
Martha J. Farah, Carnegie-Mellon University

11:45am-12:10pm
A Developmental Neural Model of Word Perception 27
Richard M. Golden, Brown University

10:30am-12:10pm **Paper Session II [Concert Hall]**

10:30am-10:55am
A Computer Model of the Neural Substrates of Classical Conditioning in the Aplysia 36
Mark A. Gluck, Stanford University
Richard F. Thompson, Stanford University

10:55am-11:20am
Structural Learning in Connectionist Systems 43
Andrew G. Barto, University of Massachusetts, Amherst
Charles W. Anderson, University of Massachusetts, Amherst

11:20am-11:45am
The Learning of World Models by Connectionist Networks 54
Richard S. Sutton, GTE Laboratories
Brian Pinette, University of Massachusetts, Amherst

11:45am-12:10pm
Learning Salience Among Features Through Contingency in the CEL Framework 65
Richard H. Granger, University of California, Irvine
Jeffrey C. Schlimmer, University of California, Irvine

12:10pm-2:00pm Lunch Break [Mesa Court]

2:00pm-4:45pm **Invited Symposium I**
Attention in Early Visual Processing
[Fine Arts Village Theater]

Anne Treisman, University of British Columbia
Michael Posner, University of Oregon
David LaBerge, University of California, Irvine
Francis Crick, The Salk Institute
Daniel Kahneman, University of British Columbia
Shimon Ullman, Massachusetts Institute of Technology

4:45pm-7:00pm Dinner Break [Mesa Court]

7:00pm-11:00pm **Poster Session and Reception**
[University Club]
(List of poster presentations at end of program)

Friday 16 August 1985

8:45am-10:15am **Invited Address:**
Allen Newell, Carnegie-Mellon University
[Fine Arts Village Theater]

10:15am-10:30am Break

10:30am-12:10pm **Paper Session III [Fine Arts Village Theater]**

10:30am-11:20am
Bounded Irrationality: The Psychology of Incoherence
(paper not published in the proceedings)
Daniel Kahneman, University of British Columbia

11:20am-11:45am
Component Models of Physical Systems 80
Allan Collins, Bolt Beranek and Newman

11:45am-12:10pm
Temporal Notation and Causal Terminology 90
Yoav Shoham, Yale University
Thomas Dean, Yale University

10:30am-12:10pm **Paper Session IV [Concert Hall]**

10:30am-10:55am
Story Telling and Generalization 100
Michael Lebowitz, Columbia University

10:55am-11:20am
MULTIPAR: A Robust Entity-Oriented Parser 110

Jill Fain, Carnegie-Mellon University
Jaime G. Carbonell, Carnegie-Mellon University
Philip J. Hayes, Carnegie-Mellon University
Steven N. Minton, Carnegie-Mellon University

11:20am-11:45am
Towards a Computational Theory of Human Daydreaming 120
Erik T. Mueller, University of California, Los Angeles
Michael G. Dyer, University of California, Los Angeles

11:45am-12:10pm
Integrating Marker-Passing and Problem Solving 130
James A. Hendler, Brown University

12:10pm-2:00pm Lunch Break [Mesa Court]

2:00pm-4:45pm **Invited Symposium II**
Integrated Empirical Models of Learning and Memory
[Fine Arts Village Theater]

Paul Rosenbloom, Stanford University
Gary Lynch, University of California, Irvine
Pat Langley, University of California, Irvine
Jaime Carbonell, Carnegie-Mellon University
David Rumelhart, University of California, San Diego

4:45pm-5:30pm Reception [University Club]

5:30pm-7:30pm Banquet [University Club]

7:30pm-9:00pm **Invited Address**
Endel Tulving, University of Toronto
[University Club]

9:00pm-12:00mid Reception [University Club]

Saturday 17 August 1985

8:45am-10:15am **Invited Address:**
Roger Schank, Yale University
[Fine Arts Village Theater]

10:15am-10:30am Break

10:30am-12:10pm **Paper Session V [Fine Arts Village Theater]**

10:30am-10:55am
The Evolution of Knowledge Representations with Increasing Expertise in Using Systems 140
Dana S. Kay, Yale University
John B. Black, Teachers College, Columbia University

10:55am-11:20am

Purpose-Directed Analogy 150
Smadar Kedar-Cabelli, Rutgers University

11:20am-11:45am

Learning Concrete Strategies Through Interaction 160
R.W. Lawler, GTE Laboratories
Oliver G. Selfridge, GTE Laboratories

11:45am-12:10pm

Failure-Driven Acquisition of Figurative Phrases by Second Language Speakers 171
Uri Zernik, University of California, Los Angeles
Michael G. Dyer, University of California, Los Angeles

10:30am-12:10pm Paper Session VI [Concert Hall]

10:30am-10:55am

Two Kinds of Feature? A Test of Two Theories of Typicality Effects in Natural Language Categories 181
Robin A. Barr, Ball State University
Leslie J. Caplan, Ball State University

10:55am-11:20am

Empirical Evidence for a Global Workspace Theory of Voluntary Control 191
Bernard J. Baars, University of California, San Francisco

11:20am-11:45am

Connectionist Parsing 201
Garrison W. Cottrell, University of Rochester

11:45am-12:10pm

A Rule-Based Connectionist Parsing System 212
Bart Selman, University of Toronto
Graeme Hirst, University of Toronto

12:10pm-2:00pm Lunch Break [Mesa Court]

1:00pm-2:00pm Business Meeting of the Society [Mesa Court]

2:00pm-4:45pm Invited Symposium III
Syntactic Language Processing
[Fine Arts Village Theater]

Robert Berwick, Massachusetts Institute of Technology
Lyn Frazier, University of Massachusetts, Amherst
Howard Kurtzman, University of California, Irvine
Eric Wehrli, University of California, Los Angeles
Ken Wexler, University of California, Irvine

4:45pm-5:30pm Farewell Reception [Mesa Court]

Poster Presentations
7:00pm-11:00pm
Thursday, 15 August
[University Club]

Adaptive Planning: Refitting Old Plans to New Situations 222
Richard Alterman, University of California, Berkeley

Memory Representation and Retrieval for Editorial Comprehension 228
Sergio J. Alvarado, Michael G. Dyer, and Margot Flowers, University of California, Los Angeles

Analogy Recognition and Comprehension in Editorials 236
Stephanie E. August and Michael G. Dyer, University of California, Los Angeles

Investigations of Information Utilization during Fixations in Reading 242
Harry E. Blanchard, University of Illinois at Urbana-Champaign

Two Endorsement-based Approaches to Reasoning About Uncertainty 247
Paul R. Cohen, University of Massachusetts, Amherst

Teleology + Bugs = Explanations 256
Gregg C. Collins, Yale University

A Model for Understanding the Points of Stories 262
Marcy Dorfman, University of Illinois at Urbana-Champaign

A Framework for Concept Formation 267
J. Daniel Easterlin and Pat Langley, University of California, Irvine

The Problem of Existence 272
Kenneth D. Forbus, University of Illinois at Urbana-Champaign

Cross-Mapped Analogies: Pitting Systematiciy Aganist Spurious Similarity 277
Dedre Gentner, University of Illinois at Urbana-Champaign; Cecile Toupin, University of California, Berkeley

Information, Uncertainty, and the Utility of Categories 283
Mark A. Gluck, Stanford University; James E. Corter, Columbia University

A Model of Question Answering 288
Arthur C. Graesser, Memphis State University; George Vamos, University of Southern California; David Koizumi, C. Scott Elofson, California State University, Fullerton

The Time Course of Anaphora Resolution 293
Raymonde Guindon, Microelectronics and Computer Technology Corporation

Using a Computational Model of Language Acquisition to Address Questions in Linguistic Inquiry 298
Jane Hill, Smith College

A Model of Acquiring Problem Solving Expertise 303
Dennis Kibler and Rogers P. Hall, University of California, Irvine

Creating and Comprehending Arguments 308
Stuart M. McGuigan, Yale University; John B. Black, Teachers College, Columbia University

Levels of Goal Direction and the Causes of Learning 313
Dale M. McNulty, University of California, Irvine

Connectionist Learning in Real Time: Sutton-Barto Adaptive Element and Classical Conditioning of the Nictitating Membrane Response 318
J.W. Moore, J.E. Desmond, N.E. Berthier, D.E.J. Blazis, R.S. Sutton, A.G. Barto,
University of Massachusetts, Amherst

Explanation and Generalization Based Memory 323
Michael J. Pazzani, University of California, Los Angeles and The Aerospace Corporation

Bayesian Networks: A Model of Self-Activated Memory for Evidential Reasoning 329
Judea Pearl, University of California, Los Angeles

Expert Variance: Differences in Solving a Dynamic Engineering Problem 335
Michael Prietula and Frank Marchak, Dartmouth College

Machine Understanding and Data Abstraction in Searle's Chinese Room 341
William J. Rapaport, University of Buffalo

Toward a Unified Model of Deception 346
Donald D. Rose, University of California, Irvine

Building a Computer Model of Learning Classical Mechanics 351
Jude W. Shavlik and Gerald F. DeJong, University of Illinois at Urbana-Champaign

Persuasive Argumentation in Resolution of Collective Bargaining Impasses 356
Katia Sycara-Cyranski, Georgia Institute of Technology

The Interaction of Lexical Expectation and Pragmatics in Parsing Filler-Gap Constructions 361
Michael K. Tanenhaus and Laurie A. Stowe, University of Rochester; Greg Carlson,
Wayne State University

Predicting Conversational Reports of a Personal Event 366
Yvette J. Tenney, Bolt Beranek and Newman, Inc.

Thematic Knowledge, Episodic Memory and Analogy in MINSTREL, A Story Invention System 371
Scott R. Turner and Michael G. Dyer, University of California, Los Angeles

Spatial Inferences and Discourse Comprehension 376
Karl F. Wender and Monika Wagener, Technische Universitat Braunschweig

Cognitive Processing Strategies for Complex Addition 385
Keith F. Widaman, David C. Geary, Pierre Cormier, Todd D. Little, University of California,
Riverside

Author Index

Alterman, R. 222
Alvarado, S. 228
Anderson, C. 43
August, S. 236

Baars, B. 191
Barr, R. 181
Barto, A. 43, 318
Berthier, N. 318
Black, J. 140, 308
Blanchard, H. 242
Blazis, D. 318

Caplan, L. 181
Carbonell, J. 110
Carlson, G. 361
Cohen, P. 247
Collins, A. 80
Collins, G. 256
Cormier, P. 385
Corter, J. 283
Cottrell, G. 201

Dean, T. 90
DeJong, G. 351
Desmond, J. 318
Dorfman, M. 262
Dyer, M. 120, 171, 228, 236, 371

Easterlin, J. 267
Elofson, C. 288

Fain, J. 110
Farah, M. 19
Flowers, M. 228
Forbus, K. 272
Fragala, J. 11

Geary, D. 385
Gentner, D. 277
Gluck, M. 36, 283
Golden, R. 27
Graesser, A. 288
Granger, R. 65
Guindon, R. 293

Hall, R. 303
Hayes, P. 110
Hendler, J. 130
Hill, J. 298
Hirst, G. 212

Kay, D. 140
Kedar-Cabelli, S. 150
Kibler, D. 303
Koizumi, D. 288
Kube, P. 1

Langley, P. 267
Lawler, R. 160
Lebowitz, M. 100
Little, T. 385

Marchak, F. 335
McGuigan, S. 308
McNulty, D. 313
Minton, S. 110
Moore, J. 318
Mueller, E. 120

Pazzani, M. 323
Pearl, J. 329
Pinette, B. 54
Prietula, M. 335

Rapaport, W. 341
Rose, D. 346

Schlimmer, J. 65
Sebrechts, M. 11
Selfridge, O. 160
Selman, B. 212
Shavlik, J. 351
Shoham, Y. 90
Stowe, L. 361
Sutton, R. 54, 318
Sycara-Cyranski, K. 356

Tanenhaus, M. 361
Tenney, Y. 366
Thompson, R. 36
Toupin, C. 277
Turner, S. 371

Vamos, G. 288

Wagener, M. 376
Wender, K. 376
Widaman, K. 385

Zernik, U. 171

Symmetry Detection and the Perceived Orientation of Simple Plane Polygons

Paul Kube

Institute of Cognitive Studies
and the
Department of Electrical Engineering and Computer Sciences
University of California, Berkeley

1. Introduction

Recent advances in the computational theory of vision have been impressive. However, some simple and basic aspects of human visual performance have yet to be explained. For example, the details of human performance in seeing slanted isolated textureless plane polygons *as* slanted plane polygons are not predicted by any current theory. Below, five constraints on any correct theory in this domain are extracted from psychological experiment and contrasted with predictions of three of the best current computational theories; no theory meets them all. A detector model is proposed which can qualitatively account for the evidence. The model is one obtained by simple elaboration of a symmetry detection model introduced by Palmer [15] to account for a number of other perceptual phenomena in a unified way.

2. What experiment says

The study of the perception of the perspective projection of simple polygons dates from before the classic monograph of Stavrianos [16]. By 1966 the literature was fairly large (see, for example, Freeman's bibliography [10]), and continued to grow into the early seventies. Most studies in this tradition were restricted to stimuli perceptually indistinguishable from rectangles centered in the line of sight and rotated on an axis parallel to one side; i.e., stimuli which would project under perspective into the frontal plane as isosceles trapezoids. Data from the presentation of other kinds of contour stimuli is virtually absent. However, it appears that facts about the perception of even such simple objects are hard to account for in a simple theory.

Data from this body of work disagree on some points (concerning, for example, how tight the coupling is between a figure's perceived shape and perceived slant, or how the accuracy of slant judgments varies with stimulus size), but there are at least two unequivocal results worth noting. As summarized by Flock et al., "the single untextured rectangular shape when viewed monocularly without parallax constitutes too great a degree of impoverishment to elicit accurate slant judgments from the human visual system. . ." [9 p. 58] That is, human viewers aren't very good (mean regression no better than about 0.7 in the reported experiments) at seeing the precise slant of slanted rectangles when outline is the only cue. On the other hand, they are fairly likely to see slanted rectangles,or figures which project outlines indistinguishable from slanted rectangles, *as slanted*; see [5]. These simple facts impose two constraints (C1 and C2, below) which must be met by any candidate theory of human vision.

Two exceptions to the isosceles-trapezoid-stimulus tradition yield three additional constraints (C3, C4, and C5 below). An experiment reported by Attneave [1] featured the solicitation of slant judgments from subjects upon viewing frontal-plane parallelograms of various shapes and orientations. Rectangles at most orientations and aspect ratios showed great resistance to being seen as slanted in depth at all. Highly nonrectangular parallelograms were relatively easy to see as slanted figures. In a pilot

study at the Berkeley ICS, subjects are shown outline quadrilaterals like those in Figure 1.

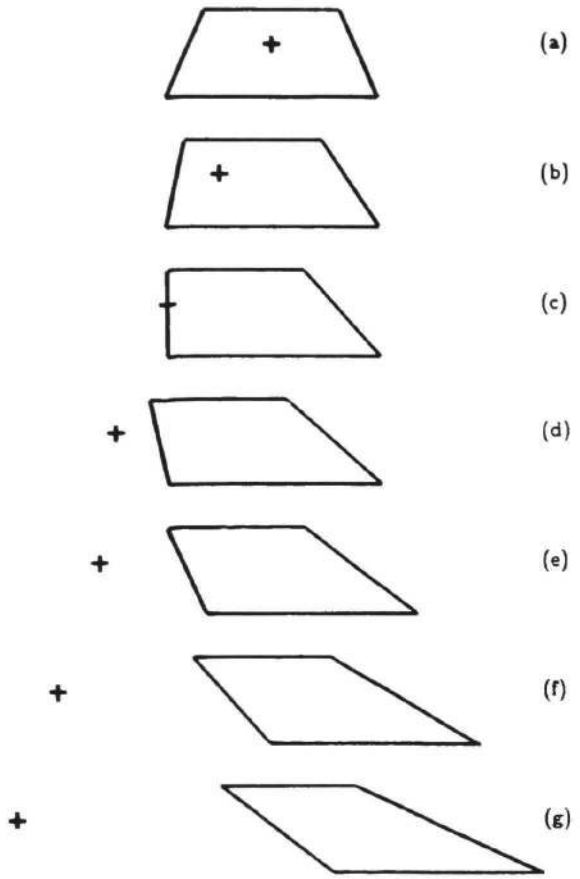


Figure 1, a-g

A series of seven perspective projections of a square in a plane with slant $\sigma = 60$ and tilt $\tau = 90$ degrees. The +'s are not part of the stimulus; they merely mark here the origin of the image plane for each projection. When a figure is foveated with the center of the lens above the origin, and such that the figure subtends fifteen degrees of arc vertically, the retinal image approximates viewing the outline of a meter-square tabletop, with near edge parallel to the image plane, at a distance of about two meters. (Of course, these reproductions need to be scaled before being used as experimental stimuli; at the size reproduced here, the focus is only about 4 cm. above the page.)

One of the figures at a time is monocularly presented to a subject so that the contour falling on the retina is the projection of a square slanted with respect to the frontal plane; the subject is asked to report whether or not she sees the figure as flat and slanted (i.e., lying in a plane not parallel to the frontal plane). Preliminary results are shown in Figure 2, with the vertical axis indicating increasing ease in seeing the figure as a slanted plane polygon. Some projections are considerably easier to see in this way than others, though all are equally *correct* projections of slanted squares.

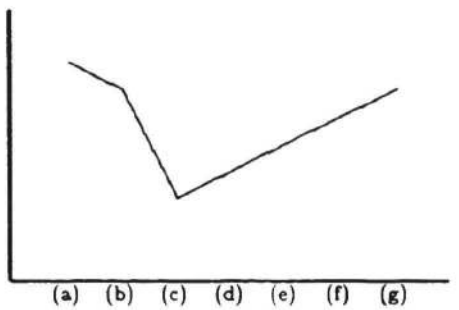


Figure 2

Relative "goodness of depth illusion" for the seven stimuli shown in Figure 1. Goodness of illusion increases upward along the vertical axis.

Thus, from experimental results, there are at least these five straightforward constraints that any theory of human visual capacity must meet:

C1. Viewers are fairly bad at judging the precise slant of slanted rectangles, even when they know they are viewing slanted rectangles.

C2. Figures which project isosceles trapezoids into the frontal plane can be easily seen as plane figures slanted in depth.

C3. Figures which project rectangles into the frontal plane are very hard to see as slanted in depth.

C4. Figures which project highly nonrectangular parallelograms into the frontal plane are easy to see as slanted plane figures.

C5. Of the actual perspective projections of a slanted rectangle at various orientations, some are more difficult to see as slanted plane figures than others.

3. What theory says

Following years of concentrated work in the development of polyhedral scene understanding systems (for example [11] [8] [14]) which had nothing to say about the perception of the orientation of isolated surfaces, recently there has been a flowering of interest in the construction of computational theories which demonstrate how, under plausible assumptions, depth information can be extracted from retinal images, and which sometimes do have consequences for the perception of slant of simple isolated polygons. To their credit, these theories typically aim for considerable generality and have these consequences, if they do have them, among many others of interest. The ones whose predictions will be examined below with respect to the experimental lessons of the previous section are all expressions of one sort or another of kind of principle of *Prägnanz*: they predict that an image will be interpreted as having been projected by the object which, of all possible projecting objects, best meets some criterion of simplicity.[1]

Several recent theories of this sort are not open for consideration because they are not even well-defined for polygonal stimuli. Witkin [19], for example, proposes that the chosen interpretation of a projection should be the one that maximizes the uniformity of distribution of orientations of contours in the scene; however, his mathematics require that the contour orientations be statistically independent, which is certainly not the case for simple polygons. Barrow and Tenenbaum [4] suggest that the contour which minimizes the integral of the square of its curvature over its length should be chosen as the interpretation; but curvature is either infinite or undefined at each vertex of a polygon, and zero everywhere else, so it fails to distinguish between any alternative projections. Barnard and Pentland [3] propose that the selected interpretation be one composed of circular arcs, and they offer an ellipse-fitting algorithm that achieves such an interpretation in some cases; but it fails on polygons. Three theories which do make predictions for the perception of simple polygons are considered below.

3.1. Kanade

A bilaterally symmetric figure is one whose shape is invariant over reflection about a line, the axis of bilateral symmetry. Consequently, such a figure is composed of pairs of points which lie at equal distances in opposite directions from the symmetry axis on lines that all meet the axis of symmetry at a 90 degree angle. If a figure is bilaterally symmetric, any affine transformation of the figure will be *skew-symmetric*: paired points in a skew-symmetric figure lie on lines (the *skew-transverse axes*) which all meet the skew-symmetry axis at some arbitrary angle, not necessarily 90 degrees. Since orthographic projection is an affine transformation, bilateral symmetries in the scene will become, under orthographic projection, skewed symmetries in the image. This fact is exploited by Kanade [12] [13], to derive constraints on perceived object orientation upon detection of image skewed symmetry, under the *Prägnanz*-like assumption that the perceptual system should prefer to see bilateral symmetries whenever possible.

To see how this constraint on orientation from skewed symmetry is supposed to work, consider the polygon in Figure 3 with its indicated skew symmetry. In this case, either dashed line can be taken as the skewed-symmetric axis, and the other as a skewed-transverse axis. The Kanade

[1] Or, at least, some criterion that is correlated with simplicity. Steve Palmer has suggested to me that, for example, the area-divided-by-squared-perimeter measure of Brady and Yuille (see below) should not be seen as a simplicity measure itself, but rather as a measure the maximization of which happens, in a range of cases, to pick out simple figures.

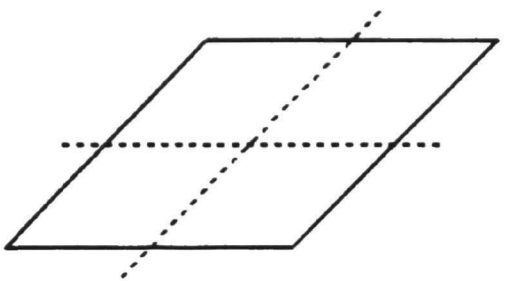


Figure 3

A skew-symmetric polygon. Dashed lines show axes of skewed symmetry.

assumption is that a skewed symmetry observed in an image was projected by a real symmetry in the imaged scene, and so the axes of skewed symmetry in the image are projections of the axes of bilateral symmetry in the scene. But axes of bilateral symmetry meet at right angles; so the assumption constrains the orientation of the plane containing the bilateral symmetry to be such that lines lying in it and meeting at right angles could have orthographically projected the observed axes of skewed symmetry in the image.

Kanade has shown that this is not sufficient to fix a unique interpretation for the orientation of the viewed figure; instead, the angle formed by the skewed-symmetric and skewed-transverse axes determines an orientation which must lie on a hyperbola (or, if the angle is 90 degrees and so the image is already bilaterally symmetric, a pair of perpendicular lines) in gradient space. (Here, gradient space is a two-dimensional space in which each point represents the orientation of a plane, and each of a set of parallel planes maps to the same point. The mapping is standardly defined as follows. Let there be three-dimensional cartesian coordinates in space such that the image plane (e.g., the 'plane' of the retina) satisfies $z=0$. Now an arbitrary plane in space will satisfy an equation of the form $pz+qy+d=z$, for some value of p, q, and d; and the equation of any plane parallel to this one will differ only in the value of d, so specifying p and q suffice to specify the orientation of the plane. The point (p,q) is then the map of the plane in gradient space. Alternatively, the arctangent of the length of the vector $<p,q>$ gives the slant σ of the plane, and the angle of the vector clockwise from the p-axis gives the tilt τ; that is, to talk of orientation in terms of slant and tilt is just to impose polar coordinates on gradient space.) In the present case, the skewed-symmetry axes shown in Figure 3 constrain the object plane to have an orientation lying on the hyperbola in Figure 4.

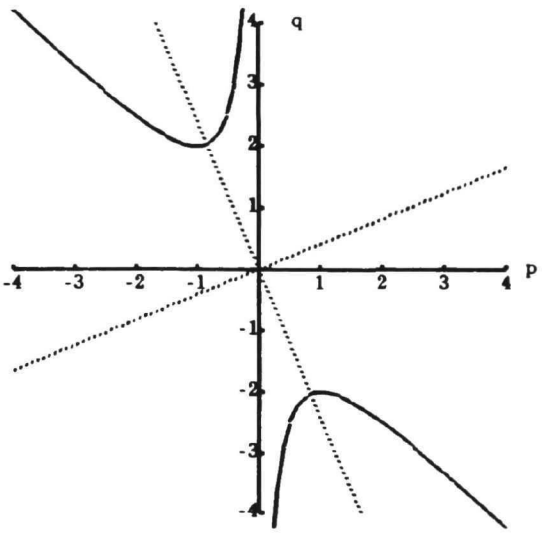


Figure 4

Gradient space constraints on the plane containing the skewed symmetry shown in Figure 3 (solid hyperbolic curve) and in Figure 5 (dotted orthogonal lines).

Thus Kanade would predict that because of the detection of the indicated skew symmetry, Figure 3 will be seen as a rectangle (a parallelogram contralateral bisectors meet at right angles) whose orientation lies on the hyperbola in Figure 4; and Kanade [12] suggests further that, in the absence of further constraints, the perceptual system should select the least slanted of the orientations that are possibile given a symmetry constraint, i. e., an orientation on one of the

apexes of the hyperbola: in this case, a slant of about 65 degrees in the direction of one or the other of the Necker-reversal tilts.

This result seems plausible, but there are inadequacies with the approach. Note that the parallelogram of Figure 3 has another pair of skewed symmetry axes, as shown by the dashed lines in Figure 5.

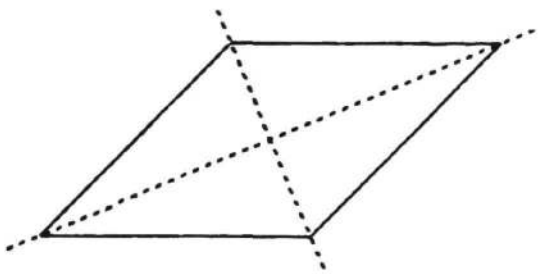


Figure 5

Another skewed symmetry for the polygon in Figure 3.

These happen to be axes of true bilateral symmetry, so they require that the plane of the projecting object have an orientation that falls on one of a pair of perpendicular lines in gradient space, which, for this example, are the broken lines of Figure 4. The theory does not explain how the constraints from these two skewed symmetries should be combined, and no reasonable combination seems to be consistent with all of the experimental constraints:

If minimization of slant is to count more heavily than reconstructing all possible symmetries, then the angle bisector axes of Figure 5 should be the ones that the Kanade constraints are meant to apply to; this gives the prediction of seeing the figure in the frontal plane, i.e., at zero slant. But this would violate constraint C4.

Perhaps, instead, the theory should be understood as predicting that all possible symmetries will be reconstructed, and that as a result the figure will be seen at an orientation that simultaneously satisfies the constraints imposed by both pairs of axes. This would correspond to the intersection, in the p,q plane, of the curves given by the two skewed symmetries; see Figure 4. In this example, this gives the same prediction as minimizing over the range of slants permitted by the contralateral bisector symmetry alone, which was a reasonable prediction.

But this suggestion fails on other examples. *Any* parallelogram in the image plane has two pairs of skewed symmetry axes, one pair connecting the midpoints of its sides and the other connecting its vertices. Now the skewed-symmetry assumption is that a skewed symmetry in the image was projected by a true bilateral symmetry in the scene; and so the skewed-symmetry axes must have been projected by perpendicular axes. But taking the intersection of the constraints imposed by both pairs of axes, this implies that the figure projecting a paralellogram must be a convex quadrilateral whose diagonals, and whose opposite-side bisectors, meet at right angles: i.e., the projecting figure must, in either case, be a *square*.

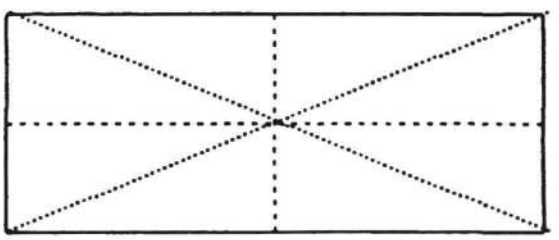


Figure 6

This rectangle, with its two skewed symmetries as shown, has the same slant constraints as the parallelogram in Figure 3

So, in avoiding a violation of C4, the Kanade assumption seems forced to predict seeing *every* nonsquare parallelogram as a square at some nonzero slant. This violates constraint C3, as can be seen by considering Figure 6, a figure

whose two pairs of skewed axes are those just those of Figure 3 rotated in the frontal plane. As a result, it has orientation constraints identical to those shown in Figure 4 up to a rotation in the p,q plane, and so the same slant constraints; but it is not naturally perceived as a slanted square in the manner of Figure 3, but rather as a rectangle in the frontal plane.

If instead (and, so far as I can see, without motivation from the theory) opposite-side bisector symmetries are taken as imposing the important orientation constraints, Figures 3 and 6 no longer provide counterexamples. However, the theory would still violate experimental constraint C2. Consider the projected isosceles trapezoid in Figure 1.1; it has only one pair of skew symmetry axes, which are also bilateral symmetry axes, and so the figure should be seen only in the frontal plane. There is nothing in Kanade's theory to account for its being easy and natural to see as a slanted rectangle.

3.2. Brady and Yuille

Brady and Yuille [6] have proposed that an image plane polygon be interpreted as having been projected by the object which, of all possible projecting objects, maximizes the ratio of area to perimeter squared. This measure tends to favor compact, nonelongated figures, and is maximized by squares within the class of quadrilaterals, so their theory entails the perception of a square whenever projectively possible. But, since their mathematics is developed under the assumption of orthographic projection, this (as for Kanade) leads to a violation of constraint C3 when applied to Figure 6. Also, it is easy to show that (under orthography) there is a family of image-plane isosceles trapezoids which are self-maximal over this measure; for these trapezoids, the preferred projecting figure is in the frontal plane, violating C2. Weiss, in a recent paper [18], proposes an improvement on the Brady and Yuille measure, but it also assumes orthography, and it violates C3.

Although the mathematics become more difficult, it is conceptually simple to consider Brady and Yuille measure maximization under perspective, instead of orthographic, projection; then it would predict that all image plane figures which are perspective projections of squares will be seen as squares. But this falls subject to the same criticism we suggest for Barnard's approach in the next section—that is, it would violate constraints C1 and C5.

3.3. Barnard

Barnard [2] suggests that angles in an image plane quadrilateral be interpreted as right angles whenever possible, and shows a way to obtain this interpretation under perspective projection. The problem of this approach, roughly, is that it works *too* well; his Figure 12a (reproduced here as Figure 7) is one that subjects find difficult to see as a slanted plane figure, whereas Barnard's method flawlessly extracts the projectively dictated interpretation of it as a slanted rectangle. The mathematics is impressive, but as a candidate psychological theory it violates both constraints C1 and C5.

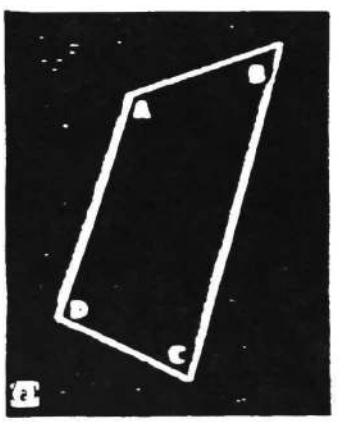


Figure 7

A perspective projection of a rectangle, from Barnard [2]

4. A proposed detector model

If the points made in the previous sections are correct, no single, simple *Prägnanz*-like account of goodness of interpretation of polygonal stimuli suggested in the literature can account for even a very modest range of results from psychological experiment. Some reasons why this might be so are discussed in the final section of the paper. In this section, a system of detectors is described which would exhibit performance consistent with constraints C1 - C5.

Note that C1 - C5 can reasonably claim to all be satisfied by a theory which at least qualitatively predicts the difficulty distribution (graphed in Figure 2) for the stimulus sequence of Figure 1, since this sequence contains both isosceles-trapezoidal (Figure 1.a) and (approximately) skew symmetric (Figure 1.g) stimuli,[2] as long as it is able to independently meet C3. This suggests a detector architecture in which at least some skewed symmetry and perspective gradients — the depth cues which seem to be presented by nonrectangular parallelograms and isosceles trapezoids, respectively — are detected independently and combined to yield a judgment of slant. Palmer [15] has argued that a wide range of phenomena in the psychology of human vision can be accounted for by appeal to a processing model which detects symmetries over members of the Euclidean similarity group exhibited in the stimulus array, certain relations among these symmetries, certain further relations among these relations, and so on. The model sketched here can be seen as an extension of Palmer's Euclidean symmetry detection model to incorporate higher-order detectors for some skewed symmetries and perspective gradients.[3]

[2] Figure 1.g is even closer to a true skew-symmetrical retinal stimulus than it may appear in the reproduction, since with the center of the lens at the focal point on the image plane z-axis and the figure foveated, the retina is slanted with respect to the image plane by about 45 degrees.

[3] Compare Clark, et al.'s [7] "retinal gradient of outline".

The present model supposes three layers of detectors. The lowest level is an array of first-order analyzers (in the sense of Palmer [15]), whose patterns of sensitivity to features of retinal stimulation can be related to one another by transformations from the Euclidean similarity group. For example, each can be taken as detecting, for some location of interest on the retina, the presence of line segments of a certain restricted range of lengths and orientations. (Obviously, such analyzers are interrelatable by translation, rotation, and dilation transformations.)

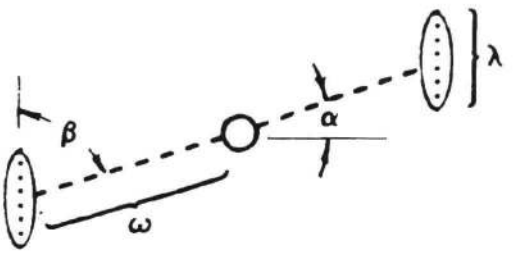


Figure 8

A translational symmetry detector element (TDE).

The second level consists of arrays of two kinds of detector elements, each of which is constructed by simple interconnections between two first-order analyzers. A translational symmetry detector element (TDE) is shown in Figure 8. Such a detector is parameterized by the retina-relative location of its center (represented by the open circle), the length and orientation (λ and β, respectively) of its constituent first-order analyzers (with receptive fields represented by the ellipses), and its own width and orientation (ω and α, respectively). A reflectional symmetry detector element (RDE), shown in Figure 9, is defined by the same set of parameters as a TDE, but its constituent first-order analyzers are related to each other differently: viz. by a reflection about a line through its center and perpendicular to its orientation, instead of a translation.

At the third level, TDE's and RDE's are connected to construct skewed symmetry and perspective gradient detectors — SSD's and PGD's, respectively. A

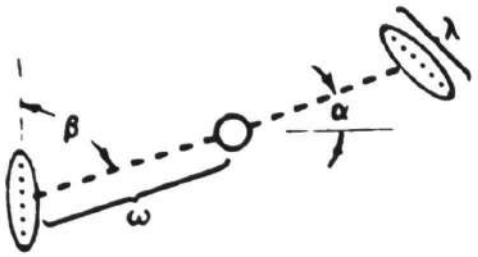


Figure 9

A reflectional symmetry detector element (RDE).

constant-width SSD is shown in Figure 10. Its constituent TDE's differ from each other only by a translation parallel to the favored direction of their constituent first-order analyzers. A bilaterally symmetric PGD, as depicted in Figure 11, is composed of RDE's which differ only by a composition of an appropriately related translation and dilation.

A detector d has associated with it a level of activation $A(d)$ which depends on the activations of its constituent detectors (or, in the case of first-order analyzers, directly on patterns of sensory stimulation). Let the activation of a TDE or RDE d be a suitable nonlinear function of the activation of its associated first-order analyzers f_1, f_2:

$$A(d) = \begin{cases} A(f_1)+A(f_2) & \text{if } A(f_1)>\theta \text{ and } A(f_2)>\theta \\ 0 & \text{if } A(f_1)<\theta \text{ and } A(f_2)<\theta \\ -\left|A(f_1)-A(f_2)\right| & \text{otherwise,} \end{cases}$$

where θ is a suitable threshold value. (What's important is that evidence for, evidence against, and the absence of evidence for or against the existence of a symmetry each be encodable in a second-level detector's activity.) Let the activation of a SSD or PGD be a monotonic function (say, the sum) of activations of its constituent detector elements.

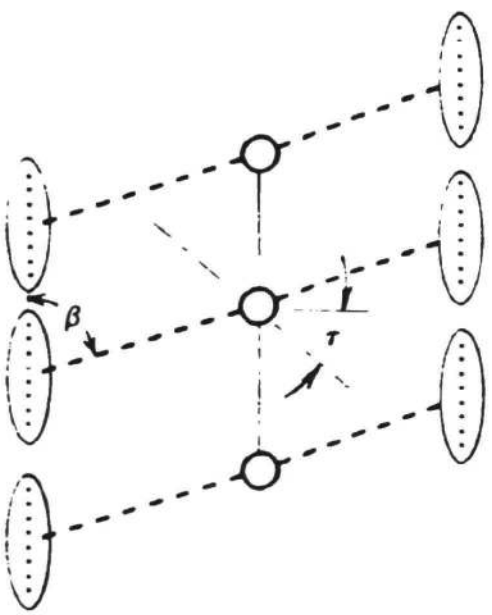


Figure 10

A constant-width skewed symmetry detector (SSD).

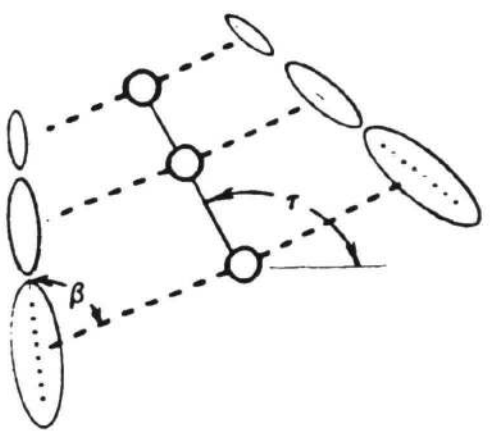


Figure 11

A bilaterally symmetric perspective gradient detector (PGD).

Now the patterns of activations of SSD's and PGD's can give rise to orientation judgments consistent with con-

straints C1 - C5 as follows. Suppose the SSD or PGD with the highest activation level, if it exceeds some threshold, is taken as encoding the viewed figure's most likely orientation in depth.[4] The tilt direction suggested by that detector is then just τ in Figures 10 or 11: the bisector of the angle between the SSD's axis and the favored direction of its constituent TDE's first order analyzers, or the axis of the PGD. Slant is a not such a simple function of detector parameters; though it is, under reasonable assumptions, monotonic increasing from zero as β in the Figures moves away from 90 degrees for both kinds of detectors. (This is suggestive of Stevens' [17] findings indicating that tilt judgments are more accurate than slant judgments.) This gives the desired results: C1 is explained, since slant is only imprecisely correlated with detector activation; C2 and C4 are satisfied, since PGD's and SSD's are suitably activated by isosceles trapezoids and nonrectangular parallelograms in the retinal image, respectively; C3 is met, since the most active detectors under stimulation by frontal plane rectangles will be a SSD or PGD with $\beta = 90$ degrees; and, since for projections intermediate between the shapes favored by the two types of detectors there may be no detector very activated, C5 is accounted for.

5. Discussion

We have extracted from the reports of psychological experiments five uncontroversial facts about human performance in the perception of simple polygons slanted in depth, and argued that no current theory of orientation from contour is consistent with all of them. We have proposed a detector model which would exhibit performance consistent with the constraints.

The model, however, is not a computational theory; it is a process model which in fact depends for its intelligibility on computational theory (without Kanade's theorems about skewed symmetry and facts about perspective projection, the correlation between β in Figures 10 and 11 and detected slant would be unexplained). It succeeds where the reported theories fail simply because it embodies an interaction between distinct kinds of evidence for orientation in depth. But this success says something about vision: the generation of orientation judgments of simple figures is not simple. The computational theories discussed in this paper are, individually, inadequate; but also the elementary two-factor model proposed here is grossly inadequate as a general account of vision, or even as an account of the monocular perception of simple plane figures. It fails on nonquadrilateral polygons and ellipses; it says nothing about orientation effects (rectangles, for example, present somewhat better depth illusions when their sides are not aligned with environmental horizontal and vertical); it's silent about how complexes of polygons might be perceived as three dimensional solids, and even about how an isolated polygon can be seen at different orientations in depth at different times. But it's unlikely that all the phenomena of monocular vision are going to be subsumed by a single computational principle. What's needed is a theory of what underlies each of them, and a theory of their interaction.

[4] Of course, the detectors need to be thickly distributed enough to respond to stimulus figures at various image locations, orientations, and scales.

Acknowledgements

The research reported here was supported by a grant from the Alfred P. Sloan Foundation to the Institute of Cognitive Studies at Berkeley. I want to thank Steve Palmer for comments on a draft of this paper, and for stimulating discussions.

REFERENCES

1. Attneave, F., Representation of Physical Space, in *Coding Processes in*

Human Memory, A. W. Melton and E. Martin (editor), V. H. Winston and Sons, Washington, D. C., 1972.

2. Barnard, S. T., Interpreting Perspective Images, *Artificial Intelligence 21* (1983), 435-462.
3. Barnard, S. T. and A. P. Pentland, Three-dimensional Shape from Line Drawings, *Proceedings, International Joint Conference on Artificial Intelligence*, 1983.
4. Barrow, H. G. and J. M. Tenenbaum, Interpreting Line Drawings as Three-Dimensional Surfaces, *Artificial Intelligence 17* (1981), 75-116.
5. Beck, J. and J. J. Gibson, The Relation of Apparent Shape to Apparent Slant in the Perception of Objects, *Journal of Experimental Psychology 50* (1955), 125-133.
6. Brady, M. and A. Yuille, An Extremum Principle for Shape From Contour, A.I. Memo No. 711, M.I.T., 1983.
7. Clark, W. C., A. H. Smith and A. Rabe, Retinal Gradient of Outline as a Stimulus for Slant, *Canadian Journal of Psychology 9* (1955), 247-253.
8. Clowes, M. B., On Seeing Things, *Artificial Intelligence 2* (1971), 79-116.
9. Flock, H. R., D. Graves, J. Tenney and B. Stephenson, Slant Judgments of Single Rectangles at a Slant, *Psychonomic Science 7*, 2 (1967), 57-58.
10. Freeman, R. B. J., Function of Cues in the Perceptual Learning of Visual Slant, *Psychological Monographs 80*, 2 (1966).
11. Guzman, A., Computer Recognition of Three-dimensional Objects in a Visual Scene, MAC-Tech. Rep.-59, M.I.T., 1968.
12. Kanade, T., Recovery of Three-dimensional Shape of an Object from a Single View, *Artificial Intelligence 17* (1981), 409-460.
13. Kanade, T. and J. R. Kender, Mapping Image Properties into Shape Constraints: Skewed Symmetry, Affine-Transformable Patterns, and the Shape-from-Texture Paradigm, in *Human and Machine Vision*, Academic Press, 1983.
14. Mackworth, A. K., Interpreting Pictures of Polyhedral Scenes, *Artificial Intelligence 4* (1973), 121-137.
15. Palmer, S. E., The Psychology of Perceptual Organization: A Transformational Approach, in *Human and Machine Vision*, Academic Press, 1983.
16. Stavrianos, B. K., The Relation of Shape Perception to Explicit Judgments of Inclination, *Archives of Psychology*, 1945.
17. Stevens, K. A., Surface Perception from Local Analysis of Texture and Contour, MIT-AI-Tech. Rep.-512, M.I.T., Cambridge, MA, 1980.
18. Weiss, I., 3-D Shape Representation by Contours, *Computer Graphics, Vision, and Image Processing*, to appear.
19. Witkin, A. P., Recovering Surface Shape and Orientation from Texture, *Artificial Intelligence 17* (1981), 17-45.

Variations on Parts and Wholes: Information Precedence vs. Global Precedence

MARC M. SEBRECHTS and JOHN J. FRAGALA

Department of Psychology
Wesleyan University
Middletown, CT 06457

Many current perceptual theories assume a two-stage model of perception in which there is a fast global analysis of a scene, followed by a slower description of constituent parts. This general view is referred to as "global precedence." This study reports experimental results that support an alternative model of "information precedence." Using pattern goodness as an operationalization of information, it is shown that speed of processing in a "same"-"different" task is affected by both global and local pattern information. This effect is partly strategic or attentional as demonstrated by a change in response pattern across three conditions: attention to global only, local only, or both global and local dimensions of a pattern. There are also perceptual effects as demonstrated by the effects of stimulus goodness within conditions. Good patterns were processed faster when they constituted the relevant dimension. When a dimension was irrelevant, good patterns slowed responding through stronger response competition. It is argued that any theory of perception must be able to account for the relative importance of these organizational factors. It is proposed that the probability of constraint satisfaction is one way to provide a processing description of these results.

Current models of visual perception frequently make assumptions about the order in which information is processed. These models are based on claims about the kinds of components that are necessary for object identification. In many cases, it is assumed that there are several stages of image formation, frequently specified by spatial frequency (see Marr, 1982). Thus, Lockhead (1972), for example, has argued that there are two stages in visual perception, an initial holistic or "blob processing", followed by a slower serial analysis.

Although there has been much controversy surrounding this issue, in many basic texts it has come to be assumed that there is an initial global analysis, followed by an identification of components. The most frequently cited evidence for this position, comes from the work of Navon (1977, 1981a, 1981b). In a series of studies he used a set of letters composed of smaller constitutent letters. He found that, in general, subjects had difficulty ignoring the global letter when asked to rapidly identify the local letter, whereas they were good at ignoring the component letters when making speeded identity responses about the global letter. Other studies have lent support to this view. Millspaugh (1978), for example, composed patterns out of a series of letters; the subjects' task was to indicate whether or not all of the constituent letters were the same. When the overall pattern in which these letters were embedded was highly organized, subjects had far more difficulty finding a disparate letter than when the overall pattern was less organized.

A number of studies have placed limits on the claims of global precedence. It has been shown that the effects of order of processing can be changed by manipulations of the visual angle (Kinchla and Wolfe, 1979) or the sparsity of the component stimuli (Martin, 1979). Others have argued that the precedence effects reflect attentional strategies (Miller, 1981; Boer & Keuss, 1982; Ward, 1982). These studies, however, argue for different ways of viewing the basic theory and generally accept the fact that global precedence is an accurate description of much of normal perception.

In contrast to these views, others have argued that there is no priority given to global aspects of a stimulus. Pomerantz and Sager (1975), for example, demonstrated local precedence in a card sorting task. Subjects were more influenced by local variation in patterns while trying to sort according to the global dimension than they were influenced by global variation when focusing on the local dimension. The explanation for this discrepancy is that Navon and others have failed to match baseline

discriminability. If the experimenter is free to choose the perceptual salience of the dimensions, it is argued, then either global or local aspects can be said to have precedence. Grice, Canham, and Boroughs (1983) used position as a way to experimentally manipulate perceptual salience. They presented patterns either in varying positions, as did Navon, or in a fixed central location. They found that when the position varied they replicated Navon's results. However, when the pattern was in a fixed location, there was no evidence of global precedence. Presumably, variation in location made it more difficult to perceive the constituent elements.

These studies again show defects in the experimental strategies used previously, without providing a conceptual basis for the alternative findings. Hoffman's (1980) studies suggested that precedence is perhaps due to the relative quality of information that is available at local and global dimensions. He selectively distorted the local and global stimulus properties, and then required subjects to determine if the stimuli matched a specified memory set (Sternberg, 1969). Using this technique, he demonstrated that it was possible to find either global or local precedence by distorting the other dimension.

The present study attempts to extend Hoffman's results by utilizing another paradigm and another set of stimuli. Hoffman used the Sternberg scanning task with a changing memory set. The items in the memory set can serve either as global or local targets. The present experiment used a sequential "same"-"different" task in which target items where themselves composed of both global and local elements; as a consequence, local stimuli and global stimuli were always present simultaneously. This makes it possible to examine the effects of goodness in a physical match. Evaluating differences in response time for a physical identity match is a fairly strong test of the power of the independent variables.

The stimulus set was also changed for this experiment. One of the important characteristics of "global" processing is that it has served as a means to explain many of the Gestalt phenomena, especially pattern goodness. If the global precedence hypothesis is correct, then the organizational properties of a stimulus may be made available before any of its elements. If, however, pattern goodness is a structural characteristic of stimuli that can be present at any level of a pattern (Sebrechts, 1980; Sebrechts & Garner, 1981), then pattern goodness should not be identified with "global" characteristics alone. In addition, although most of the research to date has utilized letter stimuli, there has been only limited study of their informational properties. In contrast, there is an extensive literature on the relation of perceived pattern goodness and information (Garner, 1962; 1976).

The following experiments suggest that perception is guided by the availability of information rather than by a strict, sequential hierarchical approach to scene analysis. On this view, perception is neither top-down nor bottom-up, in principle. Rather, it is defined by the informational context. To test this hypothesis, we presented subjects with patterns that varied in informational salience (goodness) at both a global and local level. If, in fact, the order of perceptual processing is mediated by information quality, then response time should vary as a function of pattern structure at the global and local levels. In addition, the control of perceptual consequences may be mediated by attentional mechanisms. To examine that possibility, subjects were presented with three instructional conditions: attend only to the global properties of the stimulus (the Global condition), attend only to the local properties of the stimulus (the Local condition), and attend to both (the Dual condition).

Method

Subjects

Twenty-four Wesleyan University students served as subjects in the experiment. All subjects were right-handed and had normal or corrected-to-normal vision. Subjects who had an error rate greater than 7.5% were eliminated from the study.

Stimuli

The stimuli consisted of global patterns made up of patterns that served as elements. The elements consisted of an array of 9 dots distributed in a 5 X 5 matrix with at least one dot in each row and each column. The global pattern was then constructed as an array of 9 of these elements distributed in a larger 5 X 5 matrix. The elements in any given global pattern were always the same. There were 8 patterns that could serve as elements and as global patterns. These consisted of 4 "good patterns" which were symmetrical about their vertical, horizontal, and diagonal axes, and 4 "poor patterns" that were created by rearranging the dots in one of the rows or columns in the matrix of the corresponding good pattern.

An example would therefore be a large "X" (global level) constructed of small diamonds (local level). In this case both the global and local levels

are "good". Likewise a globally good, locally poor stimulus was formed from a large "circle" formed from small aberrations of "X"s. In this way, a total of 64 stimuli can be constructed by combining factorially the 4 good and 4 poor arrangements on the global and local levels. This number was reduced to 56 by eliminating those patterns which have the same local and global arrangements (e.g., there are no global diamonds made up of local diamonds used in the study.) Examples of the four types of stimuli (Gg: Globally good, locally good; Gp: Globablly Good, locally poor; Pg: Globally Poor, locally good; and Pp: Globally Poor,locally poor) appear in Figure 1. Each of the 56 stimuli could be used as either the initial "target" or as the following "probe" on any given trial.

Apparatus

The stimuli were presented on the monitor of a Terak 8510 microcomputer. Subjects sat in a well-lit room, with their heads on a chin rest to ensure that they remained 14 inches from the screen throughout the experiment. At this distance, the overall pattern subtended a visual angle of 7 degrees, and each local element subtended a visual angle of approximately 1 degree. The subjects rested the index and middle fingers of the right hand on two keys of the Terak keyboard. Half of the subjects were told to use their index finger for "same", and their middle for "different", and the other half were given the opposite finger-response assignment. The microcomputer displayed the patterns and recorded the responses and reaction times. All stimuli appeared in the center of the screen.

Procedure

Each trial consisted of the following events. A fixation point appeared in the center of the screen for 500 milliseconds, accompanied by a warning tone. The screen was cleared and the first stimulus or "target" was displayed for a duration of 200 milliseconds. The screen was again cleared, and after a delay of 500 milliseconds (the inter-stimulus-interval or ISI), the second stimulus or "probe" was displayed. Upon striking a response key, the screen was cleared and the next trial followed after a delay of three seconds. If no response was made within four seconds, the trial was recorded as an error, the screen was cleared and the next trial began. Subject were instructed to respond as quickly as possible while avoiding errors.

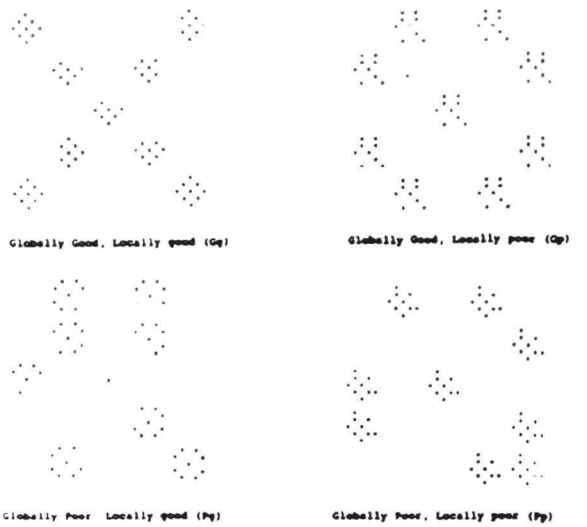


Figure 1 Examples of the four types of stimuli.

Conditions

Each subject participated in three experimental conditions which were defined by the dimension(s) which were relevant to making a response of "same" or "different": Global, Local, Dual. In the Global condition, subjects were instructed to attend only to the overall configuration. In the Local condition, subjects were told to base their decisions solely on comparison of the elements. Finally, in the Dual condition the two stimuli were to be judged "same" only when they were identical on both the global and local dimensions.

Experimental Design

A representative sample of approximately 448 pairings were selected from the pool of 3136 possible ordered pairings of the 56 stimuli. This sample was divided into two groups, with half of the subjects receiving 224 of the pairings and the other half receiving the remaining 224 pairings. There were an equal number of males and females receiving each of these sets of stimuli.

Each subject received a block of practice trials followed by seven blocks of randomized trials for each of the three conditions. Subjects were given a short rest at the end of each block of trials. The order of conditions and blocks was determined by a Latin Square design. Each subject received an equal number of "same" and "different" trials and an equal number of the four trial types (Gg, Gp, Pg, Pp) in each condition.

Results

Since the logic of this type of reaction time experiment assumes optimal performance, subjects with an error rate of greater than 7.5% were eliminated from the study. Mean error rate for the remaining subjects was 3.7%. There were too few errors for analysis of individual response categories. The following analyses are based only on median correct response time for each type of stimulus pairing for each subject.

"Same" Response: Identity Trials

Table 1 presents the response times for those trials in which the target is identical to the probe, both globally and locally. The global and local dimensions can be either good or poor, resulting in the four target-probe categories Gg-Gg, Gp-Gp, Pg-Pg and Pp-Pp for each of the three conditions. The same set of target and probe stimuli were used in all three conditions, and a "same" response was always required; the only difference between conditions for these trials was the level to which subjects were told to attend. For these identity trials, the goodness of the relevant dimension affected performance, whereas the goodness of the irrelevant dimension did not. In the Global condition, globally good pairs were faster than globally poor pairs (537 vs 563 msec; $F(1,23)=7.31$, $p=.01$) whereas there was no difference between locally good and locally poor pairs (550 vs 550 msec). In the Local condition, locally good pairs were faster than locally poor pairs (537 vs 593 msec; $F(1,23)=19.44$, $p<.01$) but no effect was obtained for goodness on the global dimension (565 vs 566 msec). In the Dual condition, both dimensions were relevant, and there were significant effects of both global ($F(1,23)=6.14$, $p=.02$) and local ($F(1,23)=16.89$, $p<.01$) pattern goodness. Response time was fastest when the global and local dimensions were good, slower when one of the dimensions was poor, and slowest for the Pp-Pp category.

Table 1
Mean Reaction Time (in msec) for "Same," Identity Trials

		Condition		
Target	Probe	Dual	Global	Local
Gg	Gg	558	544	538
Gp	Gp	591	530	591
Pg	Pg	581	556	537
Pp	Pp	604	569	596

"Same" Response: Non-Identical Trials

The irrelevant dimension of target and probe in the identity trials always matched physically. In this section, responses are analyzed for stimuli that match physically on the relevant dimension, thus requiring a "same" response, but that do not match physically on the irrelevant dimension. (This was true only for the Global and Local conditions, since any variation in the stimuli for the Dual condition required a "different" response.) In general, across these trials subjects responded faster when the relevant dimension was good. Thus for the Global condition, globally good stimuli were faster than globally poor ones (611 vs 663 msec, $t(23)=3.96$, $p<.01$), and for the Local condition, locally good stimuli were faster than locally poor ones (621 vs 681 msec; $t(23)=4.01$, $p<.01$).

In order to analyze the effect of the goodness of the irrelevant dimension, it is necessary to extract those patterns for which target and probe are matched for goodness on the irrelevant dimension. Table 2 contains the response times for those target-probe pairs which have an identical pattern on the relevant dimension and thus require a "same" response, but have different patterns with the same goodness (good or poor) on the irrelevant dimension and were therefore not identical. This subset of patterns again showed faster reaction time if the relevant dimension was good rather than poor. Thus, in the Global condition, globally good pairs were faster than globally poor pairs (594 vs 662 msec; $F(1,23)=12.88$, $p<.01$), and in the Local condition locally good pairs were faster than locally poor ones (626 vs 672 msec; $F(1,23)=8.66$, $p<.01$) The opposite effect was found for the irrelevant dimension; a good pattern on the irrelevant dimension slowed response time. In the Local condition, globally good pairs were slower than globally poor ones (668 vs 630 msec; $F(1,23)=6.27$, $p<.02$), and in the Global condition, locally good pairs were slower than locally poor pairs (650 vs 606 msec; $F(1,23)=6.11$, $p=.02$).

Table 2
Mean Reaction Time (in msec) for "Same," Non-Identity Trials

		Global		Local	
		Glob. Identity		Loc. Identity	
T	P	Loc. Same	Good	Glob. Same	Good
Gg	Gg	610		641	
Gp	Gp	578		695	
Pg	Pg	689		611	
Pp	Pp	634		648	

Note. T = target; P = probe.

"Different" Trials

Whenever the relevant dimension of target and probe were not identical, the correct response was "different." Tables 3 and 4 present the data for "different" trials in which goodness was matched for both global and local patterns of target and probe. These cases are most comparable to the "same" responses already analyzed.

In general, the RT differences within each condition were small, although there was substantial consistency in the pattern of responding. The data in Table 3 are for pairings which had different stimuli of the same goodness on a relevant dimension, and thus required a "different" response, although there was a physical identity match for one dimension of the target and probe. In these cases, responses were generally faster if the relevant dimension was good. The only effect that was statistically reliable at the subjects responded more quickly if target and probe were both locally good than if they were both locally poor ($F(1,23)=21.28$, $p<.01$). Responses also were generally faster when the dimension that matched between target and probe was poor, but these effects were again not significant.

In sum, "different" RTs were fairly uniform within conditions with few reliable effects. However, the absolute values of the RTs suggest that it may be easier to differentiate patterns when the relevant patterns are good rather than poor. In contrast, a "different" response may be slowed more by a physical match of good stimuli than by a match of poor stimuli. This parallels the results for the "same" trials, although the differences are much smaller.

Table 3
Mean Reaction Time (in msec) for "Different" Trials with Physical Identity on One Dimension

		Condition			
		Dual		Global	Local
T	P	Global Identity	Local Identity		
Gg	Gg	604	641	661	637
Gp	Gp	713	615	630	666
Pg	Pg	601	641	660	608
Pp	Pp	660	651	652	648

Note. T = target; P = probe.

Table 4 contains the results for stimuli in which neither global nor local patterns of target and probe match physically, although they are matched for goodness. For these pairings, the data are less consistent, and there are no significant differences between RT to good and RT to poor patterns.

Discussion

To the extent that reaction time is used as a measure of order of processing, the pattern of results reported here is not consistent with any simple view of global precedence. Rather, pattern perception seems to reflect a parallel analysis of global and local characteristics. The speed of response to either dimension depends on the strategic constraints (manipulated by the attention conditions here) and informational constraints (manipulated by changes in pattern goodness).

A strong version of global precedence claims that, for any object or set of objects, the visual system analyses the overall global properties first. The validity of that claim was tested here by determining whether or not changes in global and local goodness could affect the relative speed of response to a given pattern. In the limit case, on a strict precedence view, RT should take longer for local than for global aspects of a pattern. That prediction was violated by our data, as indicated by the results for the identity trials (Table 1). The stimulus pairs were the same in all three conditions, but speed of responding changed with the attentional requirements of the task. Thus a Gp pair was responded to faster when attention was directed to the global dimension, but a Pg pair resulted in a faster response in the local condition.

Table 4
Mean Reaction Time (in msec) for "Different" Trials with No Physical Identity on Either Dimension

		Condition		
Target	Probe	Dual	Global	Local
Gg	Gg	556	756	653
Gp	Gp	602	678	665
Pg	Pg	572	688	644
Pp	Pp	593	702	652

A somewhat less strict view describes visual precedence as a theory about relative interference. On that view, global properties interfere with a response to local properties, whereas local properties do not interfere with global processing. That description of precedence is also contradicted by the present results. When subjects are required to attend only to the global dimension, processing speed is affected by conflicting information on the local dimension. Likewise, information from the global dimension can interfere with that on the local dimension.

These results conflict with those presented by Navon (1977, 1981a, 1981b). It has been suggested that such differences can be attributed to the fact that in Navon's study there was no basis for determining an appropriate baseline condition. The global dimension in his study, it is argued, was more discriminable than the local dimension. As Pomerantz has noted (Pomerantz and Sager,1975; Pomerantz, 1983), a reaction time advantage can be shifted from global to local by changing the salience of dimensions. The same kind of effect has been shown through the use of changes in pattern size (Kinchla and Wolfe, 1979) and in pattern distortion (Hoffman, 1983). One response to this argument is that matching discriminabilities indirectly eliminates the global precedence effect. Thus the question remains, what should be given priority: stimulus discriminability, or global precedence.

In the present study, an alternative strategy was used to establish a baseline that avoids the need for an a priori solution to that question. Rather than trying to match the global and local dimensions, comparisons focused on the relative import of these dimensions across conditions. What was important, was not an absolute differentiation between times for global and local dimensions, but the way in which it was possible to manipulate the response on those dimensions by changing the attentional or informational constraints on the task.

The three conditions indicated that changes in attentional focus can influence speed of processing for a particular pattern, as demonstrated most strikingly in the physical identity pairings (Table 1). Although target and probe were idential and were separated by only a 500 msec interval, the pattern of responding changed among attentional conditions.

Results of the "same" trials indicate that matches of good patterns result in faster RTs than matches of poor patterns. This was true, however, only for the relevant dimensions. In the identity trials, the goodness of the irrelevant dimension did not reliably affect RT. In the non-identity "same" trials (Table 2), the goodness of the irrelevant dimension did influence RT; good patterns slowed responding. At first glance this may seem counterintuitive, since good patterns normally speed processing. However, in this case, the irrelevant dimension is providing evidence for a mismatch. Thus a good mismatch may provide stronger evidence for a "different" response and thus interferes with a corrrect "same" response.

In addition, there is interference both of global matches on local responses and of local matches on global responses. Thus, the quality of information can influence speed of processing at either level.

In contrast, there were few significant effects of goodness for the "different" trials, and the results are consistent with a deadline model in which a "different" response is made if a match is not found after a specified period. As noted in the results section, the absolute values of "different" RTs indicate that there may be some small effects of an irrelevant match, similar to those for "same" trials.

These results are generally consistent with a parallel-processing associative model like that proposed by Ratcliff (1978; 1981). Evidence is accumulated over time for both a match and a non-match of probe to target. A response depends on sufficient accumulated evidence to reach a criterion. In this experiment, it can be argued that global and local information are accumulated in parallel. An identical match in the Dual condition, for example, takes roughly the same amount of time as a match in the longer of the equivalent Global and Local conditions. This argues against serial analysis of the two stimulus dimensions.

RT differences within condition, however, indicate that the rate of accumulation of evidence is dependent on stimulus goodness or information quality. Interestingly, the goodness of stimuli that matched on the irrelevant dimension did not influence the "same" RT. It may be that any positive evidence is balanced by a greater difficulty with focusing of attention. Thus a good irrelevant match would add to the evidence for a match more rapidly than a poor irrelevant match; at the same time, insofar as good patterns are more effective in capturing attention, they would tend to make attentional allocation to the relevant dimension more difficult.

The major intereference effect that was observed appears to result from response competition. Good stimuli provide evidence more quickly; when they conflict with the match on the relevant dimension they inhibit a matching response.

Of course, the kinds of matching described here, are only one part of scene analysis. As Pomerantz (1983) has

noted, there are different ways to describe "global" stimuli, and positional location of elements is only one category of description. Nevertheless, the general problem of segregation and combination evidenced in our experimental stimuli is an essential component of more complex analyses. A classic case of a related problem is the blocks world that has been carefully examined in computer vision systems (Brady, 1981). This scene consists of a collection of objects which themselves may be organized into more complex objects. In that context, it could be asked, for example, if an arch is detected first and then decomposed into supports and bridge or if the supports and bridge are isolated and then used to constitute an arch. The present experiment argues that there is no single answer. It depends upon the characteristics of the components and the overall structure. If there are a number of irregular blocks that compose a regular arch, the arch will be perceived more readily. If, on the other hand, the arch itself is irregular (as in some of the architectural works of Gaudi) whereas the components are themselves more regular structures (good patterns), then the components will stand out first.

The processing explanation that may account for these data is a system that engages in parallel constraint satisfaction (Huffman, 1971; Clowes, 1971). Objects can be defined and segregated by selecting features. The features, however, can be grouped together in only certain ways, that is, there are constraints on the possible combinations of lines forming objects. If this is the kind of analysis that is performed in vision, however, then there is no strict hierarchy of perception. Rather, the source of constraint satisfaction can be thought of as the source specifying pattern information. If the global pattern provides constraint quickly, then perception can be thought of as top-down. However, if the constraints are generated by individual elements, then perception will appear bottom-up. Information precedence fits the data better than global precedence, and it also provides a framework for a reasonably parsimonious processing mechanism.

References

Boer, L.C. and Keuss, P.J.G. (1982). Global precedence as a post-perceptual effect: An analysis of speed-accuracy tradeoff functions. **Perception and Psychophysics, 31**, 358-366.

Brady, J.M. (Ed.) (1981). **Computer vision.** Amsterdam: North-Holland.

Clowes, M. (1971). On seeing things. **Artificial Intelligence. 2**(1).

Garner, W.R. (1962). **Uncertainty and structure as psychological concepts.** New York: John Wiley and Sons.

Garner, W.R. (1974). **The processing of information and structure.**, Potomac, Maryland, Lawrence Erlbaum.

Grice, G.R., Canham, L. and Boroughs, J.M. (1983). Forest before trees? It depends where you look. **Perception and Psychophysics, 33**, 121-128.

Hoffman, J.E. (1980). Interaction between global and local levels of a form. **Journal of Experimental Psychology: Human Perception and Performance, 6**, 222-234.

Huffman, D. (1971). Impossible objects as nonsense sentences. In B. Meltzer & D. Michie (Eds.) **Machine intelligence 6.** Edingurgh, Scotland: Edinburgh University Press.

Hughes, H.C., Layton, W.M. and Baird, J.C. and Lester, L.S. (1984). Global precedence in visual pattern recognition. **Perception and Psychophysics, 35**, 361-371.

Kinchla, R.A. and Wolfe, J.M. (1979). The order of visual processing: "top-down", "bottom-up" or "middle-out". **Perception and Psychophysics, 25**, 225-231.

Lockhead, G.R. (1972). Processing dimensional stimuli: A note. **Psychological Review, 79**, 410-419.

Marr, D. (1982). **Vision.** San Francisco: Freeman.

Martin, M. (1979). Local and global processing: The role of sparsity. **Memory and Cognition, 7**, 476-484.

Miller, J. (1981). Global precedence in attention and decision. **Journal of Experimental Psychology: Human Perception and Performance, 7**, 1161-1174.

Millspaugh, J.R. (1978). Effects of array organization on same-different judgements. **Perception and Psychophysics, 23**, 27-35.

Navon, D. (1977). Forest before trees: The precedence of global features in visual perception. **Cognitive Psychology, 9**, 353-383.

Navon, D. (1981a). The forest revisited: More on global precedence. **Psychological Research, 43**, 1-32.

Navon, D. (1981b). Do attention and decision follow perception? Comment on Miller. **Journal of Experimental Psychology: Human and Performance, 7,** 1175-1182.

Pomerantz, J.R. (1983). Global and local precedence: Selective attention in form and motion perception. **Journal of Experimental Psychology: General, 112,** 516-540.

Pomerantz, J.R. and Sager, L.C. (1975). Asymmetric integrality with dimensions of visual pattern. **Perception and Psychophysics, 18,** 460-466.

Pomerantz, J.R., Sager, L.C. and Stoever, R.J. (1977). Perception of wholes and of their component parts: Some configural superiorty effects. **Journal of Experimental Psychology: Human Perception and Performance, 3,** 422-435.

Ratcliff, R. (1978). A theory of memory retrieval. **Psychological Review, 85,** 59-108.

Ratcliff, R. (1981). Parallel-processing mechanisms and processing of organized information in human memory. In G.E. Hinton and J.A. Anderson (Eds.), **Parallel models of associative memory.** Hillsdale, NJ: Lawrence Erlbaum Associates.

Sebrechts, M.M. (1980). **On the relation between goodness and similarity: Stimulus-specific and context-specific properties.** Unpublished doctoral dissertation, Yale University, New Haven.

Sebrechts, M.M. and Garner, W.R. (1981). Stimulus-specific processing consequences of pattern goodness. **Memory and Cognition, 9,** 41-49.

Ward, L.M. (1982). Determinants of attention to local and global features of visual forms. **Journal of Experimental Psychology: Human Perception and Performance, 8,** 562-581.

The neural locus of mental image generation: Converging evidence from brain-damaged and normal subjects

Martha J. Farah
Carnegie-Mellon University

Abstract

Recent work with brain-damaged patients has provided evidence for a tentative neuroanatomical localization of mental image generation in the posterior left hemisphere. This evidence will be briefly summarized and critiqued, and a new test of the localization, using normal subjects, will be presented. When mental images of stimuli were used as templates to facilitate a visual discrimination, the effect of imagery was greater for stimuli presented in the right visual field (left hemisphere) than in the left visual field (right hemisphere). This result is discussed in relation to earlier claims about the hemisphericity of imagery.

Introduction

The ability to imagine the appearances of objects and scenes not currently in view requires more than simply having those appearances stored in long-term memory. It also requires the ability to generate, or reconstruct, from those long-term memory descriptions the short-term, array-format mental image (Kosslyn, 1980). Recent evidence from the study of two different neurological patient groups has suggested that these procedures may be critically dependent upon structures in the left hemisphere of the brain.

In this paper a third source of evidence is presented for the left-laterality of mental image generation, from a lateralized tachistoscopic stimulus presentation technique used with normal subjects. The objective of the experiment described here is to expand both quantitatively and qualitatively the evidence for the laterality of image generation; whereas each method of localization has its own weaknesses, one can decrease the liklihood of spurious or artifactual findings by using multiple different methods.

Previous findings

Evidence for a functional localization of imate generation in the left hemisphere came initially from Farah's (1984) review of published neurological case reports of imagery deficits. In this review, task analyses were constructed for the cognitive and perceptual tests that had been administered to the patients described in the case reports. By comparing the cognitive components that occurred in successfully and unsuccessfully performed tasks, it was possible to infer the particular component of mental imagery ability that was impaired. There were twelve patients who could recognize visually presented objects, yet could not imagine the appearances of the same or similar objects. To explain the imagery deficit in these patients we must postulate damage to some component of the imagery system that is not shared with visual recognition; that is, if visual recognition is intact then the long-term visual memories must be intact and so must the short-term visual memory medium in which both images and percepts occur (Farah, 1985; Finke, 1980). By a process of elimination, this leaves the image generation process, which converts the long-term memory information

into a mental image in short-term visual memory, as the impaired component of imagery ability. In these twelve cases, the predominant site of brain damage was the posterior left hemisphere.

The laterality of image generation inferred from the effects of focal brain damage on imagery ability is supported by recent studies of two split-brain patients (Farah, Gazzaniga, Holtzman & Kosslyn, 1985; Kosslyn, Holtzman, Farah & Gazzaniga, in press). Whereas the disconnected left hemispheres of these patients were able to perform tasks requiring image generation, the right hemispheres were generally not, despite adequate performance on control tasks containing all of the processing steps of the imagery task except for image generation per se. For example, in each patient both hemispheres could classify a lower case letter as including or not including a long "stem" (e.g. "t" and "g" have long stems, "s" and "r" do not) and both hemispheres could use an upper case cue to select the associated lower case form from a set of lower case letters presented in free vision. However, the performances of the left and right hemispheres diverged sharply in the corresponding imagery task: only the left hemispheres of the split-brain patients could perform the lower case letter classification using the upper case form as a cue.

On the one hand, these preliminary results from patients with localized brain damage and with surgically separated but otherwise intact hemispheres represent a significant convergence of evidence for a left hemisphere locus for image generation. On the other hand, several facts warrant caution in our acceptance of this localization.

In the analysis of cases from the neurological literature, one might argue that the procedure of case selection by literature review has a systematic bias against right-hemisphere-damaged patients: such patients are initially less likely to come to the attention of neurologists and psychologists (because their language ability will not have been endangered) and this bias may be compounded by the tendency of right-hemisphere-damaged patients to deny the existence of neurological deficits, including those far more obvious than a loss of imagery, such as hemiplegia or cortical blindness (Hecaen & Albert, 1978).

In the case of the split-brain patients, interpretation of right-hemisphere failures on the imagery tasks is complicated by the lack of independent evidence that the right hemisphere, with its inferior logical and language comprehension abilities, understood the task. One of the "Catch 22's" of split-brain research is that there is in general no way to determine that the right hemisphere has understood a task that it has failed.

A new test: A cognitive paradigm with normal subjects

The purpose of the present experiment is to assess the laterality of image generation in a third way, with normal subjects in an imagery paradigm developed and validated within cognitive science. A finding of left-hemisphere superiority in this experiment cannot be attributed to any of the potential artifacts outlined above, and would therefore strengthen the hypothesis that image generation is lateralized to the left hemisphere.

The basic experimental paradigm is a lateralized visual discrimination task, in which the subject is to decide whether a stimulus, presented briefly and to one side of a fixation point, is or is not a pre-designated "target." Past studies have shown that a visual image of the target can be used as a template to facilitate discriminations between target and

non-target stimuli (Cooper & Shepard, 1973, Posner, Boies, Eichelman & Taylor, 1969).

The two targets in this experiment are a plus sign and a rectangular-shaped capital "O" character. Non-targets are characters selected for being visually similar to either the plus or the "O": "*", "-", an oval-shaped "O" and "@". In this experiment, subjects perform two versions of the lateralized discrimination task. In the "Baseline" condition, they are pre-cued with the information about the side on which the stimulus will occur before the stimulus presentation. Their task is to respond "target" to either of the targets and "non-target" to nay of the non-targets. In the "Imagery" condition, they are pre-cued as before with the side on which the stimulus will occur, and they are also shown one of the two targets (in central vision), which they are instructed to image in the position of the up-coming stimulus. The task is the same as in the Baseline condition: Subjects respond "target" to either target, whether or not it is the same as their image, and "non-target" to all non-targets.

Subjects were given two blocks of 64 trials each of the Baseline condition, followed by two more blocks of the imagery condition. Subjects were right-handed males with no first-degree relatives known to be left-handed. As an objective criterion for determining whether or not subjects were o,agomg as omstricted. the mean latency of response to stimuli while holding similar and different images were compared for each subject. If a subject was successful at using an image as a template during the Imagery condition, then they should have responded more quickly to stimuli that were similar to the image than to stimuli that were different. Accordingly, any subject who showed an opposite trend would be eliminated from the analysis and replaced with a new subject. Data from twenty subjects is reported here. An additional six subjects were tested and replaced because they showed a negative effect of similarity (nonsignificant in all cases). Three of these subjects also estimated having imaged less than 75% of the time. All other subjects reported complying with the instructions at least 80% of the time.

The image should facilitate the visual discrimination between targets and non-targets, particularly when the image and stimulus are visually similar. Thus, there are two, independent measures of the effectiveness of imagery in the present experiment: the degree of facilitation with an image, relative to performance in the same visual discrimination task without an image, and the degree of facilitation with an image that is visually similar to the stimulus being presented, relative to performance in the same visual discrimination task with an image that is visually different from the stimulus being presented. If the left hemisphere is specialized for image generation, then these measures of image-mediated facilitation should be greatest when the image-stimulus overlap occurs in the right hemifield-left hemisphere, where the image was generated.

Results

Response latencies from incorrect responses were removed from the analysis, as were latencies that were more than 1.75 times the mean of the remaining response latencies from the same subject for comparable responses (i.e. responses from the same Order x Laterality x Response cell in the Baseline condition, and from the same Order x Similarity x Laterality x Response in the Imagery condition). Mean response latencies were then calculated for each subject in each subcondition of the experiment (i.e. each Imagery x Order x Similarity x Laterality x Response condition) and these means comprised the data upon which the statistical analyses reported below were performed.

The mean response times and error rates in the conditions of interest are shown in Table 1. The two predictions set out earlier for the differential effects of imagery on visual discrimination performance in the two hemifields were borne out by planned comparisons among the response latencies in this task. The first prediction concerned the overall effect of image generation on visual discrimination performance (i.e. the difference between performance in the Baseline and Imagery conditions). As predicted, there was a right visual field superiority only in the Imagery condition, $t = 2.81$, $df = 19$, $p < .01$. The mean response latencies to stimuli in the left and right hemifields in the Baseline condition were 818 msec and 815 msec respectively, and in the Imagery condition they were 776 msec and 754 msec respectively. Note that the small field difference in the Baseline response latencies was not by itself significant, $t < 1$.

The second prediction concerned the effect of image-stimulus similarity in the Imagery condition (i.e. the difference between performance when the image and stimulus were visually similar and when they were visually different, in the Imagery condition). Also as predicted, there was a greater right visual field advantage when the image and the stimulus were visually similar than when they were visually different, $t = 3.19$, $df = 19$, $p < .005$. The response latencies to stimuli in the left and right hemifields after generating a different image were 797 msec and 790 msec respectively, and after generating a similar image they were 755 msec and 717 msec respectively.

The possibility that hemispheric differences in speed-accuracy trade-offs could account for the present findings was dispelled by an examination of each hemisphere's error rates in the experimental conditions of interest, shown in table 1. For the first of the contrasts above, the left hemisphere error rate was the same for the baseline and imagery conditions (20.5%), and the right hemisphere error rate was only 0.3 percentage points higher in the baseline condition than in the imagery condition (23.4% and 23.1% respectively). For the second of the contrasts above, the hemispheric error rates as well as response latencies reflected the predicted pattern of performance, with greater facilitation from similar than from different images in the left hemisphere (18.5% and 22.5%, respectively) than in the right hemisphere (24.4% and 21.8% respectively).

The response latencies were also submitted to a repeated measures analysis of variance, whose factors were presence or absence of imagery, laterality of stimulus presentation, response, similarity (of image to stimulus in the Imagery condition and the identical partitioning of trials in the Baseline condition) and order of occurence (i.e. first block or second block of the Baseline and Imagery conditions).

The overall interaction between laterality of stimulus presentation and presence or absence of imagery, a specific form of which was tested with the first planned comparison above, was of borderline significance, $F(1,19) = 3.00$, $p = 1$. The overall three-way interaction between presence or absence of imagery, laterality, and similarity, a specific form of which was tested with the second planned comparison above, was significant, $F(1,19) = 5.07$, $p < .05$.

Order of occurence had a significant effect, $F(1,19) = 9.09$, $p < .01$, and an examination of the means shows this to be primarily a speeding-up between the first baseline block and the second: 850 msec mean latency for the first baseline block, 782 msec for the second baseline block, 780 msec for the first imagery block and 750 msec for the second imagery block. The disproportionate speeding-up between the first and second baseline blocks, compared with the first and second imagery blocks, is reflected in a significant interaction between order or occurence and presence or absence of imagery, $F(1,19) = 4.88$, $p < .05$.

However, there was no interaction between order of occurence and laterality, $F(1,19) = 0.08$, $p = .79$, and thus the laterality effect present in the imagery blocks and not in the baseline blocks cannot be accounted for as an artifact of order of occurence.

Similarity and its interactionm with presence or absence of imagery were highly significant, although this was at least partly artifactual, given the subject selection procedure described earlier of excluding subjects whose response times were slower when the image and stimulus were similar than when they were different. As expected, similarity had an effect only in the imagery condition, $F(1,19) = 9.21$, $p < .01$ for the interaction of similarity and presence or absence of imagery: With a similar image, a stimulus could be classified as a target or nontarget in an average of 736 msec, whereas with a different image an average of 794 msec were required; when the same two groups of trials, administered in the same order, were performed without imagery, the comparable mean latencies were 818 msec and 814 msec respectively. The effect of similarity in the imagery condition led to a significant main effect of similarity in this experiment, $F(1,19) = 20.06$, $p < .001$. Targets were responded to more quickly than nontargets, $F(1,19) = 14.54$, $p < .005$, with an average of 750 msec required for targets and 831 msec required for nontargets. The three-way interaction between presence or absence of imagery, similarity, and response was significant, $F(1,19) = 26.09$, $p < .001$, and an examination of the means in table 1 indicates that this results from the disproportionate facilitation with imagery for similar targets in the imagery condition, which, in contrast to both similar nontargets and different targets and nontargets, constitute a perfect, template-style match with the image. The two-way interaction between similarity and response is also significant, $F(1,19) = 10.71$, $p < .01$, owing to the just-discussed effect of template-style matches in the trials of the imagery condition.

Finally, there were two four-way interactions of statistical significance: A presence or absence of imagery by similarity by order of occurence by response interaction, $F(1,19) = 9.40$, $p < .01$, which the means of table 1 suggest reflects a progressive quickening in responding "no" to similar nontargets and "yes" to different targets in the imagery condition (i.e. with practice subjects became less inclined to false alarm to a nontarget that resembled their image and less inclined to miss a target that looked different from their image), and a presence or absence of imagery by order by laterality by response interaction, $F(1,19) = 14.21$, $p < .01$, for which no simple interpretation suggests itself. No other effects were significant, $p > .1$.

Discussion

Visual discrimination performance was more strongly facilitated by the presence of an image in the right visual field (left hemisphere), and was more sensitive to the visual similarity between the image and the stimulus in the right visual field (left hemisphere). Both of these results were predicted by the hypothesis that the left hemisphere is required for image generation, because according to this hypothesis, image-stimulus interactions after a left visual field stimulus presentation require either the image or the percept to cross the corpus callosum, with attendent delay and degradation of information, whereas callosal transmission is not necessarily required after a right visual field stimulus presentation.

How do the present findings relate to other recent work on hemispheric differences in visual/spatial processing? Many authors have suggested that imagery is a right hemisphere function, although Ehrlichman and Barrett's (1983) survey of the neuropsychological literature found little empirical support for this hypothesis. As Ehrlichman and Barrett pointed out,

most claims of right hemisphere specialization for imagery derive from a conflation of imagery with other forms of visual/spatial ability. When one considers the computational "problem" which must be solved in image generation, it is clearly of a different nature from the problems of maze learning, visual analogy solving, face recognition, and other typical examples of visual/spatial tasks for whicxh right hemisphere superiority has been found. In particular, the work of Kosslyn and his associates has shown that image generation involves the construction of the image from separately stored parts, even in the case of a seemingly unitary image, such as an image of a dog or the letter "a". (Farah & Kosslyn, 1981; Kosslyn, 1980; Kosslyn, Reiser, Farah & Fleigel, 1983; Kosslyn & Shwartz, 1978). Image generation therefore involves the integration of separate parts in functionally spatial array, an ability which has been identified with areas in the posterior left hemisphere (Kinsbourne & Warrington, 1962; Levine & Calvanio, 1978).

To summarize the outcome of the present experiment, the effect of image generation on visual discrimination performance was found to be asymmetrical in a direction consistent with a left hemisphere locus for image generation. Taken together with the results described earlier from neurological patients, the present result strengthens the hypothesis that image generation depends upon structures in the left hemisphere.

References

Cooper, L.A. & Shepard, R.N. Chronometric studies of the rotation of mental images. In W.G.Chase (Ed.), *Visual Information Processing*. New York: Academic Press, 1973.

Ehrlichman, H. & Barrett, J. Right hemispheric specialization for mental imagery: A review of the evidence. *Brain & Cognition*, 1983, *2*, 39-52.

Farah, M.J. The neurological basis of mental imagery: A componential analysis. *Cognition*, 1984, *18*, 245-272.

Farah, M.J. Psychophysical evidence for a shared representational medium for mental images and percepts. *Journal of Experimental Psychology: General*, 1985, *114*, 91-103.

Farah, M.J., Gazzaniga, M.S., Holtzman, J.D., & Kosslyn, S.M. A left hemisphere basis for visual mental imagery? *Neuropsychologia*, 1985, *23*, 115-118.

Farah, M.J. & Kosslyn, S.M. Structure and strategy in image generation. *Cognitive Science*, 1981, *4*, 371-383.

Finke, R.A. Levels of equivalence in imagery and perception. *Psychological Review*, 1980, *87*, 113-132.

Kinsbourne, M. & Warrington, E.K. A disorder of simultaneous form perception. *Brain*, 1962, *85*, 461-486.

Kosslyn, S.M. *Image and Mind*. Cambridge: Harvard University Press, 1980.

Kosslyn, S.M. Holtzman, J.D., Farah, M.J. & Gazzaniga, M.S. A computational analysis of mental image generation: Dissociations in split-brain patients *Journal of Experimental Psychology: General*, in press.

Kosslyn, S.M., Reiser, B.J., Farah, M.J., & Fleigel, S.L. Generating visual images: Units and Relations. *Journal of Experimental Psychology: General*. 1983, *12*, 278-303.

Hecaen, H. & Albert, M.L. *Human Neuropsychology*. New York: John Wiley & Sons, 1978.

Levine, D.N. & Calvanio, R. A study of the deficit in verbal alexia-simultanagnosia. *Brain*, 1978, *101*, 65-81.

Posner, M.I., Boies, S.J., Eichelman, W.H. & Taylor, R.L. Retention of visual and name codes of single letters. *Journal of Experimental Psychology*, 1969, *79*, 1-16.

Acknowledgements

The research reported here was supported by PHS fellowship F32 MH08876-01 and by the MIT Center for Cognitive Science under a grant from the Alfred P. Sloan Foundation. The author thanks Kaveri Suryanarayan and Ron Wilson for valuable technical assistance, and Howard Gardner for helpful comments on an earlier draft of this paper.

	BASELINE CONDITION		IMAGERY CONDITION			
			Image & stimulus similar		Image & stimulus different	
Response:	YES	NO	YES	NO	YES	NO
RVF (lh)	**777 ms** **20.6%**	**852 ms** **21.9%**	**646 ms** **13.8%**	**788 ms** **23.1%**	**802 ms** **24.7%**	**778 ms** **20.3%**
	821 ms 733 ms 20.6% 20.6%	881 ms 824 ms 22.2% 21.6%	651 ms 642 ms 16.9% 10.6%	831 ms 746 ms 28.1% 18.1%	809 ms 794 ms 24.4% 25.0%	786 ms 770 ms 18.1% 22.5%
VISUAL FIELD (hemisphere)						
LVF (rh)	**775 ms** **20.8%**	**860 ms** **26.8%**	**683 ms** **18.2%**	**827 ms** **30.6%**	**763 ms** **21.6%**	**832 ms** **21.9%**
	797 ms 753 ms 23.4% 18.1%	902 ms 818 ms 27.5% 24.4%	704 ms 663 ms 16.9% 19.4%	831 ms 823 ms 33.1% 28.1%	798 ms 728 ms 23.8% 19.4%	831 ms 833 ms 25.6% 18.1%

A developmental neural model of word perception

Richard M. Golden

Department of Psychology
Brown University
Providence, RI 02912

The Interactive Activation model (McClelland & Rumelhart, 1981; Rumelhart & McClelland, 1982) has been successfully applied to a broad range of phenomena in the "letter-within-word" perception literature. A unique aspect of the Interactive Activation (IA) model is that all processing is based upon very simple local computations similar in spirit to the types of computations that might be performed by neurons. These simple local computations however, give rise to interesting global behaviors at the network level. The IA model operates by attempting to satisfy a great many local constraints between and within a set of letter and word "nodes." These local constraints, nevertheless, are explicitly given to the IA model. How might such constraints evolve over time if learning were incorporated into the IA model?

In this paper, a specific member of the class of neural models known as Brain-State-in-a-Box (BSB) models (Anderson, 1983; Anderson, Silverstein, Ritz, & Jones, 1977) is suggested as a useful approach for considering the development of visual letter within word perception. Interestingly enough, recent theoretical results (Golden, 1985; Hopfield, 1984) indicate that the dynamic behavior of the BSB and IA models share important qualitative similarities. The BSB formalism, however, is comparatively simpler than the IA formalism, makes interesting reaction time predictions, and provides a formal framework for considering how the effects of experience create and organize letter and word representations. More specifically, using both reaction time and letter recognition accuracy as dependent variables, the BSB model suggests how the effects of experience influence the development of the ability to use information about orthographic redundancy (Juola, Schadler, Chabot, & McCaughey, 1978; Lefton & Spragins, 1974) and case type (McClelland, 1976; Pollatsek, Well, & Schindler, 1975).

Description of the neural model

The testing dynamics of the model. The Brain-State-in-a-Box model is based upon a few neurophysiological assumptions. The first assumption is that essential information about the environment is assumed to be coded by a set of neuronal firing frequencies (Anderson, 1983; Anderson et al., 1977). If there are M neurons in the system, the momentary activation pattern across the neuronal set is characterized by an M-dimensional state vector in which the ith element of the state vector represents the firing frequency of the ith neuron in the system minus the spontaneous firing frequency of that neuron. The magnitude of the state vector represents the current signal strength while the direction of the state vector indicates the identity of the activation pattern. The second assumption states that, in general, the current firing rate of a neuron may be approximately represented by the linear combination of the firing rates of the other neurons in the system and a set of "synaptic connectivity coefficients." The connectivity coefficients are an attempt to model the degree of synaptic efficacy between pairs of neurons within the system. Using matrix notation, these coefficients are arranged in a matrix such that the ijth element of the matrix represents the connection strength between the ith and jth neurons in the system. The state vector at discrete time slice t + 1 may now be rewritten as the state vector

at discrete time slice t plus the state vector at discrete time slice t multiplied by the matrix. The third assumption of the model is that each neuron possesses a maximum and minimum firing rate. This final assumption introduces an essential non-linearity into the previous linear system and gives the model an exceptionally rich range of behavior. This assumption is also the motivation behind the model's nickname since it essentially confines the M-dimensional state vector within the space of an M-dimensional box or hypercube (for additional details see Anderson et al., 1977).

The dynamics of the system are relatively straightforward. An initial pattern of neural activity is amplified using positive feedback until all neurons within the system have obtained their maximum or minimum firing rates. More formally, one cycle through the system may be written as:

$$\mathbf{S}(i + 1) = \text{TRUNC}[\mathbf{AS}(i) + \mathbf{S}(i)] = \text{TRUNC}[(\mathbf{A} + \mathbf{I})\mathbf{S}(i)] \qquad (1)$$

where **I** is the identity matrix, the notation **S**(i) indicates the activity vector after the ith feedback cycle, **A** is the synaptic connectivity matrix, and the TRUNC function sets all vector elements whose magnitudes are above some maximum firing frequency equal to that maximum firing frequency and all vector elements whose magnitudes are below some minimum firing frequency equal to that minimum firing frequency.

The initial state vector **S**(0) is presented to the system by applying equation (1) to **S**(0) to generate **S**(1). The state vector **S**(1) is then applied to equation (1) to generate **S**(2). These iterations continue until **S**(i) = **S**(i + 1). At this point, all the elements of the system state vector are firing at their minimum or maximum firing rates. Since in this situation the state vector has reached one of the hypercube corners, we will call this state vector a corner vector. If the state vector arrives at the "correct" hypercube corner, then the stimulus is assumed to have been properly categorized. The number of iterations required to arrive at a hypercube corner is taken as the system's reaction time.

The training algorithm. In typical simulations of the model, we assume that the period over which learning occurs is extremely long, relative to the period over which the model is tested. Therefore, for simplicity, learning is not permitted when the model is tested. The learning assumption implemented in this model is based upon a proposal by Hebb (1949) that states if two neurons within a neural network simultaneously fire, then a change in the nervous system occurs such that if one of the two neurons fires at a future date the probability that the other neuron will fire tends to increase.

During the training phase, a stimulus and response vector pair are randomly selected from the stimulus set. The stimulus vector is then perturbed with random noise and passed through (1) several times. The transformed stimulus vector and the corresponding response vector are then used to modify the matrix. Using linear algebra, the learning assumption is described by the following equation:

$$\mathbf{A}_{new} = \mathbf{A}_{old} + \gamma[\mathbf{g} - \mathbf{S}(K)][\mathbf{g} - \mathbf{S}(K)]^T \qquad (2)$$

where **S**(K) is the stimulus vector after K iterations through equation (1), $\mathbf{A}_{new}$ is the updated synaptic connectivity matrix, $\mathbf{A}_{old}$ is the original matrix, **g** is the desired response of the system, and γ is a scalar between zero and one. For the simulations reported here, the value of K remained constant and was always equal to seven.

Equation (2) therefore describes how the synaptic efficacy between individual

neurons within the system evolves over time as stimulus and response vectors are presented to the model. The exact form of (2) is not critical. Any learning rule that biases the coefficients of the connectivity matrix such that the stimuli within the training set become eigenvectors (associated with large positive eigenvalues) of the matrix will suffice (Golden, 1985).

Neural encoding of the stimuli. The assignment of "neural activation patterns" to specific symbols is as important to the formulation of the letter within word model as the basic BSB mechanism itself. A unique 28-dimensional state vector was assigned to each of the upper-case and lower-case forms of the nine most frequent letters of the English alphabet using a letter feature encoding scheme. A stimulus representing a letter string could then be represented by concatenating four 28-dimensional letter subvectors. Thus, four-letter words, pseudowords, and nonwords were represented by 112-dimensional vectors.

Theory

Although the proposed model superficially seems rather homogeneous, a great deal of structure exists in the synaptic connectivity matrix after learning has occurred. This specific internal structure is due to two factors. First, the system state vector is a list of position-specific letter features. And second, the learning algorithm effectively extracts frequently appearing pair-wise feature correlations from the stimuli learned by the model. Therefore, the matrix contains two distinct types of synaptic weights or pair-wise letter feature correlations. One set of synaptic weights are referred to as the *within-letter* feature correlations. The second set of weights are referred to as *between-letter* feature correlations. The within-letter feature correlations in the matrix correspond to the system's knowledge of the spatially redundant information in words. The between-letter feature correlations correspond to the system's knowledge of the transgraphemic information in words.

By definition, a nonword is a state vector that has not been "learned" by the system. Such a vector can nevertheless be categorized by the BSB model since the within-letter feature correlations can independently amplify the familiar letter subvectors representing the nonword stimulus, despite interference from the between-letter feature correlations. When a word or pseudoword is presented to the system, both the between-letter and within-letter feature correlations cooperatively amplify the system state vector. Words, however, tend to be recognized faster and more accurately than pseudowords since fewer between-letter feature correlations contribute to the amplification process during pseudoword recognition. Also note that, within the framework of this model, the superiority of letter recognition for same-case relative to mixed-case stimuli is exactly analogous to the word-pseudoword advantage.

Consider now the major effects characterizing the developmental behavior of the model. First, as the system's experience with words increases, letters are recognized faster and more accurately within words, pseudowords, and nonwords. And second, the ability to use information about the orthographic regularities within words develops very quickly. The first effect is a direct consequence of the number of times a given pair of letter features was presented to the system during the learning trials. The rapid acquisition of the ability to detect orthographic information occurs because useful letter feature correlations, obtained from only a few words, are used to categorize many other words possessing those feature correlations.

The model also makes a prediction regarding the development of alternating case effects. As experience with words increases, the advantage of letters within

same-case stimuli, relative to mixed-case stimuli, should increase at a fast rate in the initial stages of development and more slowly in the later stages. A theory suggesting that the effects of alternating case are located at the "level of single letter discriminability" (Adams, 1979, p. 154; also see McClelland & Rumelhart, 1981), might not make these predictions.

Computer simulation results

In the first set of experiments, the synaptic coefficients were initialized to zero and then letter stimuli were "taught" to the system with (2). After 1500 presentations of letter stimuli, the system was tested using a set of test stimuli. The test stimulus set consisted of 1176 different mixed-case and same-case words, pseudowords, and nonwords. The reaction time of the system for correctly categorizing each of the initial state vectors representing letter strings was then recorded. The above testing procedure was then repeated after the system had experienced 200 more presentations of word stimuli. Finally, the training of the system upon word stimuli was continued for an additional 800 learning presentations and again the testing procedure was repeated.

The simulation results are summarized in Figure 1. The reaction time of the model for recognizing four letter strings, like human subjects, tended to decrease with age and experience (Juola et al., 1978). In addition, the qualitative effects of a rapid acquisition of orthographic knowledge that becomes increasingly fine-tuned over a relatively longer period of time is also observed (Juola et al., 1978). In addition, as letter strings become more orthographically regular, letters in same-case stimuli are recognized faster than letters in mixed-case stimuli. These latter reaction time results have also been observed in the experimental literature (Pollatsek et al., 1975; Taylor, Miller, & Juola, 1977).

Figure 2 summarizes the results of a similar sequence of simulations where letter recognition errors were used as the dependent measure. In these latter simulations the model made frequent identification errors because of the addition of interfering factors (a mask and additive noise) in the testing procedure. Again, the basic qualitative effects observed in the human experimental literature were also observed in the simulations. Words were recognized more efficiently than pseudowords, which were recognized more efficiently than nonwords, and same-case stimuli were recognized more efficiently than mixed-case stimuli. The simulations also demonstrate a case-type by orthography interaction. Such an effect, although in agreement with reaction time studies of this phenomena and accuracy data obtained by McClelland (1976), was not observed by Adams (1979). The rapid acquisition of orthographic knowledge by the model has also been observed using decision tasks involving human subjects (Lefton & Spragins, 1974; Rosinski & Wheeler, 1972).

Summary

A developmental version of the Interactive Activation model has been proposed based upon a neural network model suggested originally by Anderson et al. (1977). The developmental BSB model offers a formal theory that motivates the use and connection of letter and word nodes in the IA model. To explicitly illustrate these statements, some simulations of the BSB model were then studied. The results of the computer simulations were compatible with the experimental literature. Effects of orthography and case type were observed to increase in magnitude as the system's experience with word-like stimuli was extended.

Acknowledgements

This research was supported in part by a grant from the National Science Foundation to J. A. Anderson, administered by the Memory and Cognitive Processes section (Grant BNS-82-14728).

References

Adams, M. J. (1979). Models of word recognition. *Cognitive Psychology, 11,* 133-176.

Anderson, J. A. (1983). Cognitive and psychological computation with neural models. *IEEE transactions on systems, man, and cybernetics, 5,* 799-815.

Anderson, J. A., Silverstein, J. W., Ritz, S. A., & Jones, R. S. (1977). Distinctive features, categorical perception, and probability learning: Some applications of a neural model. *Psychological Review, 84,* 413-451.

Gibson, E. J. (1969). *Principles of perceptual learning and development.* New York: Meredith Corporation.

Golden, R. M. (1985). *The "Brain-State-in-a-Box" neural model is a gradient descent algorithm.* Manuscript submitted for publication.

Hebb, D. O. (1949). *The organization of behavior.* New York: John Wiley & Sons, pp. 62-66.

Hopfield, J. J. (1984). Neurons with graded response have collective properties like those of two-state neurons. *Proceedings of the National Academy of Sciences, USA, 81,* 3088-3092.

Juola, J.F., Schadler, M., Chabot, R. J., & McCaughey, M. W. (1978). The development of visual information processing skills related to reading. *Journal of Experimental Child Psychology, 25,* 459-476.

Lefton, L. A., & Spragins, A. B. (1974). Orthographic structure and reading experience affect the transfer from iconic to short-term memory. *Journal of Experimental Psychology, 103,* 775-781.

Mason, M. (1975). Reading ability and letter search time: Effects of orthographic structure defined by single-letter positional frequency. *Journal of Experimental Psychology: General, 104,* 146-166.

McClelland, J. L. (1976). Preliminary letter identification in the perception of words and nonwords. *Journal of Experimental Psychology: Human Perception and Performance, 2,* 80-91.

McClelland, J. L., & Rumelhart, D. E. (1981). An interactive activation model of context effects in letter perception: Part 1. An account of basic findings. *Psychological Review, 88,* 375-497.

Pollatsek, A., Well, A. D., & Schindler, R. M. (1975). Familiarity affects visual processing of words. *Journal of Experimental Psychology: Human Perception and Performance, 1,* 328-338.

Rosinski, R. R. and Wheeler, K. E. (1972). Children's use of orthographic structure in

word discrimination. *Psychonomic Science, 26*, 97-98.

Rumelhart, D. E., & McClelland, J. L. (1982). An interactive activation model of context effects in letter perception: Part 2. The context enhancement effect and some tests and extensions of the model. *Psychological Review, 89*, 60-94.

Taylor, G. A., Miller, T. J., & Juola, J. F. (1977). Isolating visual units in the perception of words and nonwords. *Perception and Psychophysics, 21*, 377-386.

Appendix 1

Vector Encodings of Letter Stimuli

The following table indicates the assignment of specific letter subvectors to letters. The assignment of a vector coding to a letter was based upon an extension of Gibson's (1969, p. 88) abstract letter feature set. For example, the first eight elements of the letter subvector representing E are given by (+1,+1,+1,+1,-1,-1,-1,-1). For convenience, letter subvectors are described using hexadecimal notation by treating negative vector elements as zeros and positive vector elements as ones. Thus, the above eight-dimensional component of the letter subvector specifying E is represented as F0 using hexadecimal (base 16) notation. The letter subvector encodings using hexadecimal notation are provided below.

E	F003F3F	e	C0CF030
T	F00333F	t	F03F0FF
A	CF033CF	a	00C3030
O	00C030F	o	00C0300
N	33000CF	n	30300C0
R	33C30CF	r	303F0F0
I	30003CF	i	30003F0
S	000CC0F	s	000CC00
H	F0033CF	h	30300CF
X	0F0330F		

Appendix 2

Letter string stimuli selection

Word, pseudoword, and nonword letter strings were used as word vectors in the following experiments. The word stimuli were selected based upon moderate frequency of occurrence in the English language, and were constructed using only the nine most frequent letters in the English alphabet. The word stimuli were then scrambled, and the scrambled letter strings ranked using a spatial redundancy (i.e., using the likelihood that a particular letter would occur at a given spatial position within a word) table obtained by analyzing the original set of word stimuli (see Mason, 1975, for additional details).

Word Stimuli (ordered row-wise by decreasing frequency): THAT THIS INTO THAN THEN HERE AREA SEEN RATE SOON NEAR EAST SORT REST HEAR HAIR SENT NOTE TEST ONES SHOT NONE RISE HEAT THIN ROSE NINE TONE RAIN ARTS SITE SETS NOSE ONTO TREE SEAT HERO REAR ASIA HANS IRON ANNE EASE HATE RARE EARS OHIO HOST SEES HORN ROOT SONS TONS NOON STAR TORN HITS TIRE NEAT RENT NEST TENT TOES THEE EARN HERS SINS HIRE TIES TORE HATS NEON SHOE ROAR TROT ROSS TEAR SEAS SORE HINT HOOT HOSE IONS THOR TOSS TRIO SANE ANNA ANTS HEIR OATS RENO RIOT STIR TART OATH SITS TEEN

Pseudoword Stimuli(ordered row-wise by decreasing spatial redundancy): TENE TERE TETS TORS SENE TOSE TEOS TEIS SESE SONT SEST TETN RETS TASE TENR SOST THNE TONR TEAN TESN TOSN SOET SARE TEHE SETN TOEN NETS TISE TESR REAT ROES TOOR OENS AETS HONR SOSN TOER THOS NORT SOTR NEES NOOS SAES ROIT NOES RETN HAST TERA SOHE RAAE NOOT SONO HESR EONT SOSR EORS NEET TNOE SAET TROE AEES NOET HEOR TATR REAN HAET TNOS AOST HEER SOER REON SIST NARE EORT SOHT TRIE TRAT RAST HIST HOTO TEAH TEHN SIET SETA TOSH TNET SNAE HTNE RAET HERA HETA TOOH TAHT TAER SROE HASN SESA SERH

Nonword Stimuli (ordered row-wise by decreasing spatial redundancy): IRES AHSN OHNR TSRA OISN OTSS EASR SRTA SNTA OTAS ESSE ITSE IHTN INOT TROI OAHT SNEO ERET EOSH ORIT ESAE NSET HTRO HREO HTNA HSTO TSRI RAIH ESST ERTN STRA ANAN EISR ESAT HTTA EHSR ISST HTNI EHOR RTTO OHTO ANEN ERAN RNEO EHER ONEN ISET HSEO ERON STRI EATH ONNO OTNR ETTN IHER ESTN INEN ARER OTSN ENNO ATTR AISA HTSI EHRA EHTA ORTO OSSN IRNO OSTR AHRI OTEN ASEN EIRH ISSN IRTO ENNA OTNO OSNO ARNI ORRA EHRI ENRA ERRA ENTA OTER ITNO OSER RTOI OHOI NROI ITER ONTI ETRA ESTA OIHO ESSA OTAH ASAI

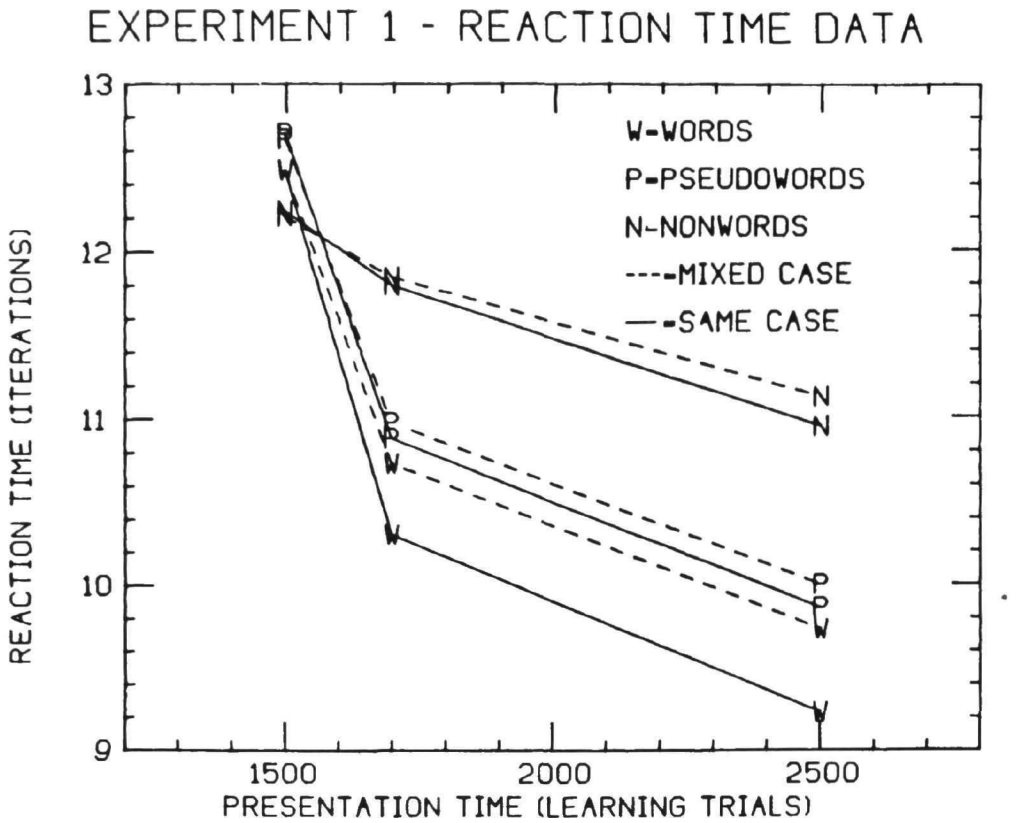


Figure 1. Reaction time plotted as a function of case type, orthography, and learning trials for Experiment 1. Solid lines indicate same-case stimuli. Dashed lines indicate mixed-case stimuli.

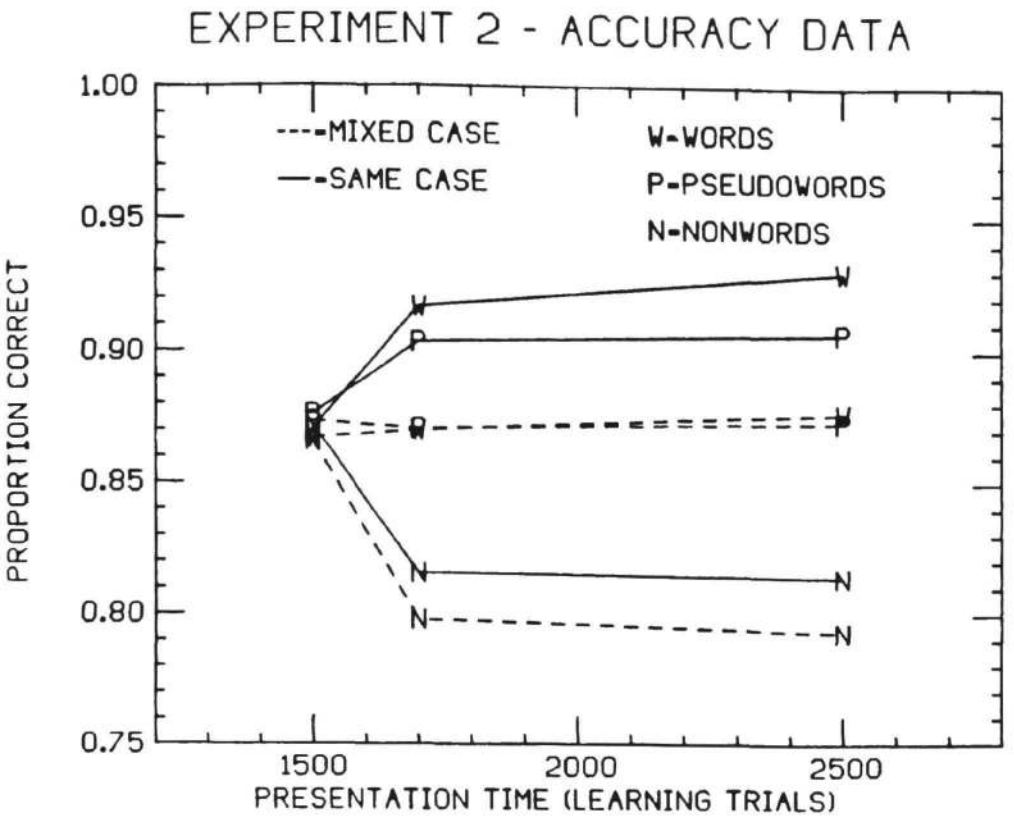


Figure 2. The proportion of correctly recognized letters plotted as a function of case type, orthography, and learning trials for Experiment 2. Solid lines indicate same-case stimuli. Dashed lines indicate mixed-case stimuli.

A Computer Model of the Neural Substrates of Classical Conditioning in the *Aplysia*

Mark A. Gluck & Richard F. Thompson
Stanford University

"In experiments extending over the past thirty years, I have been trying to trace conditioned reflex paths through the brain or to find the locus of specific memory traces" -- Karl Lashley

When the essential neural circuit of a memory trace has been defined in sufficient detail as a biological system, it becomes necessary to determine if the circuit will in fact generate the phenomena of learning and memory that it is presumed to model. Even in elementary circuits, it is not always evident what the outcome of a given set of stimulus and training conditions will be at a qualitative-logical level of analysis. We report here an initial attempt at such modeling, utilizing the general approach of associative network modeling from cognitive science. We utilize the circuit of the *Aplysia* that exhibits elementary associative learning as identified by Kandel and associates (Hawkins, Castelluci, and Kandel, 1981; Kandel and Schwartz, 1982; Carew, Hawkins, Abrams, and Kandel, 1985).

The immediate goal of our research was to implement a computational model of the basic *Aplysia* circuit. By doing so, we hoped to arrive at an appropriate level of analysis in terms of the degree to which the biological properties of the neurons in the circuit are described that will allow realistic characterization of the behavior of the circuit. Our long-term goal is to utilize this level of computational analysis to account for the phenomena of learning and memory exhibited by the more complex memory trace circuits in the mammalian brain, particularly the cerebellar circuit that appears to be the essential memory trace circuit for the learning of discrete, adaptive behavioral responses (McCormick & Thompson, 1984a, 1984b; Clark, McCormick, Lavond & Thompson, 1984; Lavond, McCormick & Thompson, 1984)

The basic reflex studied in the *Aplysia* is withdrawal of the siphon, mantle shelf and gill to tactile stimulation of the siphon or mantle shelf. If weak stimulation of the sensory nerves (CS) is followed by strong shock to the tail (US), the synaptic potential of the motor neurons to the CS is facilitated. If repeated paired trials are given, this enhancement persists, yielding the basic phenomenon of classical conditioning, a persisting associatively induced increase in response of motor neurons to the CS. This conditioning

---

For their insightful comments and suggestions, we are indebted to Joseph Steinmetz, Leon Cooper, Mortimer Mishkin, Nelson Donegan, Misha Pavel, Stephen Kosslyn, and Terry Sejnowski. The assistance of Audrey Weinland and Katie Albiston is also gratefully acknowledged. This research was supported by ONR grant #N00014-83K-0238. Please address correspondence to: Mark A. Gluck, Department of Psychology, Stanford University; Bldg. 420, Stanford, CA 94305.

depends critically on the time between presentation of the CS and the US, as noted above. The tail shock US pathway involves interneurons which are thought to exert the US presynaptic action on the sensory nerve terminals. Hawkins and Kandel (1984) propose that conditioning results from the interplay of habituation and sensitization in a manner very similar to the dual-process view of habituation suggested by Groves and Thompson (1970).

Level of Analysis

Our primary focus in this modeling effort was on the behavioral conditioning data. We began by specifying the level of description of the data in which we were interested, as opposed to specifying, *a priori*, what level of biological detail we wanted to include in the model. Our basic goal was to account for the effects of the temporal relations between input events (**CS** and **US**) on the magnitude of output events (**MN**). In this paper we focus only on short-term learning and exclude longer-term effects. Our strategy is to be only as biologically precise as necessary in order to explain the relevant behavioral phenomena. We began by starting with the simplest possible representation of the circuit. After implementing this, and understanding what behavioral phenomena it did--and did not--account for, we added complexity, *constrained* by the neurobiological data.

Components of basic model

The initial circuit is composed of three neurons and three synapses, as represented in Figure 1(a). The neurons include: a (to be) conditioned stimulus: (**CS**), an unconditioned stimulus (**US**) and a motor neuron (**MN**). One fiber originates at the conditioned stimulus and terminates as a synpase on the motor neuron (**CS→MN** synapse). Two fibers originate at the unconditioned stimulus; one terminates as a synapse on the motor neuron (**US→MN** synapse), and one terminates as a synapse on the **CS→MN** synapse (**US→** {**CS→MN**} synapse).

Neurons are represented continuously by an **Activation** which ranges from 0 to 1. This value is interpreted discretely during each time cycle as a binary value --*fired* or *not fired*-- determined probabilistically from the activation. Synapses are represented continuously by a **Strength**, which ranges from 0 to 1 and also has a probabilistic interpretation. It represents the probability of a synaptic terminal passing a "pulse" to the post-synaptic neuron if the pre-synaptic neuron has fired. Each **CS** synaptic terminal has the potential to be modified in a pairing specific manner which peaks some time after the synapse receives a pulse. The time course of this potential determines the possible *Inter-Stimulus-Intervals*. At this level of modeling we assumed that the **CS** synaptic terminals have this temporal information **without** specifying the chemical or biological source.

The simulation begins by reading the input activation levels of the **CS** and **US** neurons. From these activations the states of the neurons (e.g. *fired* or *not fired*) are probabilistically determined. If an input neuron has fired, then with a probability determined by the appropriate synaptic strengths, a pulse is received by the **MN**. Thus, if the **MN** receives a pulse from either input neuron, then **MN** Activation increases exponentially, proportional to 1 minus the current Activation, at a rate determined by the *Activation Increment Rate*. If no pulse is received by the **MN**, its Activation decreases exponentially towards 0, at a rate determined by the *Activation Decrement Rate*. Every time a synaptic

terminal passes a pulse, the strength of that synapse decreases exponentially at a rate determined by the *Habituation Rate*. If the **US→{CS→MN}** synaptic terminal passed a pulse, then with probability **CS**-Plasticity-Potential (as determined by the *Plasticity Parameter*) it will sensitize the **CS→MN** proportional to 1 - **CS.Strength**.

Simple Associative Learning

The model successfully models the basic associative learning phenomena: In the initial state, the **US** produces a large amount of activity in the **MN** compared to only a small amount produced by the **CS**. After repeated presentations of the **CS** followed by the **US** at an optimal Inter-Stimulus Interval (ISI), the **MN** response produced by the **CS** increased significantly. Following the removal of the **US**, both the **CS→MN** strength and the **MN** activity during presentation of the **CS** decay back to their initial state, resulting in the behavioral phenomena of extinction.

With simultaneous presentation of **CS** and **US** (e.g. ISI=0), little or no learning occurs because the Sensitization Potential of the **CS→MN** synapse is at 0 when the **US** fires. With an ISI that is longer than optimal, some learning occurs, but less learning than with an optimal ISI.

More Complex Associative Learning

In addition to simple conditioning, we would like to model the mechanisms responsible for differential conditioning, second order conditioning, and blocking. In differential conditioning an animal learns to respond specifically to one conditioned stimulus and not to another unconditioned stimulus. In the *Aplysia*, a *CS+* is presented to the siphon paired with a US while an unpaired *CS–* is presented to the mantle (or visa versa). In second-order conditioning a CS_1 is first conditioned via pairing with the *US*. After this training is complete, the CS_1 can serve as a reinforcing stimulus to condition a new stimulus CS_2. Blocking is a process whereby an animal learns not only about the contiguity of stimuli, but also about their predictive contingency. If the CS_1 is conditioned to predict the *US* then the addition of a second stimulus CS_2, simultaneous with the CS_1, does not produce conditioning to the CS_2 alone.

Following Hawkin & Kandel (1984) we added a second **CS** and a **Facilitator Interneuron**--see Figure 1(b)--whose behavior mimics the **MN** and which sensitizes all **CS→MN** synapses. Given these additions, the circuit model produces successful simulations of second order conditioning of CS2 to CS1, but fails to produce a blocking effect.

To produce blocking, we needed a mechanism to turn off the **US**'s ability to sensitize when it has already been predicted by some **CS**. Hawkins and Kandel (1984) suggest that the interneuron goes into a refractory period after being activated, which is longer than the possible ISI. This was implemented computationally by creating an additional variable, the **Refractory State**, which is set to a constant when the interneuron activation exceeds a threshold (.9 in the simulations shown), and then decays towards zero according to an ogive (e.g. S-shaped) function. The **Refractory State** affects the interneuron by probabilistically governing the growth of interneuron activation in the following manner: if any synapse passes a pulse to the interneuron, then activation increases with probability equal to the lesser of 0 and 1 minus the **Refractory State**. To produce the

appropriate blocking behavior the decay of the **Refractory State** was set so that the refractory period would be longer than than the potential ISI. If, however, the interneuron is in a refractory period when the **US** fires, a direct **US→MN** connection is needed in order to get an appropriate unconditioned response. Repeated attempts, however, to get this circuit simulation to produce blocking, **failed** to do so.

We were initially convinced that the circuit really should produce blocking. We tried, without success, to vary all the parameters in an attempt to produce blocking. This highlights a methodological difficulty inherent in the use of computer models for making claims about circuitry: By simulating a circuit, one can demonstrate what a circuit can do, but one cannot prove, based solely on the inability to *simulate* a desired behavior, that the real circuit is *unable to produce* this behavior. If, however, the insights gained from the "hands on" experience of building the circuit can be translated into a convincing logical demonstration of the circuit's information processing limitations, then a simulation can contribute to making an argument about a particular circuit. We outline below why we believe that the circuit simulation will not produce blocking.

The Blocking Paradox

If the activity of the **F. Int** determines both the acquisition of new conditioned pathways and the retention of previously learned pathways, then the **F. Int** must, during the presentation of a "predicted" **US**, have a differential effect on the **CS #1** and the **CS #2** for it to sensitize the **CS #1** sufficiently to retain the previously learned association but not sensitize the **CS #2** enough to acquire this new association. The current formulation of the blocking mechanism is not sufficiently detailed to give rise to these behaviors. This is not to say, however, that the current circuit could not generate blocking. Rather, the interaction between the mechanisms for blocking and habituation is more subtle than previously realized. The paradox exists not so much in the circuit, but in the current level of detail at which the circuit's mechanisms are specified, at least in our simulation.

The locus of the paradox lies in the fact that no special mechanism for the decay of a learned response is proposed. Instead, following the Groves and Thompson model, decay of learned responses during **CS** alone trials is controlled by the background phenomena of habituation. Previous theoretical models, such as Sutton and Barto (1981), have missed this paradox because--following Rescorla and Wagner (1972)--they propose an active process which extinguishes the learned association during **CS** alone trials.

Possible Solutions

A resolution of this paradox will involve specifying mechanisms of pairing specific sensitizitization which robustly predict the blocking of the **CS #2** and yet at the same time, retain the **CS #1** association. We consider here two alternatives: the first involves modeling the current circuit anatomy at a more detailed level, and the second involves postulating additional circuitry described at the current level of detail. We emphasize that these extensions are not predictions for the *Aplysia* circuit, rather they are an attempt to understand the limitations of the current circuit by exploring what extensions to the circuit might, in theory, produce blocking.

If both retention and acquisition are governed by the same Interneuron--as Hawkins and Kandel suggest--then the activity of the **F. Int**, during the presentation of a "predicted" **US**, must be sufficient to retain the **CS #1** association, but insufficient to acquire the **CS #2** association. A learning mechanism which required a stronger pulse to acquire an association than to retain an association could perhaps give rise to the desired circuit behavior. An implementation of this mechanism produced successfull, but *weak*, blocking.

An alternate method for differentiating between retention and acquisition is to posit different neural mechanisms governing retention and acquistion. Consider the addition to the circuit of a second interneuron which does not go into a refractory period (i.e. mimics the **MN**) and which sensitizes proportional to the current learned association. This interneuron would counteract the effect of the habituation of an already learned association but would would have no effect on an unlearned association. We implemented this possibility and the resulting circuit exhibited a strong blocking effect.

CONCLUSIONS

Our computational model of the basic neural circuit of the *Aplysia*, as proposed by Kandel and colleagues, is sufficient to produce basic associative learning phenomena, namely acquisition, extinction, differential conditioning, and second-order conditioning. There are, however, problems with the computational circuit in accounting for blocking. The mechanisms proposed for blocking are not sufficiently detailed to explain both blocking and the habituation (extinction) of learned responses.

Our analysis illustrates the complexities that arise in trying to understand a simple circuit involving only four neurons that generates phenomena of associative learning. Our results illustrate the need for computationally implemented quantitative theories of neuronal circuit function. If the functioning of even this simple circuit is not evident at a logical-qualitative level of analysis, then the more complex circuits that code, store and retrieve memories in the mammilian brain will certainly require quantitative modeling.

References

Carew, T. J., Hawkins, R. D., Abrams, T. W., & Kandel, E. R. (1985). *Journal of Neuroscience.*

Clark, G. A., McCormick, D. A., Lavond, D. G., Baxter, K., Gray, W. J., & Thompson, R. F. (1984). Effects of lesions of cerebellar nuclei on conditioned behavioral and hippocampal neuronal responses. *Brain Research, 291,* 125-136.

Davis, W. J. & Gillette, R. (1978). Neural correlates of behavioral plasticity in command neurons of Pleurobranchaea. *Science, 199,* 801-804.

Groves, P. M. & Thompson, R. F. (1970). Habituation: A dual-process theory. *Psychological Review, 77,* 419-450.

Hawkins, R. D., Castellucci, V. F., & Kandel, E. R. (1981). Interneurons involved in mediation and modulation of gill-withdrawal reflex in Aplysia. I. Identification and characterization. *Journal of Neurophysiology, 45,* 304-314.

Hawkins, R. D. & Kandel, E. R. (1984). Is there a cell-biological alphabet for simple forms of learning?. *Psychological Review, 91,* 376-391.

Hoyle, G. (1980). Learning, using natural reinforcements, in insect preparations that permit cellular neuronal analysis. *Journal of Neurobiology, 11,* 323-354.

Kandel, E. R. & Schwartz, J. H. (1982). Molecular biology of learning: Modulation of transmitter release. *Science, 218,* 433-443.

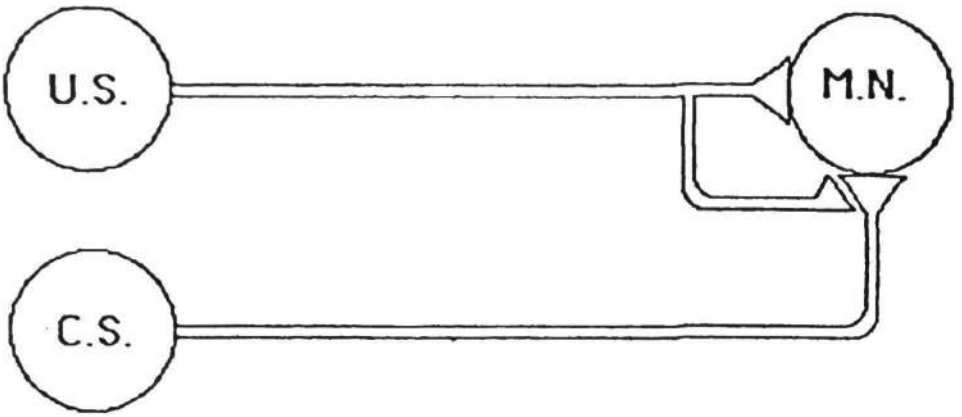
Lavond, D. G., McCormick, D. A., & Thompson, R. F. (1984). A nonrecoverable learning deficit. *Physiological Psychology, 12,* 103-110.

McCormick, D. A. & Thompson, R. F. (1984a). Cerebellum: Essential involvement in the classically conditioned eyelid response. *Science, 223,* 296-299.

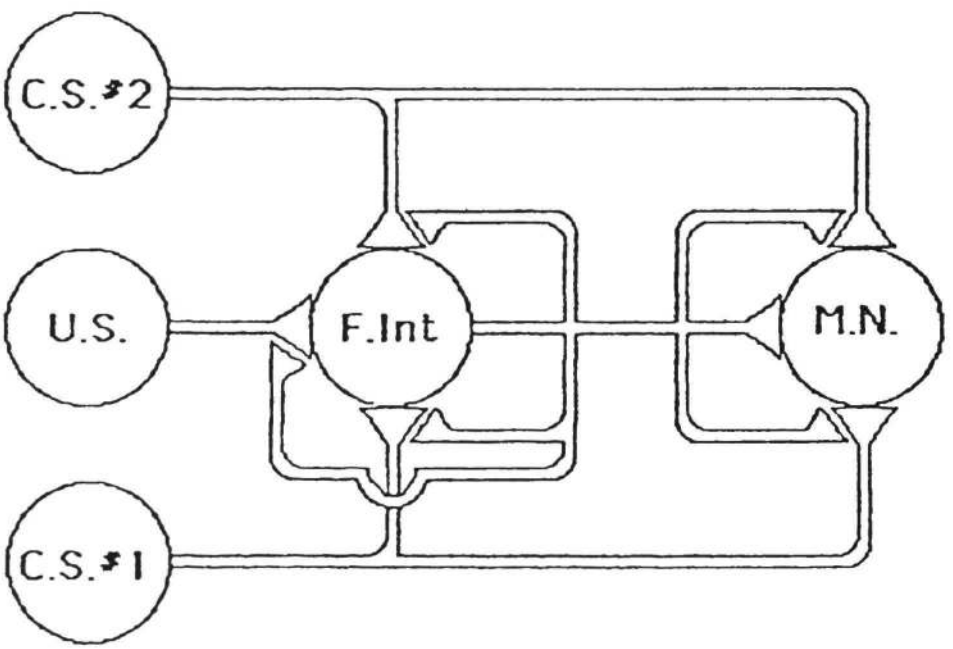
McCormick, D. A. & Thompson, R. F. (1984b). Neuronal responses of the rabbit cerebellum during acquisition and performance of a classically conditioned nictitating membrane-eyelid response. *Journal of Neuroscience, 4,* 2811-2822.

Sutton, R. S. & Barto, A. G. (1981). Toward a modern theory of adaptive networks: Expectation and prediction. *Psychological Review, 88,* 135-170.

(A)



(B)



STRUCTURAL LEARNING IN CONNECTIONIST SYSTEMS*

Andrew G. Barto and Charles W. Anderson
Department of Computer and Information Science
University of Massachusetts, Amherst MA 01003

Abstract

Although networks of neuron-like computing elements can be constructed to implement *any* function or operation one desires, it is a highly nontrivial problem to devise algorithms that permit networks to *learn reliably and efficiently* how to realize desired nonlinear functions without being provided with implementation details. In particular, learning algorithms that work for single layers of adaptive elements cannot be extended easily to multilayer or recurrently connected networks where structural changes must be produced. We describe an approach to this problem which uses stochastic search as does the Boltzmann learning procedure (Ackley, Hinton, and Sejnowski, 1985), but which is otherwise quite different from that method. We present several simulations of layered adaptive networks to illustrate our method. In addition, we briefly review a variety of previous approaches to this general problem in order to place these various methods in perspective and to suggest a range of alternatives with which the performance of novel methods should be compared.

INTRODUCTION

One goal of connectionist modelling as pursued by researchers in Cognitive Science and Artificial Intelligence is to bridge the gap between the behavior of networks of neuron-like computing elements and complex forms of behavior that appear at higher levels (Hinton and Anderson, 1981; Feldman, 1985). Learning is likely to play an important role in this research since it may be necessary in order to take advantage of the representational potential of connectionist networks (Hinton, 1984), and since these networks contain obvious parameters that can be adjusted through experience—the connection weights. However, although networks of neuron-like computing elements can be constructed to implement *any* function or operation one desires, it is a highly nontrivial problem to devise algorithms that permit networks to *learn reliably and efficiently* how to realize desired nonlinear functions without being provided with implementation details. In particular, learning algorithms that work for single layers of adaptive elements cannot be extended easily to multilayer or recurrently connected networks.

A significant advance in this area is the Boltzmann learning procedure recently described by Hinton and Sejnowski (1983) and Ackley, Hinton, and Sejnowski (1985) that is based on the analogy between thermodynamic systems and networks of neuron-like elements pointed out by Hopfield (1982). In this paper we describe an approach to this problem which uses stochastic search as does the Boltzmann procedure, but which is otherwise quite different from that method. We present several simulations of layered adaptive networks to illustrate our method. In addition, we briefly review a variety of previous approaches to this general problem in order to place these various methods in perspective and to suggest a range of alternatives with which the performance of novel methods should be compared. Most of these studies are quite old, but we think they are relevant given the renewed interest in networks of this kind.

1. PARAMETRIC VERSUS STRUCTURAL LEARNING

One of the problems with learning systems using the single-layer learning procedures is that learning proceeds up to a certain point and then stops. When the parameters that are adjusted by the learning algorithm—in a network, usually the connection weights—reach optimum values, the degrees of freedom of the system are exhausted even though the problem facing the system may be far from solved. Somehow, this *parametric* learning should be augmented with *structural* learning by which the roles of the parameters

*This research was supported by the Air Force Office of Scientific Research and the Avionics Laboratory (Air Force Wright Aeronautical Laboratories) through contract F33615-83-C-1078. We thank Rich Sutton, P. Anandan, John Moore, and Harry Klopf for their theoretical contributions and helpful discussions.

in determining behavior, and not just their values, are altered by the learning process. Since one can always regard structures as being parameterized, so that adjusting structures amounts to adjusting more parameters, this distinction is not completely straightforward. However, what we mean by structural learning generally involves a space of parameters that is so large, and a performance evaluation surface that is so complex, that the usual algorithms for parametric adaptation do not work. Structural learning is intimately related to the problem of adaptively developing new representations, for example, by the creation of "new terms," since it is the representation that determines the roles of the parameters.

One can view the adjustment of a connection weight in a complex network as a structural adjustment since it affects the roles of other weights in generating network behavior. A complex network will have very many adjustable weights, and the relationship between changes in a weight and changes in network performance (i.e., the gradient of the network performance index with respect to the weight) will be complicated by nonlinear dependencies on the weights of other elements—dependencies that do not make themselves known through information *locally* available to the connection in question. Additionally, even if this gradient could be determined locally, following it can lead to network performance that is only locally optimal. Global searches that do not suffer from this deficiency may be too slow for the large search spaces that arise in structural learning.

2. LAYERED AND RECURRENT NETWORKS

Our work to date has been restricted to the study of layered networks that do not have recurrently connected elements. The Boltzmann learning method, on the other hand, is restricted to symmetrically connected, hence totally recurrent, networks. In layered networks the entire stage corresponding to the running of a Boltzmann network or Harmony system (Smolensky, 1983) to "thermal equilibrium" using simulated annealing appears in a degenerate form: it is just the process of evaluating the input/output function realized by the network, and no iterative relaxation procedure is required. Hence, layered networks do not solve nontrivial constraint satisfaction problems. On the other hand, once a layered network has learned, its performance in computing this function is essentially instantaneous.

We have restricted attention to layered networks because it seemed to us that obtaining structural learning in this case would be easier than, and a prerequisite for, obtaining it in the recurrent case. Boltzmann learning shows that this is not true, but the principle employed there does not appear to extend to asymmetric networks. We have not yet decided on the best way to extend our approach to the recurrent case, but we do not think it is inherently limited to nonrecurrent networks. Future research will concern the case of recurrent but asymmetric networks. We have also been influenced by the extensive history of attempts to extend single-layer learning results to nonrecurrent networks having multiple layers, and we briefly review some of these studies in order to place our approach in its proper context.

3. REVIEW OF LAYERED NETWORK STUDIES

Assume that a multilayered network has been designed by defining the output of each element as a parameterized function of its input, and by specifying the interconnections among the elements. What values should be assigned to the parameters of the network in order to implement some desired input/output function? The most straightforward approach is to directly search the space of the network's parameters for those values that maximize some measure of the network's overall performance. By a direct search we mean one in which successive sets of parameter values (i.e., weights) are evaluated by seeing how the network performs with those values in its required task. Relevant gradient information is not obtained locally by the adaptive elements, and a centralized search control mechanism is required. Gilstrap (1971), who has pursued this direct-search approach, used guided random search methods that can be effective under conditions encountered with multilayered networks. Whatever the search method used, however, directly searching a space of dimension equal to the number of weights in an entire network is an extremely time-consuming process for all but the simplest cases.

Most results from early adaptive network research concern layered networks in which only the elements of the final (output) layer adapt. Examples of such networks are the Perceptron of Rosenblatt (1962) and networks of "adalines" (*ada*ptive *lin*ear *e*lements) studied by Widrow (1962). In these cases the learning algorithms are variants of the now well-known error-correction procedures that adjust weights based on the discrepancy between a unit's response and its desired response supplied by some agency in the network's

environment (a "teacher"). There are obvious difficulties in extending these error-correction techniques to layered networks. Adapting the terms of Hinton and Sejnowski (1983), let us distinguish a network's *visible elements* from its *hidden elements*. Visible elements are those whose activity is directly available to the network's environment and that are required to assume certain values for various input patterns provided to the network. Hidden elements are those whose activity is not directly visible and that are somehow to provide an encoding of input signals that will allow the visible elements to respond correctly. Although in many tasks it may be possible for the network's teacher to provide error signals to the network's visible elements (since it is these that define the network's response), it may not be possible for this teacher to provide analogous error signals to the hidden elements without *a priori* knowledge of the implementation details of the desired input/output function. If this knowledge were available, then the problem would be quite different from the ones in which we are interested: it would be more of a programming problem than a learning problem.

Some methods rely on the generation of new elements rather than on the adjustment of the parameters of existing elements. Methods that generate new elements generally divide the learning process into two stages. In the first stage, the weights of the first layer (i.e., the layer that directly receives the external input signals) are held constant while one of the familiar single-layer learning algorithms is used by the second layer. In the second stage, the second layer is held constant while new elements are added to the first layer. Those elements whose outputs are not significantly influencing the second layer might be discarded to limit the number of elements. Selfridge's Pandemonium provides an example of this two-stage learning process (Selfridge, 1959), where new elements are created by "mutated fission" and "conjugation" of existing elements. Although it is not described in network terms, the classifier system of Holland (1980) is probably the most highly developed example of generating new elements via this type of "genetic recombination" process. Uhr and Vossler (1961) presented another system that effectively adds new elements to the first layer. Each new element is set to respond to a subpattern of the current input pattern. Reilly, Cooper, and Elbaum (1982) recently proposed a somewhat related method that incorporates input patterns as "prototypes."

Methods for generating new elements are attempts to avoid the combinatorial explosion that would result from having an element for each of the possible combinations of available signals. The heuristic employed is that useful higher-order features will tend to be compositions of useful lower-order features. A technique using this heuristic may be viewed as a type of *beam search* (see Barr and Feigenbaum, 1981). The search is conducted by forming all pairwise (for example) combinations of lower-order features at each stage, and then removing from consideration all but a certain number of them before forming the next stage's combinations. At each stage, the number of features remaining is the *beam width* of the search. A beam search is not guaranteed to result in an optimal solution but can be efficient if the beam is sufficiently narrow. Although not usually associated with networks of adaptive elements, beam search can obviously be related to layered networks. Ivanhenko's (1971) Group Method of Data Handling, for example, can be regarded as a method for constructing a layered network using a beam search.

Another approach is to train the first layer of a two-layered network in isolation, independently of the second layer and of the network's performance on the required task. Such open-loop procedures are referred to as *clustering* methods. Clustering methods are based on the assumptions that the system's input patterns tend to fall into natural clusters due to their intrinsic structure, and that detecting these clusters is significant in some way for the performance of the system. Block, Nilsson, and Duda (1964) used a clustering algorithm to train the first layer of a two-layered network. Fukushima's Cognitron (1973) and Neocognitron (1980) are other examples of clustering algorithms implemented as networks. In these networks an element is selected for adjustment if its output is sufficiently in excess of the outputs of neighboring elements, and its weights are adjusted so as to cause it to be more vigorously activated by the current input pattern. This type of learning has also been discussed by Grossberg (1976a, 1976b) and Rumelhart and Zipser (1985).

For many types of problems, the need is not just to form clusters of input patterns but to form clusters that are *useful* in terms of the network's interaction with its environment. In order to accomplish this, the initial layers must use the information contained in an error or evaluation signal that is provided by the network's environment. The problem, as discussed above, is that this error or evaluation is directly a function only of the network's visible elements. Somehow this information must be used to tune the hidden elements. Rosenblatt (1962) reported experiments with a stochastic back-propagation scheme for generating error signals for interior elements, but the simplest way of doing this is to adjust a randomly chosen element when an error is made by an output element. This approach was analyzed by Alder (1975) who proved an extension of the Perceptron Convergence Theorem for layered networks. As he pointed out, however, the

algorithm was "less than efficient."

Another method for selecting elements to adjust is to select those elements that would require the least amount of change to correct the network's error. Widrow used this method in networks consisting of two layers of adalines (1962). This algorithm and similar ones (e.g., Stafford, 1963) require a rather sophisticated agent to conduct the training of the interior elements. In some cases, the sophistication required by this agent can be reduced if the network structure is sufficiently constrained. Widrow's (1962) study of "madalines" (*m*ultiple *adalines*) can be interpreted in this way. Here the final layer implements a *fixed* logical function, and only the initial layer learns in a manner that depends on this logical function. Systems like this have been called "committee machines." Methods similar to this have been discussed recently by Reilly, Cooper, and Elbaum (1982), mentioned above, and Hampson and Kibler (1982) and seem to offer promising ways of using networks for nonlinear pattern classification.

Some of the aforementioned methods represent attempts to extend error-correction methods to all elements of the network, for example, by restricting network operation so that desired responses for interior elements might be deduced. Another approach is to use elements that do not require desired responses or error signals but that implement reinforcement-learning algorithms. Such elements are capable of improving performance with respect to an evaluation signal that assesses the *collective activity* of the network components. This method differs from what we called direct search since activity patterns rather than weight settings are directly evaluated and the elements locally estimate the relevant gradient information. Our own approach and that suggested by Klopf (1972, 1982) fall into this category, and we know of only a few earlier studies that are similar. In his Ph.D. thesis, Minsky (1954) described the SNARC (Stochastic Neural-Analog Reinforcement Calculator) which he constructed in 1951. It used components implementing a simple stochastic reinforcement-learning procedure. Farley and Clark (1954) experimented with stochastic adaptive elements that are similar in principle to the adaptive elements we have developed. In simulation experiments, recurrently connected networks of these elements were able to solve some simple discrimination tasks. Widrow, Gupta, and Maitra (1973) presented an extension of the adaline algorithm to allow it to do a form of reinforcement learning which they called "selective bootstrap adaptation." They remarked that this extension may permit the elements to learn as components of layered networks. We now describe the reinforcement-learning approach in more detail.

4. LAYERED NETWORKS OF REINFORCEMENT LEARNING ELEMENTS

Consider an adaptive network operating in an environment that can evaluate the behavior of the network, that is, of the collective behavior of the network's elements, but cannot specify the desired behavior of each individual component.[1] Suppose that the environment evaluates each of the network's overt actions by generating a reinforcement signal that is made available each element of the network.[2] If we view the problem from the perspective of an individual element embedded in the interior of this network, we can gain some understanding of the type of learning capability such an element might have to possess. Even if the environment deterministically evaluates the network's actions, the relationship between this element's actions and the evaluation signal will not be deterministic because it also depends on the behavior of other elements. In addition to this, the contingencies faced by the element will vary with time as the other elements adapt. Thus, even if the overall task faced by the network involves only fixed deterministic contingencies, the task faced by an individual element will involve nonstationary random contingencies.

If all the elements in the network are able to improve their individual levels of performance under these conditions, then the collection will also tend to improve its performance. This type of process involves cooperative behavior more closely related to that discussed in game theory and economics than it is to the cooperative phenomena of physics to which Boltzmann learning is related. One can regard the elements as self-interested agents and a network as a "team." This perspective on connectionist learning is due to Klopf

[1] By an agency in a network's environment, we do not necessarily mean an agency outside of the device in which the network resides; this agency may be another component of the overall learning system, such as a module specialized for delivering reinforcement to other modules.

[2] In the research reported here, we have not focussed on problems created by delayed evaluation. We have extensively studied these problems and results are reported elsewhere (Barto, Sutton, and Anderson, 1983; Sutton, 1984).

(1972, 1982), whose theory of the "hedonistic neuron" suggests that many aspects of learning, memory, and intelligence may arise from this type of cooperativity.

5. THE A_{R-P} LEARNING RULE

Together with R. Sutton, we have studied several types of adaptive elements capable of reinforcement learning (Anderson, 1982; Barto and Sutton, 1981; Barto, Sutton, and Anderson, 1983; Sutton, 1984), but the one used in the simulations described here was developed by Barto and Anandan (in press) who called its learning algorithm the *associative reward-penalty*, or A_{R-P}, *algorithm* and proved a convergence theorem. Details of this learning algorithm are provided in the appendix. Here there is only space to point out the following. An element implementing this algorithm is a linear threshold device with a randomly varying threshold. We use the logistic distribution function so that an element fires with probability $1/(1+e^{-s/T})$, where s is the total stimulus strength. Thus, the input/output behavior of the element is identical to that of the elements used in Boltzmann learning, where T is the "computational temperature." None of our results, however, require the use of this specific distribution function. After each action, the element receives a reinforcement signal taking values $+1$ and -1 to respectively indicate "reward" and "penalty." By updating its weights at each step (see the appendix) the element is able to improve its performance when its environment provides stimulus patterns and reinforcement feedback according to the following probabilistic scheme. At each step the environment presents the element with an input pattern $x \in X \subset \Re^n$ (where $\Re$ denotes the real numbers). For each pattern x in X and each of the element's actions, $y = 0$ or 1, the environment returns "reward" with probability $d(x,y)$ when the element emits action y in the presence of input pattern x. It delivers "penalty" with probability $1 - d(x,y)$. The element would maximize its probability of receiving reward if it responded to each x in X with the action y for which $d(x,y)$ is largest. Learning tasks like this one are related to instrumental, or cued operant, tasks used by animal learning theorists (where an input pattern x corresponds to a discriminative stimulus) and to "two-armed bandit" tasks studied by mathematicians and engineers (e.g., Cover and Hellman, 1970; Narendra and Thathachar, 1974)

Since it need not be true that $d(x,1) + d(x,0) = 1$, for a given input pattern x it might be true that no matter what action the unit produces, it usually receives reward (i.e., $d(x,1) > .5$ *and* $d(x,0) > .5$); or it might be true that no matter what action the element produces, it usually receives penalty (i.e., $d(x,1) < .5$ *and* $d(x,0) < .5$). Given these possibilities, the feedback received from producing one action provides *no information about the suitability of the other action*. This property makes this task significantly more difficult than the tasks usually solvable by neuron-like adaptive elements, yet it is an unavoidable feature of tasks faced by the hidden elements. Barto and Anandan (in press) prove that the A_{R-P} algorithm is asymptotically optimal[3] for arbitrary probabilistic contingencies if the set X of stimulus patterns is linearly independent. Interestingly, when the temperature T is zero, the A_{R-P} algorithm reduces to the Perceptron algorithm modified to accept reward/penalty feedback instead of training information in the form of desired responses. So restricted, however, optimal performance is obtainable only for deterministic environments ($d(x,y) = 0$ or 1, for all x and y), and such elements are not able to learn reliably when embedded in networks when a reinforcement-learning paradigm is used.

6. A MINIMAL CASE OF COOPERATIVE LEARNING

Fig. 1 shows a network of two A_{R-P} elements, e_1 and e_2. Only e_1 receives input patterns from the environment, and only the action of e_2 is available to the environment (e_1 is hidden; e_2 is visible). Suppose the network's output, the output of e_2, affects the probability of reward for both elements in a manner that depends on the stimulus pattern presented to e_1. If there were no means for e_1 to communicate with e_2, the elements would be capable of achieving only limited reward frequencies. The action of e_2 influences the reinforcement received by both elements, but in the absence of a communication link, e_2 remains blind to the discriminative stimulus x. On the other hand, in the absence of a communication link, e_1 can sense the discriminative stimulus but cannot influence the network's actions. The complementary specialties of the two elements have to be combined in order for each to attain optimal performance. If the weight connecting

[3]More precisely, it is ϵ-optimal for each $x \in X$, to use the terminology of learning automata theory (see Narendra and Thathachar, 1974).

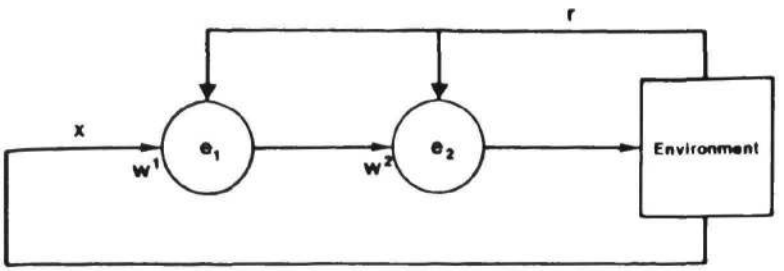


Figure 1: A two-element series in a simple discrimination task

e_1 to e_2 can be adjusted properly, the network can respond correctly. However, the correct value of the interconnecting weight depends on how e_1 has learned to respond to x. Conversely, the correct behavior of e_1 depends on the value of the interconnecting weight. Thus the two elements must adapt simultaneously in a tightly-coupled cooperative fashion.

To be more specific, we set up the simulation in the following way. A training step consists of presenting a randomly chosen input signal to the network, computing the network's output, determining the reinforcement signal, and then updating the weights.[4] The stimulus signals, $x = 0$ and $x = 1$, are equally likely to occur at each step, and the success probabilities implemented by the network's environment are given by the following table:

x	$d(x,0)$	$d(x,1)$
0	.9	.1
1	.1	.9

Thus it is optimal for the network as a whole to respond to $x = 0$ with visible action 0 to obtain reward with probability .9, and to respond to $x = 1$ with action 1 to obtain reward with probability .9, yielding an overall reward probability of .9. If the discrimination is not made so that the network responds identically to all input patterns, the overall success probability is $(.9 + .1)/2 = .5$. Since each element also adaptively adjusts its threshold (more precisely, the mean value of its threshold by adjusting a "threshold weight"), there are two ways the network can solve this problem—both elements can implement the identity map, or both can invert their input signal—and there are many ways it can fail.

Fig. 2a and 2b show the behavior of the network for a typical sequence of 500 training steps with $\lambda = .04$ and $\rho = 1.5$ (see the appendix). Fig. 2a shows the evolution of the behavior of e_1 in terms of two graphs. The first shows the conditional probability that e_1 fires ($y^1 = 1$) given that its input is 0, and the second shows the same thing for input 1. Both of these probabilities start at .5 since the weights are initially zero. Fig. 2b shows the evolution of the mapping implemented by e_1 and e_2 acting together by showing the probability that e_2 fires ($y^2 = 1$) for the different values of the network input x. Fig. 2c shows the evolution of the overall reward probability. Fig. 2d is a histogram of the number of steps required to reach a criterion of 98% of optimal performance for each of 100 sequences of trials. In all of the sequences the network reached this criterion before 1,500 steps. In about half of the sequences both elements learned to implement the identity map, and in the other half, both became inverters.

A series of two elements in a discrimination task provides one of the simplest examples we could devise to demonstrate cooperative reinforcement learning. We interpret the result as illustrating cooperativity in the literal game-theoretic sense, with the interconnecting link representing a "binding agreement" by which the elements form a coalition for mutual benefit.

[4]Note that in contrast to Boltzmann learning, the weights are updated after the each presentation of a single input pattern.

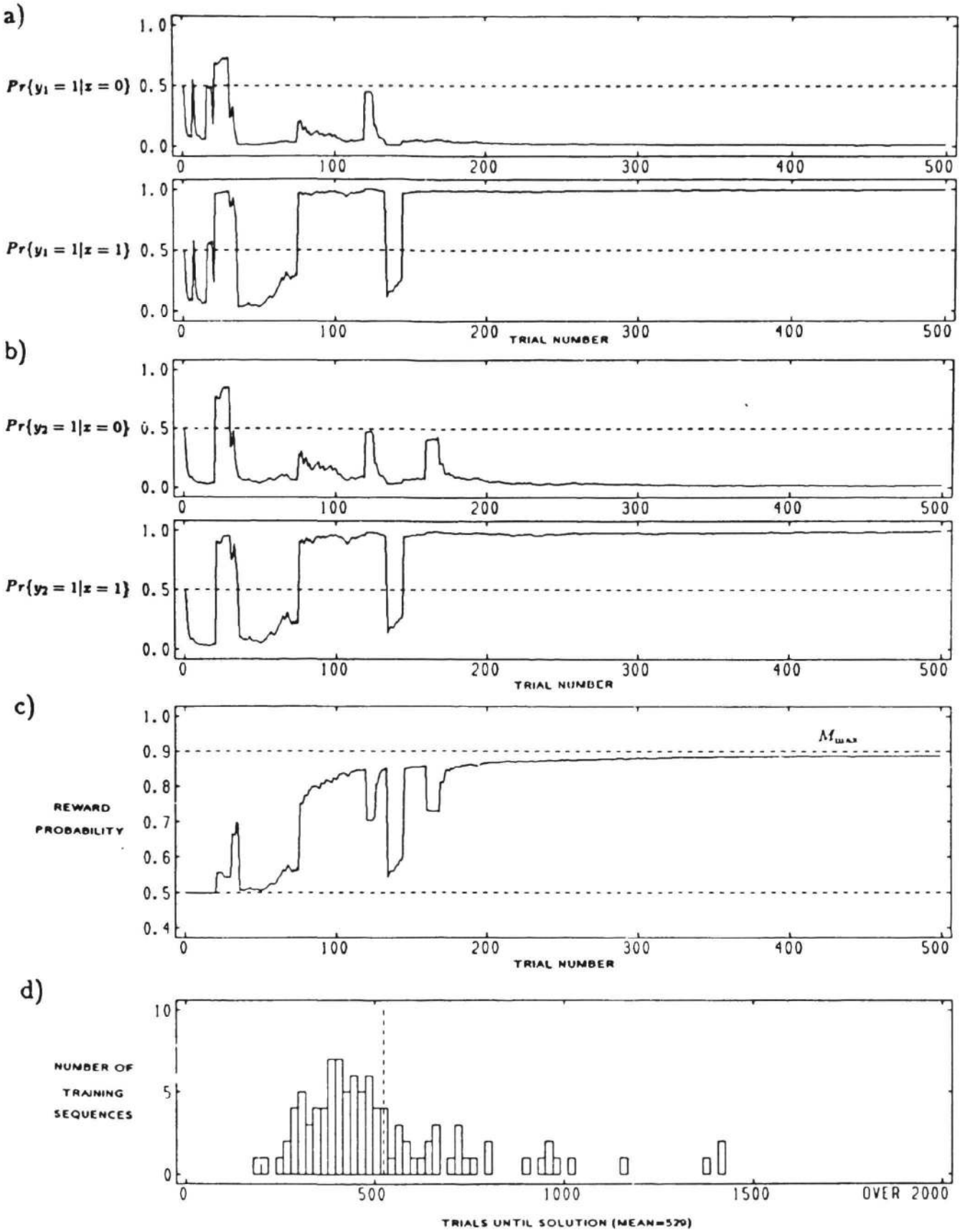


Figure 2: Simulation results for the two-element series. See text for explanation.

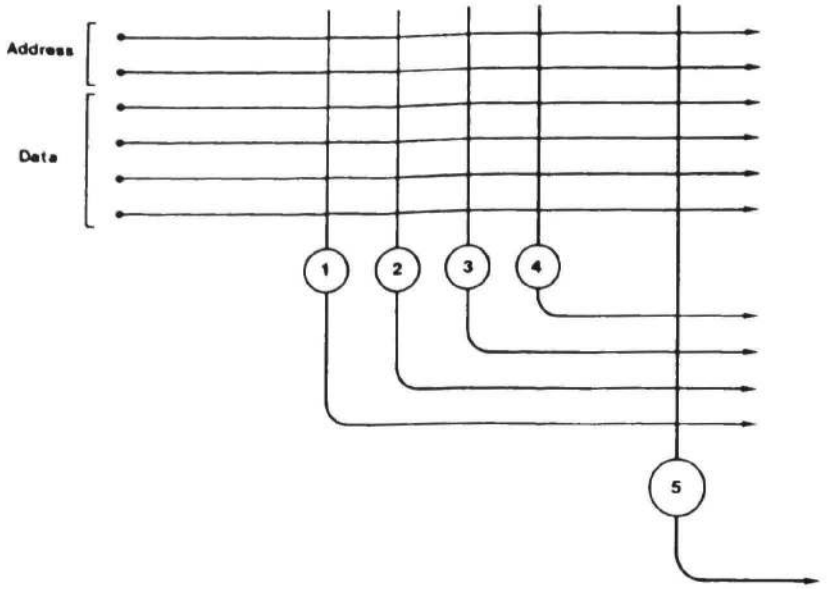


Figure 3: Network for the multiplexer task

7. A NONLINEAR TASK

In the task just described, cooperative learning is required only because the network lacks a direct pathway from input to output. The task itself is easily within the capabilities of a single element. Here we discuss a task that cannot be solved by a single linear threshold element, or any single-layer network of them. The network shown in Fig. 3 has six input components and a single principal output (from element 5). There are 39 weights to adjust, one associated with each of the pathway intersections and one threshold weight for each element. The reward contingencies implemented by the network's environment force the network to learn to realize a multiplexer circuit in order to obtain optimal performance. A multiplexer is a device with n address inputs and 2^n data inputs (here $n = 2$). Given a pattern over the address pathways, i.e., an address, a multiplexer's output is equal to whatever signal (0 or 1) appears on the data line associated with that address. It therefore routes signals from different input pathways to a single output pathway depending on the "context" provided by the pattern over the address pathways. For each of the 64 possible input patterns, we rewarded each element of the network with probability 1 if the visible element (number 5) produced the correct output, and we penalized each element with probability 1 otherwise. The input patterns were chosen randomly for presentation to the net. All of the elements implement the A_{R-P} algorithm with $T = .5$ except for the visible element (number 5) which uses $T = 0$ (and therefore essentially uses the Perceptron algorithm).

This is a highly nonlinear task since the natural generalizations over the set of input patterns tend to be wrong with respect to the required actions of the network. Consequently, it does not show the strengths of distributed representations (see Hinton, 1984), but it represents a rather stringent test of the learning method. The hidden elements (elements 1–4) are necessary in order to create new features to permit the visible element to respond correctly. Fig. 4 is a histogram of the number of steps required for the network to respond 99% correctly for 1,000 consecutive steps for each of 30 sequences of trials with $\rho = 1$ and $\lambda = .01$ (see the appendix). The average number of steps required was 133,149, or about 2,080 presentations of each stimulus pattern. In every sequence the network reached the criterion before 350,000 steps.

8. DISCUSSION

The multiplexer simulation suggests that layered networks of A_{R-P} elements are able to learn to implement associative mappings that are beyond the capabilities of individual elements. More importantly, they are able to do this when being directed by evaluative feedback that is based on knowledge of "what" the network as a whole should accomplish but no knowledge of "how" the network should accomplish it. Although we have not yet proved convergence for networks of A_{R-P} elements, all of our simulations suggest

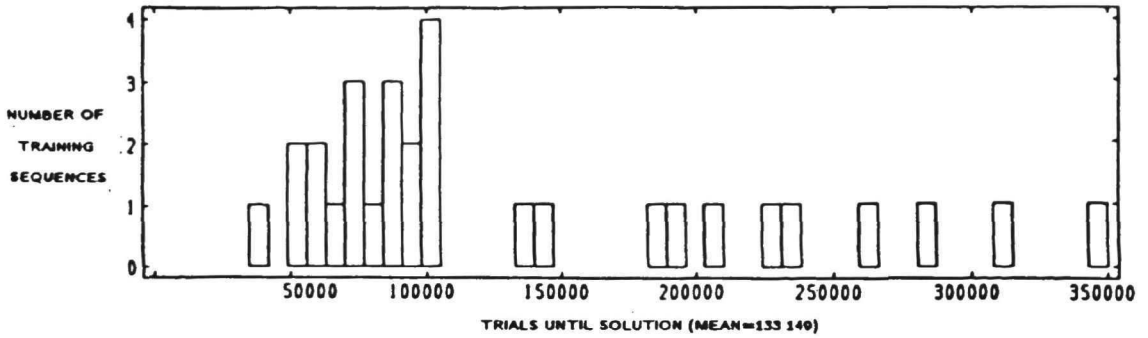


Figure 4: Simulation results for the multiplexer task. See text for explanation.

extremely reliable performance. However, these results also suggest that, like Boltzmann learning, the process may take a considerable amount of time. It is difficult to evaluate the learning rate of A_{R-P} networks without comparing their performance with that of other learning algorithms, and we are currently in the process of performing comparative simulation studies using some of the algorithms mentioned in Section 3. At present we only know that for relatively small networks the simplest direct search generally yields almost no improvement in performance by the time the A_{R-P} networks are performing near optimally. There are also a number of methods for accelerating convergence that have been developed for conventional pattern-classification algorithms with which we have not yet experimented.

Despite these important questions regarding learning rate, the learning method we have described has a number of attractive features. First, the training procedure does not require any elaborate or centralized control structures—it is a more or less "natural" consequence of the adaptive elements interacting with one another under contingencies that are simple to implement. Second, if their initial architectures permit, networks of A_{R-P} elements tend to learn the easy parts of a problem quickly so that performance tends to remain relatively high while the hard parts of the problem are being learned. Appropriate architectures are those in which hidden elements are not strictly interposed between layers but rather form auxiliary side networks. This is illustrated by the multiplexer network in which the input pathways to the network connect to the visible element as well as to the hidden elements. Finally, although it is not illustrated by the simulations described here, networks of A_{R-P} elements are able to learn in environments that cannot directly instruct even the visible elements but can only evaluate the consequences of their activity on some other process. As has been suggested elsewhere (Barto, Sutton, and Brouwer, 1981), this may be important for sensorimotor learning tasks where evaluative feedback is a function of the spatial result of a network's actions.

APPENDIX

THE A_{R-P} ALGORITHM

Assume that at the start of the t^{th} step, the environment provides an element implementing the A_{R-P} algorithm with a pattern vector $x(t) = (x_1(t), \ldots, x_n(t))$ of real numbers. The element then emits an action $y(t)$ that is determined by a random thresholding process:

$$y(t) = \begin{cases} +1, & \text{if } s(t) + \eta(t) > 0; \\ -1, & \text{otherwise;} \end{cases} \tag{1}$$

where $s(t) = \sum_{i=0}^{n} w_i(t)x_i(t)$ is the weighted sum of the input signals and the $\{\eta(t), t \geq 1\}$ are independent, identically distributed random variables (we used the logistic distribution with $T = .5$ in the simulations). Let $r(t)$ denote the reinforcement that evaluates the consequences of $y(t)$. It takes the values $+1$ and -1 to respectively indicate "reward" and "penalty." The weights, w_i, $1 \leq i \leq n$, are updated according to the following equation:

$$\Delta w_i(t) = \begin{cases} \rho[r(t)y(t) - E\{y(t)|s(t)\}]x_i(t) & \text{if } r(t) = +1; \\ \lambda\rho[r(t)y(t) - E\{y(t)|s(t)\}]x_i(t) & \text{if } r(t) = -1; \end{cases} \tag{2}$$

where $\Delta w_i(t) = w_i(t+1) - w_i(t)$, $\rho > 0$, and $0 < \lambda \leq 1$. $E\{y(t)|s(t)\}$, the expected value of the output given the weighted sum, depends on the distribution function used. For the logistic distribution function, it is

$$E\{y(t)|s(t)\} = \frac{e^{s(t)/T} - 1}{e^{s(t)/T} + 1},$$

which is a sigmoidally-shaped function of $s(t)$ with limits of -1 and $+1$ for $s(t)$ respectively approaching $-\infty$ and $+\infty$. In the simulations we recoded the values of $y(t)$ to be 1 and 0 instead of 1 and -1 when the elements communicated with one another.

If one lets the random variable $\eta(t)$ in (1) be identically zero (the deterministic "zero temperature" case) and interprets the term $r(t)y(t)$ in (2) as a training signal giving the element's desired response, then the A_{R-P} algorithm becomes an asymmetric form of the Perceptron algorithm. Additionally, if the input pattern $x(t)$ is held constant and non-zero over t, then the A_{R-P} algorithm reduces to a stochastic learning automaton algorithm (see Narendra and Thathachar, 1974); specifically it reduces to a nonlinear reward-penalty algorithm of the non-absorbing type as defined by Lakshmivarahan (1979). Barto and Anandan (in press) discuss the A_{R-P} algorithm is more detail and prove a convergence theorem.

REFERENCES

Ackley, D. H., Hinton, G. E., & Sejnowski, T. J. A learning algorithm for Boltzmann Machines. *Cognitive Science*, 1985, *9*, 147–169.

Alder, M. D. A convergence theorem for hierarchies of model neurones. *SIAM J. Comput.*, 1975, *4*, 192–201.

Anderson, C. W. Feature generation and selection by a layered network of reinforcement learning elements: Some initial experiments. COINS Technical Report 82–12, University of Massachusetts, Amherst, 1982.

Barr, A. & Feigenbaum, E. A. *The handbook of artificial intelligence, Volume 1.* Los Altos, California: Kauffman, 1981.

Barto, A. G. & Anandan, P. Pattern recognizing stochastic learning automata. *IEEE Trans. on Systems, Man, and Cybernetics*, in press.

Barto, A. G., & Sutton. R. S. Landmark learning: An illustration of associative search. *Biological Cybernetics*, 1981, *42*, 1–8.

Barto, A. G., Sutton, R. S., & Anderson, C. W. Neuronlike elements that can solve difficult learning control problems. *IEEE Trans. on Systems, Man, and Cybernetics*, 1983, *13*, 835–846.

Barto, A. G., Sutton, R. S., & Brouwer, P. S. Associative search network: A reinforcement learning associative memory. *Biological Cybernetics*, 1981, *40*, 201–211.

Block, H. D., Nilsson, N. J., & Duda, R. O. Determination and detection of features in patterns. In Tou, J. T. & Wilcox, R. H. (Eds.), *Computer and information sciences: Collected papers in learning, adaptation, and control in information systems.* Washington, D. C.: Spartan Books, 1964, 75–110.

Cover, T. M., & Hellman, M. E. The two-armed bandit problem with time-invariant finite memory. *IEEE Transactions on Information Theory*, 1970, *16*, 185–195.

Farley, B. G., & Clark, W. A. Simulation of self-organizing systems by digital computer. *I.R.E. Transactions on Inf. Theory*, 1954, *4*, 76–84.

Feldman, J. A. (Ed.) Special issue on connectionist models and their applications. *Cognitive Science*, 1985, *9*.

Fukushima, K. A model of associative memory in the brain. *Kybernetic*, 1973, *12*, 58–63.

Fukushima, K. Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. *Biological Cybernetics*, 1980, *36*, 193–202.

Gilstrap, L. O., Jr. Keys to developing machines with high-level artificial intelligence. *Design Engineering Conference, ASME*, New York, 1971.

Grossberg, S. Adaptive pattern classification and universal recoding: I. Parallel development and coding of neural feature detectors. *Biological Cybernetics*, 1976a, *23*, 121–134.

Grossberg, S. Adaptive pattern classification and universal recoding: II. Feedback, expectation, olfaction, illusions. *Biological Cybernetics*, 1976b, *23*, 187–202.

Hampson, S., & Kibler, D. A boolean complete neural model of adaptive behavior. Department of Information & Computer Science Technical Report No. 190, University of California, Irvine, CA, 1982.

Hinton, G. E. Distributed representations. Technical Report CMU-CS-84-157, Carnegie-Mellon University, Pittsburgh, PA, 1984.

Hinton, G. E., & Anderson, J. *Parallel models of associative memory.* Hilsdale, N. J.: Erlbaum, 1981.

Hinton, G. E., & Sejnowski, T. J. Analyzing cooperative computation. *Proceedings of the Fifth Annual Conference of the Cognitive Science Society*, Rochester N.Y., 1983.

Holland, J. H. Adaptive algorithms for discovering and using general patterns in growing knowledge-bases. *International Journal of Policy Analysis and Information Systems*, 1980, *4*, 217–240.

Hopfield, J. J. Neural networks and physical systems with emergent collective computational abilities. *Proc. Natl. Acad. Sci.* USA, 1982, *79*, 2554–2558.

Ivanhenko, A. G. Polynomial theory of complex systems. *IEEE Transactions on Systems, Man, and Cybernetics*, 1971, *1*.

Klopf, A. H. Brain function and adaptive systems—A heterostatic theory. Air Force Cambridge Research Laboratories Research Report, AFCRL-72-0164, Bedford, MA., 1972 (A summary appears in *Proceedings International Conference on Systems, Man, Cybernetics*). IEEE Systems, Man, and Cybernetics Society, 1974, Dallas, Texas.

Klopf, A. H. *The hedonistic neuron: A theory of memory, learning, and intelligence.* Washington, D.C.: Hemisphere, 1982.

Lakshmivarahan, S. ϵ-optimal learning algorithms—Non-absorbing barrier type, Technical Report EECS 7901, School of Electrical Engineering and Computer Sciences, University of Oklahoma, Norman Oklahoma, 1979.

Minsky, M. L. Theory of neural-analog reinforcement systems and its application to the brain-model problem. Princeton Univ. Ph.D. Dissertation. 1954.

Narendra, K. S., & Thathachar, M. A. L. Learning automata—a survey. *IEEE Transactions on Systems, Man, and Cybernetics*, 1974, *4*, 323–334.

Reilly, D. L., Cooper, L. N., & Elbaum, C. A neural model for category learning. *Biological Cybernetics*, 1982, *45*, 35–41.

Rosenblatt, F. *Principles of neurodynamics.* New York: Spartan Books, 1962.

Rumelhart, D. E., & Zipser, D. Feature discovery by competitive learning. *Cognitive Science*, 1985, *9*, 75–112.

Selfridge, O. G. Pandemonium: A paradigm for learning. *Proceedings of the Symposium on the Mechanisation of Thought Processes.* Teddington, England: National Physical Laboratory, H.M. Stationary Office, London, 2 vols., 1959.

Smolensky, P. Harmony theory: A mathematical framework for stochastic parallel processing. *Proc. Nat. Conf. on Artificial Intelligence AAAI-83*, Washington, DC, 1983.

Stafford, R. A. Multi-layer learning networks. In Garvey, J. E. (Ed.), *Symposium on self-organizing systems.* Office of Naval Research, 1963.

Sutton, R. S. Temporal aspects of credit assignment in reinforcement learning. Univ. of Massachusetts Ph.D. Dissertation, 1984.

Uhr, L., & Vossler, C. A pattern recognition program that generates, evaluates and adjusts its own operators. *Proc. Western Joint Comp. Conf.*, 1961, 555–569.

Widrow, B. Generalization and information storage in networks of adaline "neurons." In Yovits, M., Jacobi, G., & Goldstein, G. (Eds.), *Self-organizing systems.* Spartan Books, 1962.

Widrow, B., Gupta, N. K., & Maitra, S. Punish/reward: Learning with a critic in adaptive threshold systems. *IEEE Transactions on Systems, Man, and Cybernetics*, 1973, *5*, 455–465.

THE LEARNING OF WORLD MODELS BY CONNECTIONIST NETWORKS*

Richard S. Sutton
GTE Labs, Waltham, MA 02254

Brian Pinette
Department of Computer and Information Science
University of Massachusetts, Amherst, MA 01003

Abstract—Connectionist learning schemes have hitherto seemed far removed from cognitive skills such as reasoning, planning, and the formation of internal models. In this article we investigate what sort of world models a connectionist system might learn and how it might do so. A learning scheme is presented that forms such models based on observed stimulus-stimulus relationships. The basis of the scheme is a recurrently connected network of simple, neuron-like processing elements. The net produces a set of predictions of future stimuli based on the current stimuli, where these predictions are based on a model and involve multiple-step chains of predictions. Results are presented from computer simulations of the scheme connected to a simple world consisting of a stochastic maze (Markov process). By wandering around the maze the network learns its construction. When reinforcement is subsequently introduced, the solution to the maze is learned much more quickly than it is without the "exploration" period. The form and logic of the experiment is the same as that of the *latent learning* experiments of animal learning research.

World Models and Connectionism

Many of the predominant high-level theories of cognition give a central role to internal models (e.g., see Gelertner and Gerstenhaber, 1956; Craik, 1943; Dennett, 1978; Simon, 1969; or Sutton and Barto, 1981, for a review). Their central idea is that much of what we mean by "thought" can be understood as the formation, modification, and use of internal models of the world. Here, by internal models we mean cognitive structures that allow us to predict the future and to anticipate the results of possible actions we might take. In particular, we mean models that allow chains of anticipation and inference, e.g., "if I do A then B will occur, and if B, then C will occur, etc."

In connectionist networks, all concepts (e.g., perceptions of the environment, stimuli and actions) are represented as activity in the elements of the network. In other words, the activity of the network acts as a static model of the current situation. The natural place to model the dynamics of the environment is in the connections between elements: if stimulus or action A is always follows by stimulus B, then a strong connection from A to B models

* This research was supported by the Air Force Office of Scientific Research and the Avionics Laboratory (Air Force Wright Aeronautical Laboratories) through contract F33615-83-C-1078. The authors also wish to thank Andy Barto, Chuck Anderson, John Moore, and Martha Steenstrup for discussing and contributing to these ideas.

this fact in the sense that it causes the evolution of activity in the network to mimic the evolution of states in the external environment. Of course the network's evolution will be much quicker than the environment's, so its predictions will be anticipatory and thereby valuable for directing behavior.

That the behavior of activity within a network might be used to model the behavior of the world is a simply stated idea with many ramifications. The advantages are apparent: besides inheriting the parallelism and mechanistic advantages of connectionism, the theory is also extremely parsimonious, since world modeling is achieved using the same apparatus—the connections of a network—that is used to implement simple stimulus-response mappings. Connectionist internal models could conceivably form a sort of "missing link" between cognitive processes such as reasoning and planning on the one hand, and stimulus-response learning and neuroscience on the other.

Problems with the idea are also apparent. If a network's connections are used to implement the dynamics of an internal model, how can they also be used to implement stimulus-response mappings? Or how can they also be used to implement the structures necessary for recognizing complex concepts and executing complex actions? In addition, models can easily involve self-referential chains, e.g., "A is followed by B which is followed by A again." If a network forms cycles of excitatory connections in order to model such an environment, it courts the danger of positive feedback and instability. The results presented here deal with the self-reference issue, but not with the issue of how to integrate the different functions of a network. We assume that the other functions are completely separate from the network or sub-network that learns and embodies the internal model.

From the point of view of artificial intelligence, where the use of internal models is commonplace, the primary advantage of connectionist models is that they are well-suited for learning. Standard AI techniques include sophisticated methods for searching a model efficiently when the model is completely defined and known. In practice, the model is often only partially known, and search methods for such cases are less well understood. What is needed is the ability both to learn a model and to search it as it is being learned. Connectionist structures seem to be well suited to learning methods, particularly in uncertain and noisy domains. As such they suggest an attractive direction in which to look for ways of learning searchable models.

To limit the scope of this paper, we pay little attention to the issues involved in how a connectionist net should best use a model, and instead concentrate on how it maybe acquired. We also concentrate on capturing inferences that are not dependent on the actions selected, that is, we focus on stimulus-stimulus rather than response-stimulus inferences. A network and learning procedure are presented which can learn to mimic environments, including multiple-step causal chains, as outlined above. The learning procedure, though deterministic, is similar to the stochastic learning procedure used in the Boltzmann machine (Ackley, Hinton, and Sejnowski, 1985) in that it involves letting a recurrent network run to equilibrium in response to two different stimulus patterns.

There are numerous ties between this work and animal learning theory. The learning involved in building an internal model is analogous to the S-S learning effectively advocated by Tolman and others in response to the rise of behaviorism (e.g., see Hilgard and Bower, 1975). The final simulation experiment reported here is directly analogous to their "latent learning" experiments, in which it was shown that animals learn extensively about their environment even if never rewarded or punished, the learning remaining latent and unexpressed in behavior until reinforcement is introduced. The ties between this work and modern animal learning theory are even stronger. Contemporary animal learning theory, particularly that part concerning classical or Pavlovian conditioning, has a very cognitive orientation (e.g., see Dickinson, 1980). Many of its results bear directly on the issues of how a declarative internal model is learned by animals. The learning scheme used here is a descendant of the model of classical conditioning developed by Sutton and Barto (1981), pursuing the network approach illustrated by Moore and Stickney (1980).

Recurrent Networks and Markov Worlds

The network used here is of the simplest of connectionist designs. Each element receives as input the activity of every element and an external input from outside the net. These external stimuli are the variables to be predicted. They correspond to the stimuli A, B, etc., mentioned earlier, while the connections within the net are meant to model the world's dynamics. Accordingly, each connection between elements has a modifiable connection weight, while each external input has a direct and unmodifiable effect on the activity of its corresponding element. The activity of the network is updated according to

$$y' = x + Vy \tag{1}$$

where y is the old activity vector, y' is the new activity vector, x is the external stimulus vector, and V is the matrix of modifiable connection weights. The process given by (1) may converge (reach equilibrium), diverge, or oscillate, depending on the weights. For the weights found by the learning scheme in our simulations, it has always converged.

A perfect stimulus-stimulus model would produce complete information about future stimuli. Given a stimulus x_t at time t, it would be able to reply with $E\{x_{t+k} \mid x_t\}$, for any $k > 0$. Consider the special case of the stimuli x_t being generated by a Markov process with state transition matrix P *, where each x_t is a vector with one component for each state, that component being 1 when the Markov process is in that state and 0 otherwise. Then $E\{x_{t+1} \mid x_t\} = Px_t$ and, in fact, $E\{x_{t+k} \mid x_t\} = P^k x_t$. Recurrent nets are good at computing such powers of matrices: If an autonomous (inputless) recurrent network with connection matrix $V = P$ is initialized to $y = x_t$, then its activity will naturally compute, in sequence, the desired $P^k x_t$.

---

* This is the matrix $[p_{ij}]$, where each p_{ij} is the probability of a transition from state j to state i, given that the process is in state j.

The above is a particularly clear special case, but is of little value in most practical cases. The principal problem is that this approach focuses on a single-step model of the world. It can know that A precedes B only if A *immediately* precedes B, or if distinct stimuli C, D, E, etc., occur for every step intervening between A and B. This approach forces the model to be an extremely local, myopic one, while the major observable regularities inevitably appear at a variety of time scales. Alternatively, we can imagine learning separately the 1-step, 2-step, 3-step, etc., transition probabilities for the environment, and then using each to answer specific questions. However, this approach violates the whole idea of making multiple-step predictions by combining separate predictive steps. In addition, it is not clear that all this information is needed. Do we need to know what is going to happen at each future time step separately, or can some of this information by merged without significant loss? To answer this question we must turn to that of how the predictions are going to be used.

The ultimate use of the model will be to answer questions of the form "how desirable are the consequences of this possible action?" and "now that I've made an action, how well does it appear to have turned out?" One consequence of this is that we will not be so concerned with making predictions of what will happen at specific future times as much as we we will be concerned with the sum of future predictions. Rather than predicting that "A will probably happen at $t+k$," we will want to predict "A will probably happen n times in the near future." Rather than predicting x_{t+k} we wish to predict something more like $\sum_{n=1}^{k} x_{t+n}$. A similar approach, but which avoids an abrupt cutoff at an arbitrary time limit $t+k$, is to predict $\sum_{n=1}^{\infty} \gamma^n x_{t+n}$, where $0 \leq \gamma < 1$. This measure adds up all future stimuli x_{t+k}, weighting each according to γ^k, i.e., the farther in the future a stimulus will occur, the less it contributes to the sum. The value of γ determines how rapidly the weighting of future stimuli drops off.

An estimate of $\sum_{n=1}^{\infty} \gamma^n x_{t+n}$ is also readily computed by a recurrent network if its connection matrix incorporates the environment's probability transition matrix. The equilibrium value for the process given by (1) is $\sum_{n=0}^{\infty} V^n x_t$ (taking $x = x_t$). If V is a fraction γP of the probability transition matrix P of a Markov process, then this sum is $E\left\{\sum_{n=0}^{\infty} \gamma^n x_{t+n}\right\}$, from which the desired quantity, $E\left\{\sum_{n=1}^{\infty} \gamma^n x_{t+n}\right\}$ can be immediately obtained by subtracting x_t. In other words, if V is a fraction of the one-step predictive relationships of the environment, then the network will converge to predictions of how often each stimulus will occur in the near future.

A major advantage of being concerned only with the sum of all future predictions is that one can easily incorporate predictions that span several steps and that cannot be made as chains of one-step predictions. Recall that these are needed when A precedes B with a "gap" of one or more time steps with no distinct stimuli filling the gap. Let us call a prediction "irreducible" if it cannot be reduced to a chain of shorter predictions. All single-step predictions are irreducible predictions, and, as we discussed above, for the special case of Markov environments that reveal their state completely, these are the only irreducible

predictions. In almost any real problem, however, multi-step irreducible predictions are common.

Irreducible predictions have a nice relationship to sums of future predictions. As a generalization of the one-step prediction matrix P, we propose the irreducible prediction matrix $\hat{P}$, whose elements $\hat{p}_{ij}$ represent the irreducible prediction of the ith input from the presence of the jth input. We conjecture that as long as one is interested only in the sum of future predictions one can replace P with $\hat{P}$, i.e., that $\sum_{n=0}^{\infty} P^n x = \sum_{n=0}^{\infty} \hat{P}^n x$, $\forall x$.* If one is discounting, things are slightly more complicated. In order for $\sum_{n=0}^{\infty} \hat{P}^n x$ to match $\sum_{n=0}^{\infty} \gamma^n P^n x$, the irreducible predictions in $\hat{P}$ should be discounted according to their length, i.e., one-step predictions should be discounted by γ, two-step irreducible predictions by γ^2, etc. An example is given below as part of the simulation results.

The Learning Procedure

At each new time step $t+1$ the environment generates a new stimulus vector x_{t+1}. How can this information be used to update the predictions made by the net at time t using x_t? Within each time step the network is run completely to equilibrium (by repeated application of (1)) for both x_t and x_{t+1}; let us denote the equilibrium value for the net's activity in response to x_t as z, and the equilibrium value in response to x_{t+1} as z'. What is the desired relationship between z and z'? Recall that z is supposed to estimate the discounted sum of future x's:

$$z \approx \sum_{n=0}^{\infty} \gamma^n x_{t+n}.$$

Similarly, $z' \approx \sum_{n=0}^{\infty} \gamma^n x_{t+n+1}$, so z's desired value can be rewritten in terms of z' as

$$z \approx x_t + \gamma z'.$$

The difference between z and $x_t + \gamma z'$ can be taken as an error in z and used to update the connection matrix V. According to the learning scheme, each connection weight is incremented proportionally to the product of this difference $(x_t + \gamma z' - z)$ at the post-connection element and the recent external input to the pre-connection element. This change tends to cause z to change so as to reduce the difference. Symbolically, the learning scheme is defined by:

$$\Delta V_t = \beta(x_t + \gamma z' - z)\bar{x}_t^T, \qquad V_0 = 0,$$

where β is a positive learning constant and $\bar{x}_t^T$ denotes the transpose of a vector $\bar{x}_t$ of averages of recent values of x (taking the transpose makes the vector product here an

---

* This should probably be taken as a formal statement of the definition of an irreducible prediction, with the conjecture being that such a matrix $\hat{P}$ exists.

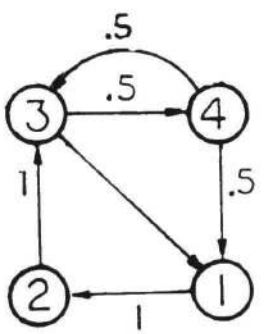
outer product). $\bar{x}_t$ is iteratively computed as follows:

$$\bar{x}_t = \lambda \bar{x}_{t-1} + (1 - \lambda) x_t, \qquad \bar{x}_0 = 0$$

where the parameter λ, $0 \leq \lambda < 1$, determines how strongly previous values of x influence current changes to V.

Simulation Results

Figure 1 shows the state transition structure of a simple autonomous Markov process. The numbers on the arcs indicate the probability that the indicated transition occurs; this information is also given in the state transition matrix P. Using the methods described above, a network successfully learned to model this environment. The stimuli presented to the net were unit basis vectors distinctly identifying the current state: if the environment was in state i at time t, then the ith component of x_t was 1 and all other components were 0. The parameters used were $\gamma = .5$, $\beta = .1$, and $\lambda = .5$. The activity of the network after 200 steps was used as an approximation to its asymptotic behavior. Figure 2 shows a comparison of γP and the V matrix found by the learning algorithm after 1000 steps: The match is quite good; by choosing β smaller and taking more steps we were able to make V approach γP as closely as we desired.



$$P = \begin{pmatrix} 0 & 0 & .5 & .5 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & .5 \\ 0 & 0 & .5 & 0 \end{pmatrix}$$

Figure 1. State transition structure and matrix of a simple autonomous Markov process. Each circle represents a state of the process and each arc a possible state transition. The numbers of the arcs indicate the probability of the state transition.

Next we introduced a "gap" in the environment. The stimulus vector for State 2 was changed to 0, so that nothing could be associated with it. As a result, the two-step predictive relationship between State 1 and State 3 became irreducible. Figure 3 shows the new matrix found. Since stimulus component 2 no longer occurs, nothing has become associated with it. The associations from 1 to 2 and from 2 to 3, which were formerly each of strength γ, have been replaced by a single association from 1 to 3 of strength γ^2. This two-step, direct association from 1 to 3 compensates for the network's inability to form

$$\gamma P = \begin{pmatrix} 0 & 0 & .25 & .25 \\ .5 & 0 & 0 & 0 \\ 0 & .5 & 0 & .25 \\ 0 & 0 & .25 & 0 \end{pmatrix} \qquad V = \begin{pmatrix} .00 & .00 & .31 & .27 \\ .50 & -.01 & .01 & -.01 \\ .00 & .51 & .01 & .25 \\ .00 & -.01 & .17 & -.02 \end{pmatrix}$$

Figure 2. Comparison of γP and the found connection matrix V for the environment shown in Figure 1.

an indirect association from 1 to 3 by way of the now undetectable State 2. As a result, the predictions from all states other than State 2 are exactly the same as they were when State 2 was observable. Figure 4 compares the ideal total prediction matrix $\sum_{n=1}^{\infty} \gamma^n P^n$ with the total prediction matrix $\sum_{n=1}^{\infty} V^n$ found by the network.

$$V = \begin{pmatrix} .00 & 0 & .26 & .26 \\ 0 & 0 & 0 & 0 \\ .25 & 0 & -.00 & .30 \\ .01 & 0 & .25 & -.01 \end{pmatrix}$$

Figure 3. The connection matrix found for the environment shown in Figure 1 when State 2 was a "gap" and could not be the basis for predictions.

Finally, we simulated the larger Markov environment shown in Figure 5, and we introduced a reinforcement learning mechanism to illustrate how a model learned by this method can be used to improve performance on reinforcement learning tasks. At each time step one of two actions, LEFT or RIGHT, was selected by the reinforcement learning system. This selection determines which of the arcs from each state shown in Figure 5 was followed; where a state has only one arc leaving it, the arc was followed independent of the action selected. The experiment had two phases, 1) a pretraining phase, during which no reinforcement was delivered and the environment was randomly explored, and 2) a test phase, during which reinforcement was delivered upon reaching state F. During the test phase, optimal performance is achieved by selecting the RIGHT action from states A, B, C, and D, thereby moving around the maze as rapidly as possible and visiting state F as

$$\sum_{n=1}^{\infty} \gamma^n P^n = \begin{pmatrix} .09 & .18 & .36 & .36 \\ .55 & .09 & .18 & .18 \\ .29 & .58 & .16 & .36 \\ .07 & .15 & .29 & .09 \end{pmatrix} \qquad \sum_{n=1}^{\infty} V^n = \begin{pmatrix} .09 & 0 & .38 & .37 \\ 0 & 0 & 0 & 0 \\ .29 & 0 & .15 & .34 \\ .08 & 0 & .28 & .08 \end{pmatrix}$$

Figure 4. Comparison of ideal and found total prediction matrices for the environment shown in Figure 1 with State 2 a "gap".

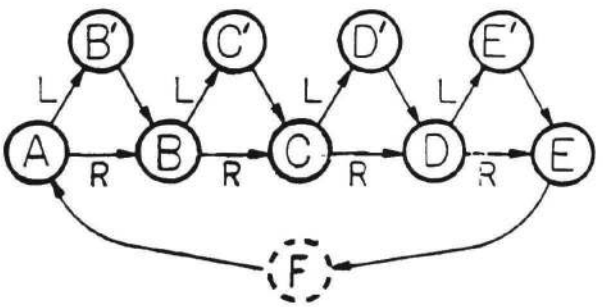


Figure 5. State transition structure of a larger Markov environment. State F is observable and reinforcing during the test phase of the experiment.

frequently as possible.

During the test phase, all states were represented in the usual way: There was one component of the stimulus vector for each state, and if the environment was in that state, then the component was 1, otherwise it was 0. A non-zero 10th component, corresponding to state F, was viewed as reinforcement by the reinforcement learning algorithm. To prevent reinforcement during the pretraining phase, the 10th component of the stimulus vector was held at 0, making state F a "gap" state.

We ran two learning systems on this experiment, called MODEL and NOMODEL. MODEL constructed an internal model during the pretraining phase and then used the model during the test phase by reinforcing its action selections according to changes in the models *predictions* of reinforcement, i.e., according to the 10th component of $x_t + \gamma z' - z$. NOMODEL did not construct a model and reinforced its action selections according to external reinforcement directly, i.e., according to the 10th component of x. Both systems select their actions by maintaining a weight w_i for each state i. When state i is entered, RIGHT is selected with probability $1/(1 + e^{-w_i/T})$, T a positive parameter; otherwise

LEFT is selected. If w_i is zero (as it is initially), then this probability is $1/2$. If w_i is positive then RIGHT will be favored in state i, and if w_i is negative, LEFT will be favored. Whatever action is selected in a state, the probability of repeating that action will be incremented or decremented proportionally to the reinforcement (as defined above for MODEL and NOMODEL) that follows the action selection. The size of the increment or decrement also depends on how closely the reinforcement follows the action selection; it is proportional to λ_w^k, where k is the (possibly zero) number of steps intervening between action and reinforcement. For a more complete discussion of the type of reinforcement learning algorithm employed here see (Sutton, 1984; Barto, Sutton, and Anderson, 1983).

In theory, the use of a model can be a great advantage on this task. The "wrong" actions from states A, B, C, and D all ultimately lead to the goal state, just more slowly than the correct actions. Without a model, a learning system can easily become stuck making a wrong action selection without realizing that it could do better. To avoid this, learning must be very slow. With a model, on the other hand, this problem can be completely avoided. Once reinforcement has been received in state F, a pre-existing model can make good predictions of reinforcement from every state. For example, the model would immediately predict that state C is better (closer to reinforcement) than state C', that state B is better than state B', etc. Armed with this information about the relative desirability of states, in theory a much better job can be done assigning credit and blame to action selections.

Both the pretraining and test phases lasted until the learning system had completed 100 circuits of the environment from state A back to state A. The first reinforcement occurred on the 101st circuit. For the NOMODEL learning system we used the parameters that gave the best performance on this problem. These were T=0.5 and λ_w=0.3. For the MODEL learning system, we chose "reasonable" parameter values without making any attempt to find the best. For the reinforcement-learning part of the system, the parameters used were T=1.0 and λ_w=0 ; for the model-learning part of the system, the parameters used were λ=0 , β=0.01, and γ=0.8. The activity of the network after 10 applications of (1) was used as an approximation to its asymptotic (equilibrium) activity.

We found that the MODEL system did in fact perform much better than the NOMODEL learning system on this task. Figure 6 compares the performance of the MODEL and NOMODEL learning systems on this experiment. Average performance over 10 runs is plotted for each learning system for each circuit of the test phase. The performance measure used for a circuit is the number of steps actually taken in completing the circuit. Note that lower values are better for this performance measure. The learning system without a model clearly had a difficult time with this task. Its performance improved over the duration of the test phase, but remained erratic and far from optimal. The learning system with a model, on the other hand, acheived optimal or near optimal performance after just 3 circuits on all ten runs. The use of a model results in a much higher performance level after fewer experiences with reinforcement on this task.

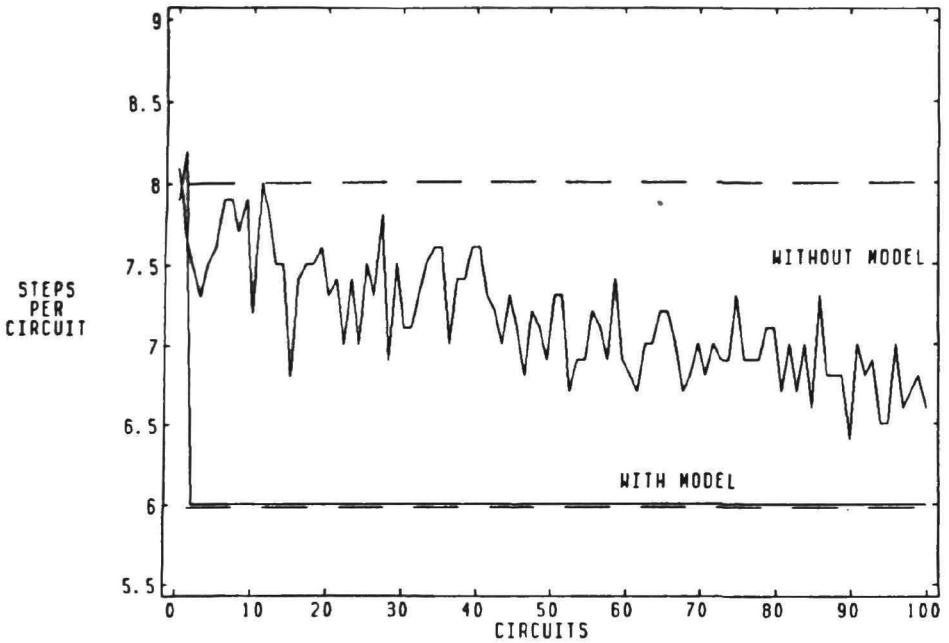


Figure 6. Average learning curves for the MODEL and NOMODEL learning systems during the test phase of the experiment whose environment is shown in Figure 5. A circuit is a complete trip from state A back to state A. The horizontal axis shows circuit number, and the vertical axis shows steps taken to complete the circuit. Note that a lower circuit time corresponds to better performance. The lower dashed line indicates the optimal (fastest) performance level, and the upper dashed line indicates the chance performance level (acheived if RIGHT and LEFT actions are chosen with equal probability from all states).

Conclusions

The modeling and internal simulation of the world is an important part of cognition, and one which the connectionist approach must eventually address. Surprisingly, in certain respects connectionism is very well suited to world modeling. A dynamically evolving network can be used as a world model by linking its activity to states of the world, as is commonly done, and by linking the evolution of that activity to the evolution of world states, as has been done here. The specific method used here to shape that the network's evolution has advantages with regard to the spanning of gaps in stimulus sequences, but is limited to forming stimulus-stimulus as opposed to response-stimulus models. We have tried to make the point that connectionist internal models are feasible, even natural, and to show an example of how they can be used to improve performance on simple learning tasks.

REFERENCES

Ackley, D.H., Hinton, G.H., & Sejnowski, T.J. A learning algorithm for Boltzmann machines. *Cognitive Science*, 1985, *9*, 147–169.

Barto, A. G., Sutton R. S. & Anderson, C. W. Neuronlike elements that can solve difficult learning control problems. *IEEE Trans. on Systems, Man, and Cybernetics SMC-13*, 1983, 834–846.

Craik, K.J.W. *The Nature of Explanation.* Cambridge: Cambridge University Press, 1943.

Dennett, D.C. Why the law of effect will not go away. In: *Brainstorms.* Montgomery, Vermont: Bradford, 1978.

Dickinson, A. *Contemporary Animal Learning Theory.* Cambridge: Cambridge University Press, 1980.

Galanter, E., & Gerstenhaber, M. On thought: The extrinsic theory. *Psychological Review*, 1956, *63*, 218-227.

Hilgard, E.R., & Bower, G.H. *Theories of Learning* (fourth edition). Englewood Cliffs, New Jersey: Prentice-Hall, 1975.

Moore, J.W., & Stickney, K.J. Formation of attentional-associative networks in real time: Role of the hippocampus and implications for conditioning. *Physiological Psychology*, 1980, *8*, 207-217.

Simon, H.A. *The Sciences of the Artificial.* Cambridge, Mass.: The MIT Press, 1969.

Sutton, R. S. Temporal credit assignment in reinforcement learning. Doctoral Thesis, Dept. of Computer and Information Science, Univ. of Massachusetts, Amherst, MA, 1984.

Sutton, R.S., & Barto, A.G. An adaptive network that constructs and uses an internal model of its world. *Cognition and Brain Theory*, 1981a, *3*, 217–246.

Sutton, R.S. & Barto, A.G. Toward a modern theory of adaptive networks: Expectation and prediction. *Psychological Review*, 1981b, *88*, 135–171.

Learning Salience Among Features Through Contingency in the CEL Framework

Richard H. Granger, Jr.
Jeffrey C. Schlimmer

Irvine Computational Intelligence Project
Department of Information and Computer Science
Cognitive Sciences Program
and
Center for the Neurobiology of Learning and Memory
University of California
Irvine, California 92717

Abstract

Determining which features in an environment are salient given a task, *salience assignment*, is a central problem in machine learning. A related phenomenon, *contingency* (the conditions under which relative salience among environmental features is acquired), is central to learning and memory in animal psychology. This paper presents an analysis of a set of empirical data on contingency and an algorithm for the salience assignment problem. The algorithm presented is implemented in a working computer program which interacts with a simulated environment to produce contingent associative learning corresponding to relevant behavioral data. The model also makes specific empirical predictions that can be experimentally tested.

1 Introduction

A rat in a laboratory cage hears a tone. It also hears the air conditioning system start, and sees a lab assistant taking notes. Shortly afterwards, it feels an unpleasant electric shock. The animal's task corresponds to a *salience assignment* problem: which of the many possible cues are the predictive or salient ones, i.e., the ones to be learned? Rats solve the salience assignment problem under constraints more severe than those faced by most AI systems. For instance, learning must be *incremental*, for the rat in a natural setting must make good use of the experiential data already gathered while gathering more. Moreover, the environment is 'noisy' and may not provide perfect predictors; the animal must make predictions as best it can when cues indicate only a change in the probability of an event.

Using the CEL framework (Components of Experiential Learning) [Granger, 1983; Granger and McNulty, 1984], we present a method of determining which features of an event are predictive of others and distinguishing useful cues from context and background noise. The method determines the relevance of individual predefined features, and forms new feature descriptions by conjoining

This research was supported in part by the Office of Naval Research under grant N00014-84-K-0391, the National Science Foundation under grant IST-81-20685, and by the Naval Ocean Systems Center under contract N66001-83-C-0255.

or negating existing features. The effectiveness of the method is due to taking into account, in addition to successful predictions and errors of commission, also errors of omission and events in which the absence of a cue correctly prevented prediction of a second event. This extension to the simpler idea of 'strengthening' and 'weakening' of associations corresponds to the distinction in psychology between learning based on number of pairings and learning based on contingency. Using this method the CAP-CEL (Contingent Associative Processes in CEL) [Granger, Schlimmer and Young, 1985; Granger and Schlimmer, 1985] program exhibits contingency-based learning behavior, modeling the learning behavior of animals and humans in classical conditioning tasks. The program is able to function correctly even with a large number of erroneous training instances.

One of the basic findings of the model is the identification of categories of noise that either will or will not interfere with associative contingent learning. In classical conditioning terms, learning should proceed in the presence of spurious CSs or USs mixed with pairings but should be blocked when *both* spurious CSs and USs occur. Learning in the presence of spurious CSs is equivalent to 'partial reinforcement' in classical conditioning; learning in the presence of spurious USs has not been systematically studied or reported on in the psychological literature. We are currently running animal experiments in our lab at UC Irvine to test the specific experimental predictions that arise from the model.

2 Contingency

2.1 The data

A rat in a classical conditioning task must attempt to decide, over trials, which of several environmental features or events should be learned to be a predictor of a recurring shock event. We can view the animal's task as hypothesizing potential causal relationships between the shock and various features, individually and in various combinations, and weighing these relations against each other.

We might initially assume that a particular feature or event would be inferred to be the predictor of the shock depending on the number of times that feature actually occurred immediately before the shock. Each time the feature is paired with the shock, the 'association' between the feature and the shock might be *strengthened* (see e.g.,[Anderson, 1983]). Extensive experimental evidence in the psychological literature shows this to be false. Over time, a particular feature, say a light, may be paired with the shock more often than is some other feature (e.g., tone). However, this condition is not by itself sufficient to warrant the animal's inference that the light is more likely to predict the occurrence of shock. In particular, even if it happens over a number of trials that the light precedes the shock more often than the tone precedes the shock, say 7 times versus 4 times, *but* the shock also occurs a large number of times without having been preceded by the light (say 8 times) while the shock only rarely occurs without the tone (say 3 times), then an animal will learn to predict that the tone, and not the light, is the better predictor of the occurrence of

shock. Hence, the naïve idea that the number of pairings alone determines the predictiveness (or salience) of candidate predictive features, or that 'strengthening' alone could be the mechanism for learning associations, is false.

This result requires a somewhat counterintuitive computation on the part of the animal: the animal must be computing the probability of the shock occurring given the light and given the tone. Rescorla's [1966, 1967, 1968] formulation of the necessary computation is that the probability of the shock outcome (the US) given the conditional stimulus feature (the CS, e.g., the tone) must be greater than the probability of the outcome occurring without that feature having occurred, or $p(US|CS) > p(US|\overline{CS})$.

This measurement of relative probabilities is referred to as *contingency*; animals, and humans in analogous circumstances, exhibit *contingency-driven learning* in the sense that they somehow maintain incrementally-updated knowledge of the relative predictiveness or *salience* of features. Through experience the animal must pick out the relevant (salient) features from the background of uncorrelated features and use only these salient features to predict future events.

2.2 Required computation

We term the problem described here as *salience assignment*, i.e., the differential assignment of predictive value to candidate predictive features (or combinations of features). All that the animal has available to it as input from the environment is the presence of features sensed over time. What it must compute from these inputs is the relative probability of some features relationships to others over time.

There are four logical categories of these relationships that can be computed: positive predictions, negative predictions, uncorrelated cues, and context; each has a behavioral correlate in animals. First, there are two types of what we term *predictive* cues: positive and negative predictions. Positive predictive cues are those that accurately predict the occurrence of an outcome. Negative cues accurately predict the absence of an outcome (e.g., these 'safety signals' might predict that the shock will not follow the cue, and therefore that the animal need not fear its coming). *Uncorrelated* cues are irrelevant and therefore not necessary for prediction of an event. Finally, an animal cannot readily evaluate the predictive importance of a *context* cue, i.e., one that occurs constantly in the background of a training session. It is impossible to know whether such a cue is a necessary precondition for predicting a shock, unless the shock has been predicted a few times in the absence of the context cue. When this happens, either the context cue will become a positive predictive cue (if the prediction was successful), or an uncorrelated cue (if the prediction failed).

The inputs to be categorized as positive, negative, context or uncorrelated are occurrences of features. For simplicity, we can categorize the logically possible pairwise combinations of two feature events F1 and F2: either F1 occurs and then F2 occurs (*prediction*), F1 occurs and then F2 does not occur (error of *commission*), F1 does not occur and then F2 does occur (error of

omission), or F1 does not occur and neither does F2 (*non-prediction*).[1] The first and last of these combinations strengthen the predictive value, or association, between F1 and F2, while errors of commission and of omission weaken the association.

	F2 present	F2 absent
F1 present	++ Prediction	+− Error of Commission
F1 absent	−+ Error of Omission	−− Non-prediction

Table 1: Possible combinations of F1 and F2

The proposed algorithm for the calculation of contingent predictiveness essentially just keeps a running count of each of these four categories of pairwise events;[2] these counts are used to calculate an estimate of the likelihood that one of these two features predicts the other.

3 An algorithm and implementation for contingent learning

3.1 The basis of the algorithm: sufficiency and necessity

Bayesian statistics (see e.g., [Duda et al., 1979]) provide formulae for the calculation of two values in inductive logic: *Logical Sufficiency (LS)*, which indicates the extent to which the presence of one event predicts, or increases the expectation of, another particular event; and, reciprocally, *Logical Necessity (LN)*, which represents the extent to which the *absence* of an event *decreases* expectation or prediction of the second event. LS and LN are defined to be:

$$\mathrm{LS} = \frac{p(F2|F1)}{p(F2|\overline{F1})} \qquad \mathrm{LN} = \frac{p(\overline{F2}|F1)}{p(\overline{F2}|\overline{F1})}$$

These may be approximated by a pair of simple formulae composed of precisely the four possible categories of pairwise features occurrences given above.

$$\mathrm{LS} = \frac{s(n+o)}{o(s+c)} \qquad \mathrm{LN} = \frac{c(n+o)}{n(s+c)}$$

where s is the count of successful predictions, c is errors of commission, o is errors of omission, and n denotes non-predictions.

[1] Note that we make the simplifying assumption that event occurrences may be described in terms of discrete time and trials. This is a common assumption in the learning literature [Rescorla and Wagner, 1972; Mackintosh, 1975; Pearce and Hall, 1980].

[2] Although a non-prediction will *only* be considered to happen when F2 has been predicted but did not occur. This is because all non-predictions would otherwise give rise to a huge, ongoing number of counts. Hence, in this algorithm, non-predictions are systematically 'undercounted'.

Our proposed algorithm makes use of the calculation of LS and LN values to categorize the relationships between a pair of cues. The categorization is based on the interpretation of LS and LN values. LS values range from 0 to ∞, with high LSs corresponding to a feature F1 strongly predicting a second feature F2, (since high LS implies a high ratio of successes to errors of commission); and very low LSs corresponding to the case where F1 implies that F2 will *not* occur (low ratio of successes to commissions). Hence, for a high LS value, F1 is a positive predictor of F2; for low LS, F1 is a negatively predictive cue, i.e., the presence of F1 predicts that F2 will not occur. An LN value near 1 indicates that the absence of a cue may be ignored, while a low LN value (near zero) indicates that a presence cue is quite necessary for prediction. Note that when LS $>$ 1 it is also true that $p(F2|F1) > p(F2|\overline{F1})$. Furthermore, LN < 1 since $p(\overline{F2}|F1) = 1 - p(F2|F1)$ and $p(\overline{F2}|\overline{F1}) = 1 - p(F2|\overline{F1})$. However, it is not true in general that LS $=$ LN.

When the value of LS is approximately 1, i.e., neither very high nor very low, then the cue F1 may be either a context cue or uncorrelated. In such a case, if there are more errors of commission than of omission, i.e., more failed predictions than unexpected shocks, then the cue is categorized as a context cue since it is often present, it often fails to predict the shock, but the shock doesn't often occur in its absence. If there are more errors of omission than commission, however, then the cue is categorized uncorrelated.

Positive cue	LS $>>$ 1
Negative cue	LS $<<$ 1
Context	LS $\approx$ 1, omissions $<$ commissions
Uncorrelated	LS $\approx$ 1, omissions $\geq$ commissions

3.2 Gathering evidence

All counts in CAP-CEL's memory are initially 1. These counts are updated only when an index node (corresponding to a feature complex) is triggered by matching cues in the environment, at which point one of the relevant long-term memory traces organized below this index node is chosen for reconstruction; i.e., the trace contains predictions of what will happen and which behavior is associated with these predictions.

This trace is matched against new events. When a prediction succeeds, the success scores of matched features in the environment are incremented. Cues failing to match receive incremented omission scores. If a prediction fails, each cue feature that matched the environment scores a commission; each cue feature that was absent from the environment, a non-prediction. Novel features present in the environment are added with an initial score of 1 commission, 1 prediction, 1 omission, and 1 non-prediction.

	++	+−	−+	−−	LS	LN
Cage	52	11	1	1	1.65	0.35
Tone	52	7	1	5	5.29	0.14
Light	52	8	1	4	4.33	0.17
Buzz	19	4	34	8	1.02	0.91
Whirr	43	10	10	2	0.97	1.13
And[Tone,Light]	48	3	1	5	5.65	0.07

Table 2: Positive Contingency

3.3 A detailed example

Assume CAP-CEL is in a situation where tones, lights, noises, and shocks are occurring. CAP-CEL's task is to construct a memory record which will allow it to predict the occurrence of the shock accurately (presumably in order to avoid it). Specifically, given a situation where the shock is reliably preceded by a conjunction of features (e.g., tone and light), a *positive* contingency, a table representing a portion of CAP-CEL's memory about the shock will look similar to table 2. (Note that successes are indicated by '++', commissions by '+−', omissions by '−+', and non-predictions by '−−'. The figures in table 2 are taken from runs of an early version of our computer model.)

The LS (logical sufficiency) value indicates the degree to which a cue is sufficient to cause expectation of a result feature, with values greater than 1 indicating a positive contribution to expectation. The LN (logical necessity) value indicates the degree to which absence of a cue precludes expectation of a result feature. An LN value near one indicates that absence of a cue may be ignored, while an LN value near zero indicates that a cue is quite necessary for expectation. The conjunction of light and tone has been proposed by the CAP-CEL program itself (see discussion in section 3.4).

This chart illustrates important differences between contingency learning and more intuitive notions of strengthening based on number of pairings. Cage and tone receive the same number of pairings with shock, but tone is a much better predictor of shock. Moreover, tone was involved in a greater number of mistaken predictions (errors of commission) than was buzz, but tone is still recognized as the better predictor.

3.4 Combining features

Learning in complex environments necessitates noting useful combinations of features. For instance, a conjunction of a tone and a light may indicate a shock while neither the tone nor the light alone do. CAP-CEL uses current associations between features to form combinations of features.

The introduction of new feature combinations is failure driven. When CAP-CEL makes an

error of commission, a new clause[3] may be introduced from among the predictive clauses. A clause with a low LS value acts as a negative predictor; if satisfied in a negative instance, it is a candidate for negation. A pair of clauses with low LN values act as required positive predictors; if one is satisfied in a negative instance and the other is not, they are candidates for conjunction. We can think of LS and LN values as guidance for a plausible move generator searching through the space of conditions.[4]

Propose	When error of commission
Not[A]	LS(A) << 1, A satisfied
And[A,B]	LN(A) << 1, A satisfied LN(B) << 1, B unsatisfied

CAP-CEL does not expand its representation of clauses without bound. Two mechanisms serve to limit this growth. The first is simply that new clauses are only introduced following a failure. In an environment without erroneous training instances this alone can be quite effective.

Secondly, CAP-CEL utilizes competition between a newly introduced clause and its components. When a new clause is more predictive than its components, the latter are deactivated and are isolated from retrieval processes. CAP-CEL measures the relative predictiveness of clauses by comparing LS and LN values. When a new clause is introduced it is assigned an LS *threshold*[5] set at the maximum of the LS values of the components. The competing component clauses are deactivated when the LS value of the new clause exceeds its threshold. Until this time the new clause cannot be combined with other clauses.

Utilizing this mechanism has the additional advantage that CAP-CEL can correct some erroneous clause formations whether they arise from mistaken actions on the part of the program or are a result of environmental changes. When a clause falls below its threshold, the clauses that led to its formation are reactivated and now compete with the ineffective clause. Each reactivated clause is assigned a threshold that is the counterpart of its rival; if the ineffective clause has an LS threshold, each reactivated clause will have an LN threshold set at the LN value of the ineffective clause. When either reactivated clause surpasses its threshold[6], the ineffective clause is deactivated and the clauses are free to form new combinations. This allows the program to recover from mistakes and follow changes among effective predictors in the environment.

[3] We will use the term *clause* to refer to both individual features and combinations of features in the discussion that follows.

[4] It is desirable to allow both discrimination (through conjunctions of clauses) and generalization (through disjunction). At this time, however, the CAP-CEL model proposes only conjunctions and negations. A weaker form of generalization is achieved by dropping clauses with low predictive value.

[5] An LS threshold is chosen since a more restrictive clause is *falseness preserving* [Michalski, 1983] and thus LN is guaranteed to be as good.

[6] LN thresholds are satisfied if the LN value falls below the threshold.

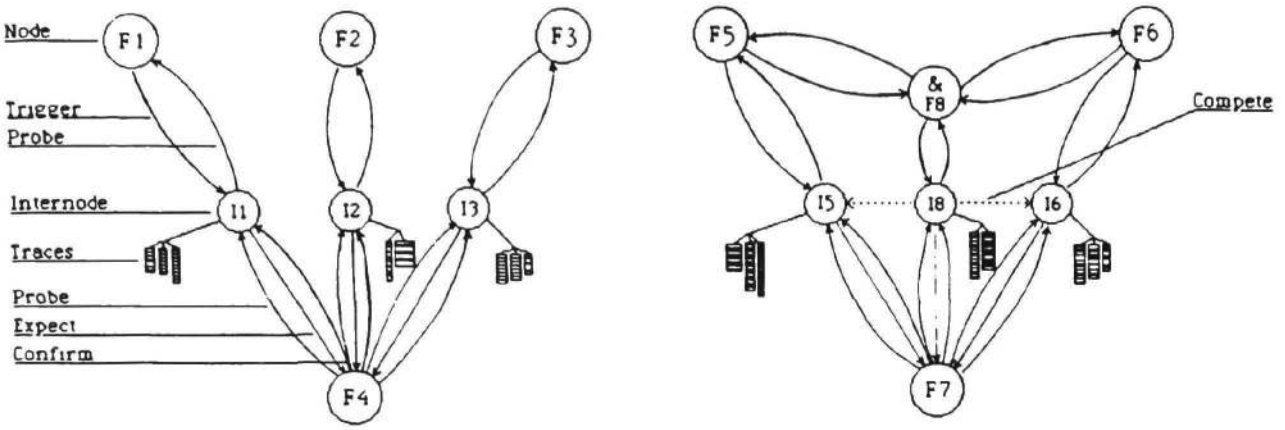


Figure 1: An example memory index structure.

3.5 Organization of indexed memory

The algorithms described here for the incremental calculation of LS and LN and the use of those values to successfully categorize feature cues are grounded in the operation of an indexed network memory. Long-term memory in CAP-CEL is organized as a network of nodes and links accessed via parallel matching processes. Besides providing a degree of parallel processing our model provides a principled method of limiting the spread of activation during retrieval, by *diagnosticity* of tests applied during activation.

Links in our model are simple one-way transmission channels capable of communicating a single non-symbolic value (magnitude) between nodes. These are categorized according to the interpretation of the signal they carry as one of four types of signal: probe links, trigger links, expectation links, or confirmation links. Nodes are of two types: feature or feature combination nodes (henceforth nodes) and intermediate nodes (henceforth internodes) which record relationships between feature nodes.

If a feature F1 is indexed as a cue which may lead to an expectation of feature F4, then an internode will lie between node F1 and node F4, as illustrated in figure 1. Long term memory traces which record occurrences of both F1 and F4 are accessed via the internode between them. Feature F1 may be triggered by incoming sensory data and in turn send a signal to I1, which adjusts the signal strength based on the predictive value of F1 for F4, and passes it on.

Suppose that F1 and F2 are necessary for an expectation of F4, but F3 is not highly correlated with F4. Consider what happens when only F1 is present. Activation from F1 will trigger internode I1 which will scale expectation by LS and pass it to F4 along the expectation link. F4 will then send signals along each of its outgoing probe links. I2 receives a probe and, finding the LN value relating F2 to F4 is quite small (i.e., the presence of F2 is quite necessary for any expectation of F4), I2 sends a probe signal to F2. Since F2 is not present at this time, F2 does not return a signal along a trigger link and I2 sends an inhibitory signal along the expectation link to F4, reducing

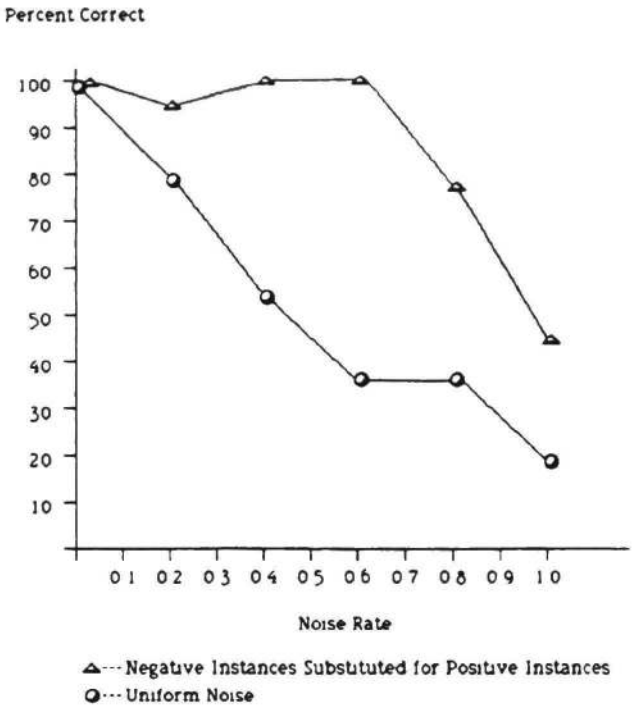


Figure 2: Performance of CAP-CEL as a function of noise.

the expectation of F4. (Values are multiplied together by the receiving node; hence, the LN value which has a value less than one acts as an inhibitory signal.) I3 also receives a probe signal from F4. Since the LN value relating F3 to F4 is about 1 (indicating that F3 is not strongly correlated to F4), I3 does not send a probe signal on to F3 and the spread of activation is attenuated.

Feature nodes may also represent combinations of features. Node F8 is triggered when F5 and F6 are both present, so F8 functions as the boolean AND operator. If F8 is triggered by either F5 or F6, it sends probe signals to both. When it is triggered by both, it triggers I8. Other boolean combinations can be similarly represented. The dashed lines in figure 1 indicate the competition between the recently introduced I8 internode and the component internodes I5 and I6.

The principle advantage of the index network as described above is that *test diagnosticity*, as expressed by LS and LN values stored in internodes, provides natural control for parallel retrieval processes; the spread of activation is limited in a principled way by these tests on links between nodes, rather than by a pursuit strength decay of links over time. The number of nodes needed to represent complex patterns is moderate, and the network can be modified incrementally as new relations between features are discovered.

3.6 Experience with the CAP-CEL system

3.6.1 Robustness

Real-world environments invariably entail some degree of noise, so a learning engine must be able to tolerate erroneous training instances. We have been pleasantly surprised by the performance of the CAP-CEL program in these circumstances. Figure 2 depicts the asymptotic performance of CAP-CEL when trained for a conjunction of tone and light with various rates of spurious trials.

The line plotted with circles in figure 2 shows performance under conditions of 'uniform' noise, that is, an equal number of spurious CS trials and spurious US trials have been substituted for paired and non-paired trials, respectively. As the ratio of spurious CS trials to all trials that include a CS (or spurious US trials to all trials that include a US) approaches 0.3, CAP-CEL's performance falls toward a chance level (50%). As one would expect, spurious trial rates in excess of 0.5 cause CAP-CEL to acquire the opposite association and perform at less than chance level.

The triangles in figure 2 plot the performance of CAP-CEL when there are spurious CS trials but no spurious US trials. This is similar to partial reinforcement in conditioning. CAP-CEL's performance remains well above chance in this case even for levels of noise in excess of 50% because the tone and light are in a positive contingency relation with shock even when the absolute probability of shock following the cues is low. CAP-CEL's level of performance is similar when only spurious US trials are introduced (see section 4 for further discussion).

CAP-CEL's tolerance of erroneous training instances is partly due to the smooth weighting functions LS and LN. In addition, though, we found that robustness depends critically on the introduction of combined features (section 3.4). With no noise in the data, CAP-CEL can achieve perfect performance for simple conjunctive classifications even when the combination proposer is disabled, since the extreme values of LS and LN are sufficient to express logical necessity and logical sufficiency. But when even a small number of erroneous instances are introduced, performance falls off precipitously unless combinations are proposed. The predictive value of a combination of features is higher than that of any of its component features, and the influence of erroneous instances on that value is correspondingly less. Methods which rely on an implicit encoding of feature combinations will not perform as well in the presence of noise as those that use an explicitly combined representation.

3.6.2 Annotated run-time output of the CAP-CEL program

We have implemented CAP-CEL in Franz Lisp on a VAX 11/750 running under Unix. In the following transcript, CAP-CEL learns that the conjunction of tone and light is a positively contingent cue for the onset of shock. Annotations are separated from actual program output by semicolons.

```
                                        :

Receiving: cage, light, whirr, buzz      ; These are external cues.
    strongly expecting shock (odds = 7.81 >> 1).
Receiving: nothing                       ; CAP-CEL predicts the shock,
Updating expectations                    ;   but doesn't get one.
    marking commissions                  ; Satisfied cues get a commission.
    marking non-predictions              ; Unsatisfied cues get a
                                         ;   non-prediction.
Deactivating clause:                     ; This clause isn't proving
    and[whirr,light]                     ;   effective.
Introducing new clause:
    and[light,tone]

        ; The light cue is satisfied (present) in this instance
        ; and has a LN << 1.  The tone cue is not satisfied
        ; (missing) and also has a LN << 1.  Their conjunction
        ; is suggested as a new clause.

                                        :

Receiving: cage, light, tone, whirr
    strongly expecting shock (odds = 31.49 >> 1).
Receiving: shock
Updating Expectations
    marking successes
    marking omissions
Establishing clause:                     ; This clause is now above its
    and[light,tone]                      ;   threshold and its rivals
Deactivating clause:                     ;   are deactivated.
    light
Deactivating clause:
    tone

        ; Long-term memory looks like this now:

shock predicted by:
    ------------------------------------------------
Pattern          ++  +-  -+  --     ls   ||   ln   |
    ------------------------------------------------+
and[light,tone]  8   1   1   3     3.56  ||  0.15  |
    ------------------------------------------------+
buzz             5   2   9   7     1.27  ||  0.65  |
    ------------------------------------------------+
whirr            11  7   3   2     1.02  ||  0.97  |
    ------------------------------------------------+
cage             13  8   1   1     1.24  ||  0.76  |
    ------------------------------------------------
```

3.6.3 Explanation of program behavior

Confidence in CAP-CEL is calculated by multiplying together the LS values of each satisfied feature and the LN values of each unsatisfied feature. This confidence measure is then interpreted

in terms of odds: much less than 1 indicates that F2 is not expected; about 1 indicates uncertainty; much greater than 1 indicates that F2 is expected.

CAP-CEL introduces new clauses only on errors of commission and avoids endlessly making proposals when it reaches proficiency. Possible new clauses are proposed from the satisfied and unsatisfied features on the basis of LS and LN values. In the example above, light has an LN value below well below 1, as does tone, and an error of commission occurs where light is present and tone is absent; CAP-CEL hypothesizes that light *and* tone together might be a better predictor than either alone.

4 An experimental prediction

In review, contingency theory [Rescorla, 1972] states that a necessary condition for the formation of an association between a CS-US pair is $p(US|CS) > p(US|\overline{CS})$. This characterization arose out of Rescorla's original animal experiments [1966, 1967, 1968] where one group of subjects was given random presentations of the CS and random presentations of the US: $p(US|CS) = p(US|\overline{CS})$. A second group received the same random sequence except that the US was not given as scheduled unless it had been preceeded by a CS. Using the terms of section 3.6.1, this group was presented with spurious CS trials substituted into CS-US trials: $p(US|CS) > p(US|\overline{CS})$.

A survey of the relevant literature indicates that animal researchers have failed to test the situation where spurious USs are substituted into CS-US trials. $p(US|CS)$ is greater than $p(US|\overline{CS})$ in this situation as well, so the model predicts that learning should occur. Using the measures of LS and LN, the CAP-CEL program acquires a significant association between the CS-US pairing given spurious US substitutions in a series of pairings. This leads to the prediction that animal subjects in a similar situation also will; an experiment to test this prediction, following Rescorla's design, is currently being run in the authors' lab at UC Irvine.

Rescorla's original characterization would also make this prediction, though his experiments didn't test for it. However, if associational strength is taken to be a function of the magnitude of the inequality $p(US|CS) > p(US|\overline{CS})$, then it may fail to be above some significance threshold. The use of the ratio of the conditional probabilities (LS) and the ratio of their algebraic duals (LN) yields measures which are tolerant of spurious CS substitutions or spurious US substitutions but not both (see figure 2).

5 Related work

Each development and refinement of the CEL framework is driven by an attempt to accurately model experimental data. The model of contingency described in this paper were motivated by results from animal behavior experiments. Animal learning theorists, notably Rescorla and Wagner [1972], Wagner [1981], Mackintosh [1975], and Pearce and Hall [1980], have proposed

models of associative learning which account for contingency. These theories predict the strength of associative learning as a function of experience, but do not describe the detailed processing necessary to form, retrieve, and modify associations. Our model is intended to supply this further level of detail.

Other researchers have also formulated systems for the purpose of modelling experimental data. One of the most comprehensive models of learning behavior is the ACT* family of programs were developed by John Anderson [Anderson, 1983]. ACT* uses productions systems as a framework for describing the processing underlying complex behavior. ACT* creates new production rules through processes of composition, proceduralization, generalization, and discrimination. Of these, generalization and discrimination address the problem of discovering which features are relevant for determining when an operator should be applied. ACT* creates a generalized rule by omitting a condition from the antecedent part of another rule. Discrimination adds a new clause either to the antecedent or to the consequent part of a rule. Rules in ACT* are strengthened when they are reinvented and when they are activated through the spread of activation in memory. They are weakened by negative feedback.

The ACT* framework has been used to account for a wide variety of data from the psychology of human learning. The scheme for strengthening and weakening rules, however, does not appear to be consistent with the basic psychological data concerning contingency.

Another cognitively oriented artificial intelligence program employs learning while parsing English stories: IPP [Lebowitz, 1983]. Stories read by IPP are used to form groups of story features that frequently occur together. These groups of features provide top-down direction for IPP's parsing mechanism. The number of features in common between two groups of features is used to determine when a new feature group should be introduced. These feature groups are strengthened and weakened by a unary amount given positive or negative evidence. Individual features within a group that increase expectation of other features in the same group are termed *predictive*; those that do not are termed *predictable*. The degree to which a feature is predictive is based on how infrequently it has occurred in feature groups.

IPP successfully improves its parsing of texts within the domain of newspaper stories. However, the mechanisms for determining predictiveness do not appear to be consistent with the findings of contingency experiments. Secondly, as Lebowitz notes, IPP has some difficulty when a set of features in a group are predictive while any single one isn't. Lacking a representation for explicit conjunctions prohibits IPP from assigning predictiveness to only a subset of features.

Researchers in cognitive psychology have also outlined constraints for mechanisms that attempt to account for experimental results. Barsalou and Bower have expressed three concerns over the formulation of memory models: elimination of test contingency, parallel traversal, and storage requirements [1984]. The first is the characteristic of some memory models that allows or inhibits retrieval of items stored in a network based on the success of some distantly related test. Barsalou and Bower argue that partial matching should be allowed in memory retrieval and that there

should be a meaning related ordering of tests if any. Secondly, experimental evidence that human memory is parallel have lead Barsalou and Bower to criticize inherently serial models of memory. Thirdly, Barsalou and Bower doubt the validity of memory schemes that require exponential memory space, for human memory systems do not appear to be limited in the ways predicted by such schemes.

CAP-CEL addresses each of these concerns in that it employs a memory scheme which allows partial matching (long-term memory traces are retrievable by any of the predictive cues in that trace), a parallel retrieval scheme, and a memory containment scheme which displays a growth bounded on the average by n^2 (where n is the number of novel stimuli in the environment) since there are at most n^2 pairwise associations. In the most pathological case, explicit representation of boolean combinations requires a bound of 2^n.

6 Conclusions

The importance of contingency is well known in animal learning theory, and the extensive experimental data concerning contingency provide a clear set of computational requirements for a process model of learning. We have expressed this set of requirements algorithmically as a *salience assignment* problem, and we have shown how this problem is solved within the CEL framework, via formulae based on Bayes' algorithm. The CAP-CEL program demonstrates that the account we have offered is in fact adequate to distinguish useful, predictive cues from context and uncorrelated cues. Moreover, the performance of CAP-CEL comprises a set of detailed predictions of our hypotheses that may be confirmed or rejected on the basis of experiments.

There is a wealth of empirical behavioral data waiting to be accounted for. For instance, there is an interesting body of data concerning the response latencies of classically and instrumentally conditioned subjects. We have begun to address these issues and hope to integrate our findings with previous results [Granger, Schlimmer and Young, 1985]. We intend to continue to concentrate on basic phenomena of animal learning rather than following the current artificial intelligence and cognitive science fashion of building computer models of complex human problem solving tasks; we believe these basic phenomena shed more light on the fundamental properties of learning in humans as well as animals.

7 References

Anderson, J.R. *The Architecture of Cognition*. Harvard University Press, Cambridge, Massachusetts, 1983.

Barsalou, L. and Bower, G. Discrimination nets as psychological models. *Cognitive Science*, **8**, 1–26, 1984.

Duda, R.O., Gaschnig, J.G., Hart, P. Model design in the Prospector consultant system for mineral exploration. In *Expert Systems in the Micro-electronic Age*. Michie, D. (Ed.), Edinburgh University Press, Edinburgh, 1979.

Granger, R.H. Identification of components of episodic learning: The CEL process model of early learning and memory. *Cognition and Brain Theory*, **6**, 5–32, 1983.

Granger, R.H. and McNulty, D.M. Learning and memory in machines and animals: An AI model that accounts for some neurobiological data. *Proceedings of the Sixth Annual Conference of the Cognitive Science Society*, Boulder, Colorado, 1984.

Granger, R.H., Schlimmer, J.S., and Young, M.T. *Contingency and latency in associative learning: Computational, algorithmic and implementation analyses*. In *Brain Structures, Learning and Memory*. Davis, J., Wegman, E. and Newburg, R. (Eds.), 1985 (to appear). Also appears as: Computer Science Department Technical Report #85-10. University of California, Irvine.

Granger, R.H. and Schlimmer, J.S. Combining numeric and symbolic learning techniques. *Proceedings of the Third International Machine Learning Workshop*, 1985.

Lebowitz, M. Generalization from natural language text. *Cognitive Science*, **7**, 1–40, 1983.

Mackintosh, N.L. A theory of attention: Variations in the associability of stimulus with reinforcement. *Psychological Review*, **82**, 276–298, 1975.

Michalski, R.S. A theory and methodology of inductive learning. *Artificial Intelligence*, **20**, 111–161, 1983.

Pearce, J.M. and Hall, G. A model for Pavlovian learning: variations in the effectiveness of conditioned but not of unconditioned stimuli. *Psychological Review*, **87**, 532–552, 1980.

Rescorla, R. Predictability and number of pairings in Pavlovian fear conditioning. *Psychonomic Science*, **4**, 383–384, 1966.

Rescorla, R. Pavlovian conditioning and its proper control procedures. *Psychological Review*, **74**, 71–80, 1977.

Rescorla, R. Probability of shock in the presence and absence of CS in fear conditioning. *J. Comparative and Physiological Psychology*, **66**, 1–5, 1968.

Rescorla, R. and Wagner, A.R. A theory of Pavlovian conditioning: Variations in the effectiveness of reinforcement and non-reinforcement. In *Classical Conditioning II: Current Research and Theory*. Black, A.H. and Prokasy, W.F. (Eds.), Appleton-Century-Crofts, New York, 1972.

Wagner, A.R. SOP: A model of automatic memory processing in animal behavior. In *Information processing in animals: memory mechanisms*. Spear, N.E. and Miller, R.R. (Eds.), Erlbaum, Hillsdale, New Jersey, 5–47, 1981.

COMPONENT MODELS OF PHYSICAL SYSTEMS

Allan Collins
Bolt Beranek & Newman, Inc.

In order to get around in the world people have to make sense of Coke machines and computers, home heating systems and electric circuits, and even evaporation processes and bouncing balls. They must build their own folk models of how these things behave e.g., what will happen if they put different amounts of money in a Coke machine, what to do if the machine doesn't behave as they expect. "Mental Models" is the term that has evolved for a new view of how people conceptualize physical systems. Mental models are meant to imply a conceptual representation that is qualitative, and that you can run in your mind's eye and see what happens.

In order to make the discussion concrete, I will display some mental models for your inspection. I've picked three domains – electricity, home heating systems, and evaporation, – because they cover the diversity of different mental models, and because it is possible to illustrate a variety of hypotheses about mental models in terms of these domains.

Mental Models of Electric Circuits

A number of investigators, have studied people's naive models of electric circuits (Fredette & Lochhead, 1980; Gentner & Gentner, 1983; Osbourne 1981; Osbourne & Wittrock, 1983; Steinberg, 1983). Since my taxonomy is more differentiated then the others, I will describe the naive models I found and point out how these relate to the mental models of electricity found by other researchers.

In the study I analyzed people's naive models of a simple battery, switch, and lightbulb circuit. For the circuit, I administered a questionnaire to twenty-four subjects who had no special knowledge of circuits. The questionnaire asked them (1) to draw a schematic circuit and explain how it worked, (2) to answer a variety of questions about the working of the circuit that they drew, such as which way the current flows in each component and wire, and (3) to evaluate different circuits in order to decide whether they would work properly.

I was able to categorize the types of models people were using both at the circuit level and at the individual component level. Seven of the subjects had a more or less correct model of the way a battery circuit functions. That is, one pole of the battery acts as a source of electrons and the other as a sink. The current is maintained by an ion flow in the battery between the two poles. I explain each of the incorrect models below:

(1) Converging flow model (3 subjects). This model posits that there are two kinds of electricity (i.e., positive and negative) flowing out from the battery, and that you need both kinds to make the bulb light. I have called this model the "epoxy glue" model of circuit flow, while Steinberg (1983) refers to it as the "sink" model and Osbourne (1981) the "bipolar" model. Like me, Steinberg finds this model held by approximately one-seventh of Smith College undergraduates first taking physics, but Osbourne and Wittrock (1983) report finding this model in approximately one-third of young children.

(2) Circular flow model (4 subjects). In this view electrons flow around and

around the circuit, and each time that they pass through the battery, they are given additional momentum. This view is incorrect in that it assumes that individual particles go through the battery much like water particles go through a pump in a circulatory water system. I have called this the "racetrack" model, and it is a version of the "moving crowd" model discussed by Gentner & Gentner (1983).

(3) Impulse-signal model (3 subjects). This model posits that the switch sends an impulse to the battery to trigger current flow from the battery to the light bulb. It views a switch like the trigger of a gun. This is a variation on the "consumer" model described by Steinberg (1983) and Fredette & Lochhead (1980), and the unipolar model described by Osbourne (1981).

(4) Gate-switch model (3 subjects). This model posits that current flows only from the battery to the light bulb. For the switch to work properly, it must be inserted between the battery and the bulb, where it acts as a gate regulating current flow. This too is a version of the unipolar model found by Osbourne, and the consumer model of Steinberg (1983) and Fredette & Lochhead (1980). It seems to derive from the water-flow analogy described by Gentner and Gentner (1983), where the switch is viewed as a kind of valve regulating flow.

(5) Gate-switch model with circuit (2 subjects). This is a variant of the above model. It posits that current flows in a circuit back to the battery, but you need the switch between the battery and the bulb to keep the current from reaching the bulb when the switch is off. Thus, it is like water flowing in a circulating system; if the water can reach a small hole in the pipe (analogous to the light), then water will trickle out (i.e., the bulb will light). This is a partial consumer model of the bulb, where most of the current is assumed to flow back to the battery.

(6) Controller-switch model (2 subjects). This model views the switch as a control device that is hooked up to the battery-bulb circuit by its own control circuit. It was the least coherent of the six incorrect views I encountered and the two subjects who proposed it seemed not to believe it very strongly. None of the previous studies talk about any such view.

I also attempted to analyze the mental models subjects had of the three major components in the circuit. Here I made my best guess on the basis of their answers to the questions as to each subject's view of each component.

Based on the analysis there were four different views of the battery. Ten subjects had a source-sink view of the battery, with electrons flowing from one pole to the other in the circuit. Seven subjects viewed the battery as simply a source of electricity which flowed out to the bulb. Four subjects considered the battery to be an energizer, like a pump in a water circulation system, and three subjects considered it a source of both positive and negative electricity.

There were three prevailing views of the light bulb that I could distinguish. Twelve subjects viewed it as a narrow passage or resistor through which current must pass. Ten subjects viewed it as a consumer of electric current, three of these requiring both positive and negative electricity to come together to be consumed. The last two subjects had what I call the partial consumer view where a little bit of the electricity is burned up as it goes through the bulb.

We distinguished four views of the switch. Most subjects had the correct

contact–gate view, where the switch is viewed as a swinging gate attached to one wire and making contact (or not) with the other wire. The incorrect views included the impulse– signal view that I likened to a gun's trigger, a blocking notion like a valve in a pipe, and a control–device notion like the handle on a faucet that is mechanically linked to the valve in the pipe (though both subjects had the control device linked electrically to the circuit).

What these models illustrate is that naive subjects have a diverse set of views on circuits, which are more or less coherent. These views can be used generatively to guess how circuits they have not encountered before will behave, as we will discuss later. Furthermore, these views are structured: as we have shown here, they can be regarded as models of the entire circuit or more locally as views of specific components of the circuit. Different component views can in fact go with different system views. For example, one could have an energizer view of the battery with either a contact gate, control device, or blocking view of a switch and either a resistor or partial consumer view of a light. Thus, mental models of components can be regarded as building blocks for a global view.

Mental Models of Home Heating Systems

Kempton (in press) carried out a series of forty–two, in depth interviews with average Americans to determine how they set their thermostats and why. Based on his interviews he identified two common folk theories of how thermostats work, which lead to quite different patterns of setting a thermostat. The two theories he identified are the Feedback Theory and the Valve Theory.

The Feedback Theory. This theory views the thermostat as a device that senses the temperature in the room, and if it falls below the temperature setting, the furnace turns on, and if it rises above the temperature setting, the furnace turns off. Actually, a thermostat has two set points equidistant from the thermostat setting; the lower of which turns on the furnace and the higher of which turns it off. Kempton quotes one of his subjects who held this view.

"You just turn the thermostat up, and once she gets up there (to the desired temperature) she'll kick off automatically. And then she'll kick on and off to keep it at that temperature."

The Valve Theory. This view holds that the setting on the thermostat controls the rate of heat flow, so that the higher you set the thermostat, the harder the furnace works to produce heat. He quotes a subject who held this view:

"Um, I assume, um, that there is some kind of linear relationship between where the lever is and the way some kind of heat generating system functions. And, um, that it's like stepping on the gas pedal; that there I have a notion of hydraulics, you know, the harder you push there is, the more fluid gets pushed into the engine, and the more explosions there are, and the faster it goes."

There are hints in the data Kempton quotes to indicate that the dichotomy he makes between the two theories really should be thought of as two points in a larger space of possible models. We can see this space of models best in two examples he gives of people who do not quite fit the valve/feedback dichotomy.

One informant quite clearly enunciated the feedback theory at first:

"I guess, what I always thought was when you turn the, the temperature, you turn the thermostat to 65, the furnace works to keep the room at 65 and then as soon as it's 65, the furnace stops working and then when it starts to get a little bit cold again the furnace will work again."

But when Kempton questioned her about heating up the house, she verbalized the valve theory:

Q: Let's say it's very cold ... you come into the house and it's very cold, and you want to heat the house up. Let's say you want to heat the house up to 65. What would you do ...

R: If it's very, very cold?

Q: Uh-huh.

R: I might turn it up to 70, for maybe 20 minutes, half an hour and then turn it back down to 65 to see if I can get it warmer faster

Let me consider how such a hybrid model might work. The essence of the valve theory is that the amount of heat flow is proportional to the temperature: this is a proposition about a factor I will be able rate of heat flow. The essence of the feedback theory is that the heat turns on when the temperature falls below a set point, and turns off when the temperature rises above another set point: this is a proposition about a factor I will call type of control. Because these two theories address different factors, it is perfectly possible to believe in variable heat flow, as in the valve theory, and indirect control, as in the feedback theory. The hybrid theory is simply that the temperature setting controls both factors: the higher the setting, the greater the heat flow, and the higher the set points at which the furnace turns on and off.

Another of Kempton's protocols points up a third factor, heat loss, that affects the way people set their thermostats. In particular, a woman respondent believed in setting the thermostat back at night, but her husband reasoned from his feedback theory that it did not pay to do so:

"Now, my husband disagrees with me. He, he feels, and he will argue with me long enough, that I do not save any fuel by turning the thermostat up and down.... Because he, he feels that by the time you turn it down to 55 and all the objects in the house drop to 55°, you're going to use more fuel than if you would have left it at 65 and it just kicks in now and then."

The essence of the husband's argument is that the time the furnace is on when you move the thermostat up in the morning is as great or greater than the time it is off when you set the thermostat down at night. The two times are in fact roughly equivalent. The savings from turning the thermostat down at night come because the furnace turns on less frequently during the night. But it is easy to see how her husband might not think of this savings, if he thinks the only effect of lowering the setting is the two transients which offset each other. He is implicitly assuming that

the furnace is turning on and off at the same rate at a lower setting as a higher setting, once steady-state is achieved. Such a view makes sense if you think heat loss from a house is constant, or simply depends on the temperature outside. In order to really "understand" why there is a savings, you have to have a notion that heat loss is proportional to the difference between the temperature indoors and outdoors, or at least that it depends on the indoor temperature.

We have thus identified three factors that appear in Kempton's protocols, for which people may have different models. These factors are equivalent to the component models I discussed with respect to electricity. I would argue that most combinations of the different models for these three components are possible.

Let me explain each of the possible models and their combinations. With respect to the "heat flow" component, the graded heat flow model is what the valve theory implies. The constant heat flow model, which is implied by the feedback theory, is the correct model (ignoring transients). With respect to the "type of control" component, direct control, which the valve theory implies, means that the rate of heat flow changes as you move the thermostat up or down. Indirect control by temperature, as the feedback theory implies, refers to the model where changes in temperature cause the furnace to turn on and off. A third possible model is an adjustable ratio model: this model assumes that the furnace turns itself on and off with a particular frequency, and changing the temperature setting causes the furnace to turn on and off at a different ratio. This might be called an open-loop model as opposed to a feedback model. Finally, with respect to heat loss, people might have any of the three models I described in the discussion of the husband who wouldn't turn the heat back at night.

In summary, I see there are two dominant views of how a thermostat works and that these influence people's behavior in substantive ways. But in addition to these two dominant views, at a finer-grain level of analysis, there are a host of possible models. These derive from combining component models in various ways.

Mental Models of Evaporation

In a study where they posed eight difficult questions about evaporation to four novice subjects, Collins and Gentner (in press) found that subjects' models of evaporation consisted of five component subprocesses:

1) How molecules behave in the water
2) How molecules escape from the water to the air
3) How molecules behave in the air
4) How molecules return to the water from the air
5) How molecules go from liquid to vapor, and vice versa

They enumerated a number of different views for each of these component processes, that were suggested in subjects' protocols.

Behavior in the water. Figure 1 shows four models for how molecules behave in water. The first view is called the sand-grain model - the molecules just sit there like grains of sand, moving and slipping when something pushes on them. The temperature of the water is the average temperature of the individual molecules. This is a very primitive model. The next two views assume that the molecules are bouncing around in the water like billiard balls in random directions. In both these views, the speed of the molecules reflects the temperature of the water. The difference is that

in one version -- the equal speed model -- all the molecules are moving at the same speed. The other version is a random speed model, that allows for differences in speed for different particles. On this view temperature reflects the average speed of a collection of molecules. The fourth view, called the molecular attraction model, incorporates attraction between molecules into the random speed model. In it molecules move around randomly, but their paths are constrained by the attractive (and repulsive) electrical forces between molecules. This view is essentially correct.

Escape from the water. Figure 2 shows three possible component models for escape (pictorially two are the same). The heat-threshold model is a threshold view of escape: The molecules have to reach some temperature, such as the boiling point of the liquid, and then they pop out of the liquid, the way popcorn pops out of the pan when it is hot enough. The remaining two models focus on molecular velocity, rather than the incorrect notion of molecular temperature. The rocketship model is based on the assumption that the molecules in the water are moving in random directions. In order to escape from the water (like a rocketship from the earth), a molecule must have an initial velocity in the vertical direction sufficient to escape from gravity. The third view, the molecular escape model, posits that the initial velocity must be great enough to escape from the molecular attraction of the other molecules. Both these latter models are in part correct, but the major effect is due to the molecular attraction of the water.

Behavior in the air. There are three component models of how the water molecules behave in the air are depicted in Figure 3. The container model posits that the air holds water molecules and air molecules mixed together until it is filled up (at 100% humidity). The variable-size-room model is a refinement of the container model to account for the fact that warm air holds more moisture than cold air. In this model, molecules in warm air are further apart, and so are less dense than molecules in cold air.
That leaves more space to put water molecules in warm air than in cold air. In the exchange-of-energy model, the chief reason that cold air holds less moisture than warm air is that its air molecules are less energetic. When water molecules in the air collide with air molecules, they are more likely to give up energy if the air is cold (and hence less energetic) than if it is warm. If the water molecules become less energetic, they are more easily captured by the molecular attraction of other water molecules (or a nucleus particle). When enough water molecules collect around a particle, they will precipitate. This latter view is essentially correct.

Return to the water. Figure 4 shows three models of how water molecules return to the water. The crowded room model assumes that when all the space in the air is filled, no more water molecules can get in. The aggregation model assumes that water molecules move around in the air until they encounter a nucleus or particle (which could be another water molecule) around which water accumulates. The less energetic the molecule, the more likely it is to be caught by the molecular attraction of the particle. As these particles accumulate water, gravitational forces overcome the random movement of the particles and they precipitate. The recapture model assumes that particles are attracted by the surface of the water (or other surfaces). The less energy they have, the more likely they are to be recaptured. The action in this view takes place near the surface, unlike the aggregation view. Both the aggregation and the recapture models are essentially correct, but the aggregation model takes place over a long time period with relatively high humidities, whereas the recapture model is applicable in any situation where evaporation is occurring.

Liquid-vapor transition. Figure 5 shows four different views for the transition

from liquid to vapor and from vapor to liquid. One view, the coterminus model, is that the transition occurs when the molecules leave the water and escape to the air, and vice versa. On this view the two transitions, between water and air, and between liquid and vapor, are the same transition. In other words, whether a molecule is in the vapor or liquid state depends solely on location. all molecules beneath the surface of the water are liquid, and all molecules above the surface of the water are vapor. A second view, the intrinsic state model, treats the liquid or gas state as an intrinsic property of the molecule. If the molecule becomes hot enough, it changes from liquid to vapor, and if it becomes cold enough it changes from vapor to liquid. Location is correlated with state, in that molecules in the vapor state tend to move into the air, while molecules in the liquid state remain in the water. A third view, the disassembly model, is based on a little chemistry: in it liquid water is thought of as made up of molecules of H_2O, whereas the hydrogen and oxygen are thought to be separated in water vapor. The expert view, called the binding model, is based on molecular attraction: water molecules in the liquid state are partially bound together by electrical attraction of the neighboring molecules, whereas molecules in the gaseous state bounce around rather freely. The bubbles in a boiling pan of water are thus water molecules that have broken free of each other to create a small volume of water vapor, and clouds and mist are microscopic droplets of liquid water that have condensed, but are suspended in the air.

Combining component models. Table 3 summarizes all the component models described above. Subjects can combine these component models in different ways. Collins and Gentner (in press) show answers from two subjects who had different combinations of these component models. One subject had a model constructed from the random speed model of water, the rocketship and molecular escape models of escape, the variable-size-room model of the air, the crowded room model of return, and the coterminus model of the liquid-vapor transition. The other subject had a less, consistent and less stable model of evaporation. His view included something like the heat-threshold model of escape, the container model of the air, the recapture model of return, and the intrinsic state model of the liquid-vapor transition. In contrast, as I have indicated, the expert view is made up of the molecular attraction model of water, the rocketship and molecular escape models of escape, the exchange-of-energy model of the air, the aggregation and recapture models of return, and the binding model of the liquid-vapor transition.

Summary

There have been a variety of attempts to identify naive subjects' mental models, but most of these have settled on two or three global views (e.g. Kempton, 1984; Osbourne, 1981; Steinberg, 1983). At a finer grain level of analysis, however, it is possible to identify a number of different components and a variety of mental models for each component. These component models can be combined in many different ways. But frequently two or three combinations predominate, giving rise to the global mental models identified by different researchers.

The componential approach has been pursued by a number of researchers in artificial intelligence (de Kleer 1979; de Kleer & Brown 1981, 1983; Forbus 1981, 1982; Hayes 1984). But none of these researchers had tried to analyze the component models human subjects actually have. What I have tried to show is how a componential analysis is necessary to characterize the way people understand physical systems.

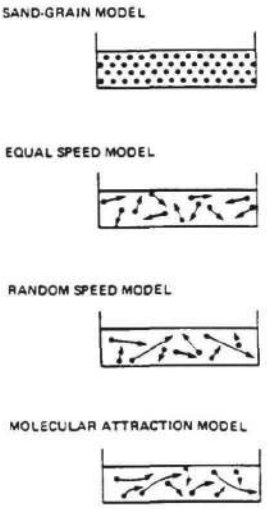


Figure 1. Component models of the behavior of molecules in water

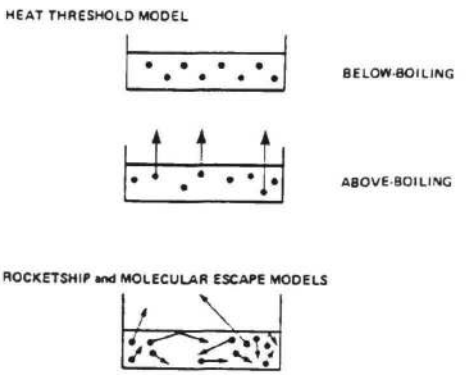
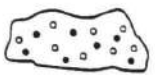


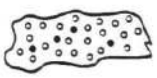
Figure 2. Component models of how water molecules escape from water to air

CONTAINER MODEL

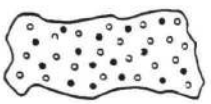


VARIABLE SIZE ROOM MODEL

COLD AIR

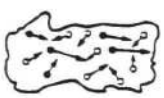


WARM AIR



EXCHANGE-OF-ENERGY MODEL

COLD AIR



WARM AIR

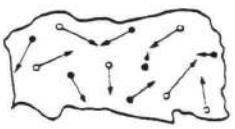
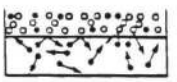
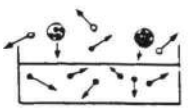


Figure 3. Component models of the behavior of water molecules in air (water molecules are filled circles; air molecules are open circles)

CROWDED ROOM MODEL



AGGREGATION MODEL



RECAPTURE MODEL

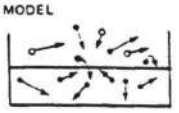


Figure 4. Component models of how water molecules return from air to water

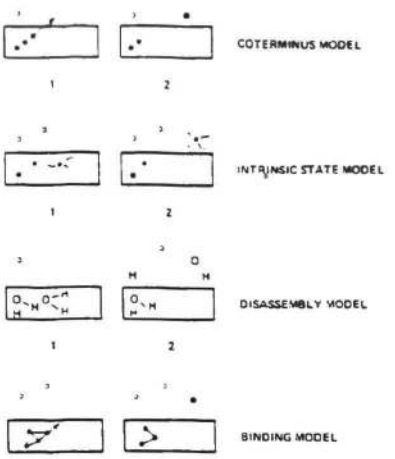


Figure 5. Component models of the liquid-vapor transition

References

Collins, A. & Gentner, D. How people construct mental models. In N. Quinn and D. Holland (Eds.), Cultural models in language and thought, in press.

de Kleer, J. The origin and resolution of ambiguities in causal arguments. Proceedings of the Sixth International Joint Conference on Artificial Intelligence, Tokyo, Japan, 1979, 197–203.

de Kleer, J., & Brown, J.S. Mental models of physical mechanisms and their acquisition. In J.R. Anderson (Ed.), Cognitive skills and their acquisition. Hillsdale, NJ: Erlbaum, 1981.

de Kleer, J., & Brown, J.S. Assumptions and ambiguities in mechanistic mental models. In D. Gentner & A.L. Stevens (Eds.), Mental models. Hillsdale, NJ: Erlbaum, 1983, 7, 121–153.

Forbus, K. Qualitative reasoning about physical processes. Proceedings of IJCAI-7. 1981.

Forbus, K.D. Qualitative process theory (A.I.M. 664). Artificial Intelligence Laboratory. Massachusetts Institute of Technology, February 1982.

Fredette, N., & Lochhead, J. Student conceptions of simple circuits. The Physics Teacher, 1980, 18, 194ff.

Gentner, D. & Gentner, D.R. Flowing waters or teaming crowds: Mental models of electricity. In D. Gentner & A.L. Stevens (Eds.), Mental models. Hillsdale, NJ: Erlbaum, 1983.

Hayes, P. Ontology for liquids. In J. Hobbs and R. Moore (Eds.), Formal theories of the common sense world. Norwood, NJ: Ablex, 1984.

Kempton, W. Two theories used for homes heat control. In N. Quinn and D. Holland (Eds.), Cultural models in language and thought, in press.

Osborne, R. Children's ideas about electric current. New Zealand Science Teacher, 1981, 29, 12–19.

Osborne, R.J., & Wittrock, M.C. Learning Science: A generative process. Science Education, in press.

Steinberg, M.S. Reinventing electricity. In Proceedings of the International Seminar, Misconceptions in Science and Mathematics, Ithaca, New York, June 1983.

Temporal Notation and Causal Terminology

Yoav Shoham and Thomas Dean
Yale University
Department of Computer Science

ABSTRACT

We argue that causal reasoning is an essential part of intelligent human behavior, and that discussion of it cannot be divorced from discussion of temporal reasoning. We therefore set out to define causation in three stages. In the first, we present an ontology of time. We then outline a theory of "causal conditionals", which allows one to reason about multiple possible courses of events. Finally, we define causation in terms of *direct causation* and *causal origins*.

1 Introduction

Philosopher's have long disputed the relative merits of one causal theory over another. Some even question whether or not the notion is a useful one at all, suggesting that causality is simply an anachronism that "scientific" man would well be rid of.

- Nothing exists from whose nature some effect does not follow. (Spinoza)
- I assert that nothing ever comes to pass without a cause. (Jonathan Edwards)
- Causality is *to us* the cement of the universe. (David Hume)
- The Law of Causality, I believe, like much that passes among philosophers, is a relic of a bygone age, surviving like the monarchy, only it is erroneously supposed to do no harm ... All philosophers, of every school, imagine that causation is one of the fundamental axioms of science, yet, oddly enough, in advanced science ... the word "cause" never occurs. (Bertrand Russell)

Whether or not causality is respectable in the scientific literature, it is clear that people employ causal terminology in their day to day communication and, we will assume, in their understanding of the world around them. Consider the following excerpts from recent issues of Newsweek magazine (the italics are ours).

- March 4, in an article on the dangers of DDT, the chemical is described as "a pesticide that *silenced* birds, *threatened* West Coast peregrine falcons and bald eagles, and possibly *caused* cancer in humans".
- March 11, on Page 3, it reads: "Newly disclosed evidence has *reignited* the controversy over Bernard Hugo Goetz ... mounting furor seemed to be *leading towards* a new grand jury investigation ..."
- In the same issue, in a letter to the editor beginning on page 4, one reads: " ... tough laws *will not stop* the use of dangerous drugs but will instead *lead to* obscene profits ..."
- and on page 60 an article is titled "A strike cripples Pan Am".

The italics in the quotations are meant to highlight implicit causal statements. The particular text, Newsweek magazine, was not carefully selected for our purposes. The reader is invited to look in any magazine or paper on his local newstand to verify the ubiquity of causal terminology: bringing about, provoking, instigating, affecting, preventing, enabling, etcetera. Interestingly this tendency is not as evident in the scientific community. Generally the more rigorous or mathematical the subject the more subtle the causality. However, lest the reader be given the impression that causation is a lowly form of reasoning, he is referred to [Lerner 65] for an account of causation's role in scientific research.

Since causal reasoning appears to be an important component of intelligent human behavior, it stands to reason that researchers in AI should strive to understand it and make use of it in their theories. Indeed, in recent years there has been some interest in causation. Medical diagnosis systems stress the "causal reasoning" component of their systems [Pople 82] [Patil et al. 82]. Theories of "qualitative physics" employ causal notions - see the special issue of the Journal of Artificial Intelligence on the topic.

In this paper we propose a general theory of causation. A theory intended to capture our intuitive understanding of the term and at the same time be rigorous in its definition. Before we introduce our theory, however, let us explore some of the properties of causation which makes its definition hard.

In the first example, DDT is said to have "caused" various things, including the silence of birds. What the paragraph in Newsweek says is that the silence of the birds (without going into its metaphorical meaning) is *explained* by the presence of DDT. But what is the nature of this explanation? That this explanation is distinct from material implication is obvious. To quote Quine, "Whatever the proper analysis of the contrafactual may be, we may be sure in advance that it cannot be truth-functional" [Quine 59] (actually reproduced from [Barwise 85]). Thus X cannot cause X even though X logically implies X, and thunder does not cause the preceding lightning although it implies it. Causality contains a "direction"; there is no causal analog of the logical equivalence, or "if and only if". Causal explanation embodies some notion of a process: of machinery in action.

The presence of DDT, Newsweek statement claims, triggered a physiological process whose outcome was "the silence of birds". But what exactly does "triggering" mean? Consider the later example about the Goetz controversy. The new evidence "reignited" the debate - does that mean that if that evidence had not been revealed that the controversy would necessarily have remained dormant? Of course not; there are many other possible factors that could revive it: Goetz could have confessed to be a member of the Ku Klux Klan, a similar case of a "subway vigilante" could have occurred in the NY subway, and so on. So the statement "A caused B" does not mean that A is a necessary condition of B, that if A had not occurred that B would not have occurred. Similarly, it is not a sufficient condition. The newly disclosed information (namely, that Goetz had said to one of his victims "you don't look too bad, here's another" and shot him a second time) would have passed unnoticed if at the same time the USSR had declared war on the U.S., or alternatively if all media went on strike and the news could not spread. Most causal explanations hide many implicit conditions, usually an infinite number of them. In the excerpt from the letter to the editor it says that the use of dangerous drugs would lead to obscene profits, but that is true only if the potential profiteers did not incure equally large losses, which in turn could happen in any number of ways. To give a more macabre example, John's pulling the trigger caused Mary's death, but that is only because she was not wearing a bullet-proof vest, that the

gun was loaded, that the bullets were made of lead rather than marshmellows, and the list could be continued indefinitely.

There is a great deal of philosophical literature on causation spanning several thousand years. For an overview the reader is referred to [Mackey 74], and for the current views of some of the outstanding contemporary philosophers to [Sosa 75]. Philosophy is not, however, constrained by the need for process models and not surprisingly none have surfaced in the literature. Philosophers are concerned with the world *as it really is*, while we will be satisfied with finding a useful notation for expressing knowledge. With some luck our notation should correspond with concepts employed by humans, but at no time do we expect our theories to be "true" in any sense.

We are concerned with the way humans reason about causality because (a) they are relatively good at it and (b) it appears to be a very hard problem. While causality might profitably be excised from physics, we suspect that the basic causal notions play an important computational role in the way humans reason about the world.

In trying to pin down causation there are three main issues to be settled:

1. Ontology: What entities participate in the causal interactions? What is the relation between them and how does one derive new entities from old ones?
2. Organization: Given a commitment to what things exist, how is the information about such entities stored and how does the storage arrangement impose constraints on the computational processes that will make use of this information.
3. Computation: What processes are required to make use of the stored information.

The three issues are in many ways inextricably intertwined. However, the first can be reasonably tackled in isolation and that is precisely what we intend to do in this paper. Our solution will constrain the organization and suggest a process model but our immediate concern is to settle the ontological questions.

Organization of the paper: In section 2 we state our main thesis about causation, and contrast it with two attempts within AI, one by Schank and one by Simon, to deal with causation in a somewhat general way. In Sections 3, 4 and 5 we outline our theory of time and causation, and in Section 6 we step back and attempt some perspective on what we presume to have accomplished.

2 Thesis

In [Simon 65] Herbert Simon offers a specific definition of causality.[1] The situation in which the causality is to be determined is represented by a set of linear equations, and the entities participating in the causal relation are the variables appearing in the equations. Certain restrictions apply to the set of equations, and causation is defined essentially by the effect of perturbing one

[1]Since in the following we will criticize Professor Simon's formulation, it is only fair to point out that it is an old one, dating back in its original form to the late fifties. We understand from Professor Simon that he has recently done more work on the topic together with a student of his. We have not yet seen that work and our criticism has no bearing on it.

variable on other variables. The relation defined by Simon and called "causal ordering" is not transitive, and there is no temporal information (for example, an effect may precede the cause).

In [Schank 75] Schank identifies four types of causal relations: result, enable, initiation, and reason. In order to define these relations he distinguishes between "actions" and "states". To reason about causality Schank constructs "causal chains", and suggests how such chains can be reconstructed given only partial description of them.

Neither of these formulations, we claim, is acceptable. Our thesis is that in order to define causation one must have a well defined theory of time and a representation and means of reasoning about what we will call causal conditionals. In more detail we posit the following.

- A prerequisite for defining causation is having a precise notation for temporal information, which includes precise definitions of terms like facts and events. We will call it an ontology of time. Simon's representation contains no temporal information, as he himself points out. Schank does pay attention to entities like "action" and "state" and to the sequencing of such entities in time, but their meaning is left intuitive; he offers no precise definition of them.
- Causation cannot, in principle, be defined solely in terms of constraints on the simultaneous truth value of temporal assertions. For example the constraint F=MA does not contain causal information, nor would that information be obtained by the addition of other constraints. This is in contrast to the approach proposed by Simon.
- Instead it is defined in terms of what we will call a theory of causal conditionals. For example, the rule F=MA is replaced by others, one of which may have the rough form of "if you did this to F then that would happen to A, all other things being equal". Causally conditional statements do not suffer from the inherent symmetry of constraints.

Based on this we will outline a theory of causation in three stages - develop a theory of time, in terms of that a theory of causal conditionals, and finally use the latter to define causation.

3 An ontology of time

In this section we will briefly outline the theory which is described in more detail in [Shoham 85]. The theory and notation are influenced by the work of James Allen [Allen 84] and Drew McDermott [McDermott 82]. In this paper we will ignore most of the technical details and try to convey the main ideas. These details are, however, important, and the interested reader is encouraged to refer [Shoham 85].

The basic statement about time relates an interval of time, denoted by its two endpoints, and an associated proposition. So, for example, we will write $HOLD(t_1, t_2, LOCATION(ball, loc))$ to represent the fact that the ball *ball* is in location *loc* from t_1 to t_2. The term $LOCATION(ball, loc)$ is a *fact type*, and the tuple $< t_1, t_2, LOCATION(ball, loc) >$ is a *fact token*. It is possible, as several people have done, to similarly define *event types* and *event tokens*. We find that this separation is somewhat arbitrary and tends to obscure important distinctions. For example, *The ball rolled* is an fact and *The ball changed location* is a event, though they both denote the same situation. Also, we will want to speak of facts like *The ball rolled* AND *there was another ball in*

its path, or *The ball did* NOT *roll.* How do we conjoin two different entities like facts and events? Furthermore, consider the definition of an *action.* Actions are usually defined in terms of an event or fact, associated with other concepts like an actor and an intention. But that intuitively would call for two kinds of actions: "event actions", like *John went home*, and "fact actions", like *John stood still.*

Events and facts have enough in common for us to lump both into the generic category facts. We will separate out "event-like" facts from "fact-like" facts by their particular properties.

We classify facts along several dimensions. A fact type can be *holographic*, meaning that if it holds over an interval I then it holds over every subinterval of I. "John is stood still" is an example of a holographic fact. A fact can be *gestalt*, meaning that if it holds over an interval I it does *not* hold over any subinterval of I. For example, "John walked 3 feet south" is gestalt. Independently, a fact type may or may not be *mergeable.* Intuitively, a fact type is mergeable if whenever it holds over two overlapping intervals it holds over their union (in fact, the technical definition is slightly different). Fact types that are both holographic and mergeable are call *liquid.* For example, "John stood still" is liquid.

Along with the classification of fact types we impose a certain structure on fact types, by recasting James Allen's [Allen 84] definitions into our notation. This structure allows us to achieve the effect of having logical connectives inside the scope of $HOLD$, so that the meaning of $HOLD(t_1, t_2, AND(f_1, f_2))$, $HOLD(t_1, t_2, NOT(f))$, etcetera are well defined. Quantifiers receive similar treatment, so one can speak of fact types of the form $FORALL(x, f(x))$. The particular definitions have interesting properties such as that for any fact type f, $NOT(f)$ is holographic (these are explored in [Shoham 85]).

4 A theory of causal conditionals

To summarize the previous section, all temporal information is expressed by fact tokens, and inference rules which refer to the classification of fact types and allow the deduction of new fact tokens from old ones. This information contains no modality. Each tuple presumably represents an actual fact in the world. We say "the world", since our notation so far only allows us to describe one state of affairs. The theory of causal conditionals, we said, would represent and reason about information of the rough form "if X occurs then Y will occur, all other things being equal". We must define what the X and the Y are, and above all spell out the meaning of "all other things being equal". Here too we will only sketch the main ideas of what we call "potential facts theory", keeping formalism down to a minimum.

Imagine that our world - past, present and future - is inhabited by a tremendous number of *potential facts* which try to manifest themselves, as it were. For example, consider a billiard ball rolling across the table. There is the potential fact of its rolling, which in this case also manifests itself - call it ROLLING1. Now suppose that we roll another ball that collides with the first rolling one. The laws of physics tell us that at that point the balls will stop rolling in the "expected" direction and begin rolling in a new direction. However we do not forget where each ball *would have rolled* were it not for the collision. In particular we remember ROLLING1, only it is not a fact any longer - it manifested only an initial segment of itself. We call it a *potential fact* - it starts out as a fact but due to interactions with other potential facts it becomes a counterfact. In the first scenario, where the second ball was not rolled, ROLLING1 actually managed to manifest

all of itself. In the second scenario it managed to manifest only the part of it which preceded the collision. If, for example, we rolled yet a third ball which collided with the second ball before the second ball has a chance to collide with the first one, ROLLING1 would again manage to manifest all of itself.

The image to keep in mind then is of a world full of potential facts fighting for existence and very few "real" facts which are selected by some central arbiter - reality is but the tip of an iceberg.

Without going into too much detail, the way we represent potential fact tokens is by replacing the previous 3-tuples $< t_1, t_2, \textit{fact-type}>$ by 5-tuples $< \textit{world}, \text{token-name}, t_1, t_2, \textit{fact-type}>$. $\textit{Token-name}$ is a unique name associated with each such token, and *world* defines the world to which that fact token belongs. We will return to world definers soon, but first let us look at what these fact tokens mean. They can mean one of four things, depending on whether the token is *potential, materialized, ghost* or *real*. By convention, the token-name determines which type the token is. For each potential fact token $< w, \textit{pid}, t_1, t_2, f >$ there exist unique $< w, \textit{mid}, t_1, t_3, f >$ and $< w, \textit{gid}, t_3, t_2, f >$ such $\textit{mid} = MATERIAL(\textit{pid})$ and $\textit{gid} = GHOST(\textit{pid})$. We will say more about real fact tokens later.

For example, consider two billiard balls colliding. We have the two potential fact tokens (for convenience we replace logical functions by English text in the fact type):

$< w, \textit{rolling}1, 1, 9, \text{"}\textit{ball1 rolls from loc1 in direction dir1}\text{"} >$
$< w, \textit{rolling}1, 2, 8, \text{"}\textit{ball2 rolls from loc2 in direction dir2}\text{"} >$

Now suppose we know that the balls will collide at time t=4 and location *loc*3 and roll in new directions. We have not yet said *how* we know that the balls will collide - we will soon - but here we are just demonstrating the representation. We then also have the following:

$< w, MATERIAL(\textit{rolling}1), 1, 4, \text{"}\textit{ball1 rolls from loc1 in direction dir1}\text{"} >$
$< w, MATERIAL(\textit{rolling}2), 2, 4, \text{"}\textit{ball2 rolls from loc2 in direction dir2}\text{"} >$

$< w, GHOST(\textit{rolling}1), 4, 9, \text{"}\textit{ball1 rolls from loc1 in direction dir1}\text{"} >$
$< w, GHOST(\textit{rolling}2), 4, 8, \text{"}\textit{ball2 rolls from loc2 in direction dir2}\text{"} >$

$< w, \textit{rolling}3, 4, t_1, \text{"}\textit{ball1 rolls from loc3 in direction dir3}\text{"} >$
$< w, \textit{rolling}4, 4, t_2, \text{"}\textit{ball2 rolls from loc3 in direction dir4}\text{"} >$

where the $\textit{dir}3$, $\textit{dir}4$, t_1 and t_2 are again determined in an as yet unknown way.

What determines which potential fact tokens exist, and how each is decomposed into its materialized and ghost part? The answer to the first question is what we call *the generating function*, and the answer to the second is what we call *the clipping function*. Together they represent the "physics" of the world, the set of rules that are believed to govern the world.

The generating function is a collection of statements of the form "if X, Y,... materialize then A,B,... potentially hold". For example, if you (actually) fire a gun loaded with n bullets then from then on it (potentially) has $n-1$ bullets. Or if you (actually) put a block on the table then from then on it (potentially) is on the table. Notice that the generating function only adds potential facts to existing ones - so where do the "first" potential facts come from? This is where what we called the "world definer" comes into play. A "world definer" is just a set of potential facts: a set of initial conditions so to speak. Together, the world definer and the generating function

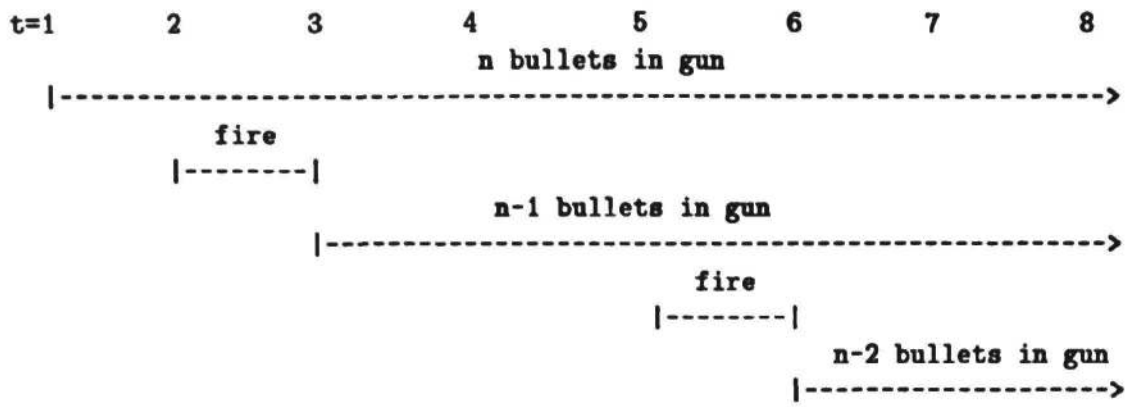


Figure 1: The potential fact tokens describing John's double shooting

determine what potential facts exist[2]. For example, in a world defined by the set {"the gun had n bullets as of t=1", "John fired the gun from t=2 to t=3", "John fired the gun from t=5 to t=6"}, and assuming that the only generating rule is the one about gun firing described above, we have exactly potential facts shown in Figure 1.

Notice that the generating function says nothing about there not being n bullets in the gun after it was fired. This is the responsibility of the clipping function. The clipping function is a collection of statements of the form "if Y, Z,... ever materialize then the ghost part of X will begin no later than t". For this to make sense, the type of the fact token X must be holographic. For example, the clipping function may include the rule "if you fire a loaded gun it no longer has the number of bullets it used to". Adding this clipping rule to the scenario just described, we get situation described in Figure 2.

This is the basic outline of the theory of causal conditionals. The precise definitions of the generating and clipping functions appear in [Shoham 85] but are too lengthy to include here.

5 Causation

Let's summarize the theory of causal conditionals presented thus far. A world is defined by a set of initial potential facts called the world definer. Given a world, the generating function determines what potential fact tokens exist. For each such potential fact token, the clipping function determines its decomposition into materialized and ghost parts.

5.1 Direct causation

Each potential fact token X that is not in the world definer was derived by the generating function applied to a set of other potential fact tokens. Each token in that set is said to be *a direct cause* of X. Notice that in general a token has several direct causes. Referring back to Figure 2, the potential fact token "there are $n-2$ bullets in the gun from t=6 onwards" has two direct causes:

[2]This is not strictly true as we shall see; the clipping function will also influence which facts materialize.

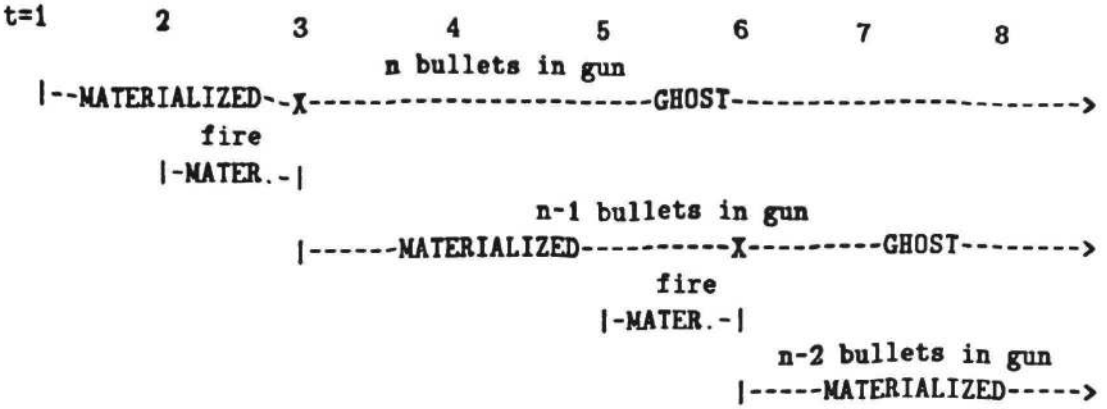


Figure 2: The potential, materialized and ghost fact tokens describing John's double shooting

"there are $n-1$ bullets in the gun from t=3 onwards" and "John fired the gun from t=5 to t=6". In turn, "there are $n-1$ bullets in the gun from t=3 onwards" has two direct causes of its own: "there are n bullets in the gun from t=1 onwards" and "John fired the gun from t=1 to t=2". However, "there are n bullets in the gun from t=1 onwards" has no direct causes.

The direct causes of materialized fact tokens are defined to be simply the direct causes of the potential tokens to which they belong. So in the previous example, the materialized fact token "there are $n-1$ bullets in the gun from t=3 to t=6" has two direct causes: "there are n bullets in the gun from t=1 onwards" and "John fired the gun from t=1 to t=2".

Definition of the direct cause of ghost fact tokens is slightly more complex and we will not give it here. We will only make a note of the fact that "causing a ghost fact token X" is not the same as "preventing X". If "X prevented Y" then if X had not occurred then Y would have, but we have already seen that "X caused Y not to occur" does not mean that if X had not occurred then Y would have.

Finally, we define the direct causes of real fact tokens. Real fact tokens were mentioned earlier on but not explained. We will explain them through an example. Consider a world in which John and Bill both shot Mary simultaneously. Assume that there is one generating rule, stating that when the trigger of a loaded gun is pulled then the gun has one less bullet and the shot person is dead. There is also one clipping rule stating that under the same conditions the gun no longer has as many bullets as it did and that the person no longer lives.

In this case there are two materialized tokens asserting Mary's being dead over the same interval. However, to us, observers of reality, there is only one death going on. We therefore speak of real fact tokens, which are the result of projecting materialized fact tokens onto reality, so to speak. In particular there is exactly one real fact token asserting Mary's death over the interval. In general, $< w, rid, t_1, t_2, f >$ is a real fact token exactly when there exists (at least one) materialized fact token $< w, mid, t_1, t_2, f >$. In ordinary speech, when we speak about the causes of X, X is a real fact token. When we speak about the causes of Mary's death, there is exactly one death we are referring to. The fact that in our notation two potential deaths materialized is transparent.

For each real fact token there is the set of materialized fact tokens that gave rise to it, namely

all the materialized fact tokens with which it shares the temporal extent and fact type (and of course the world definer). In the last example, the real fact token asserting Mary's death has two materialized tokens that gave rise to it, all the rest have exactly one. We now define the set of direct causes of a real fact token X to be the set of direct causes of all the materialized fact tokens that gave rise to X. Notice that this is a set of sets. In the same example the direct cause of Mary's death is the set {{"Bill's gun was loaded", "Bill pulled the trigger of his gun"},{"John's gun was loaded", "john pulled the trigger of his gun"}}.

As we mentioned, the set of direct causes of a real fact token is a a set of sets. There are a few special cases. If the set of direct causes of a real fact token Y is {{X},S1,S2,...,Sn} then we will say that X is *a direct disjunctive cause* of Y. If the set of direct causes of a real fact token Y is {{X,Z,...,T}} then we will say that X is *a direct conjunctive cause* of Y. If the set of direct causes of Y is {{X}} then we will say that X is *the unique direct cause* of Y.

In the next two subsections we introduce broader notions of causation. Unfortunately, due to length limitations, we can do little more than introduce them.

5.2 Indirect causation

We will say that X is an *indirect cause*, or simply a cause, of Y if one of the two holds:

1. X is a direct cause of Y, or
2. Z is a direct cause of Y and X is an (indirect) cause of Z.

This is a generalization of the naive notion of a "causal chain". The latter is a special case of a list of fact tokens each being the unique direct cause of the following one.

5.3 Causal origin

Given our (informal) definition of indirect causation, we now (informally) define the causal origins of fact tokens, "the original sin". The causal origin of a potential fact token are all its indirect causes which are in the world definer. For example, again referring back to Figure 2, the causal origin of "there are $n-2$ bullets in the gun from t=3 onwards" is the set consisting of the three tokens "there are n bullets in the gun from t=1 onwards", "John fired the gun from t=2 to t=3" and "John fired the gun from t=5 to t=6".

The significance of causal origins derives from the significance of world definers. The latter completely characterize the state of affairs, assuming a given physics (that is, fixed generating and clipping functions). This means that in order to control the world an agent need only control the facts and events described in the world definer. If he wishes to eliminate an undesirable phenomenon he need only eliminate some of its causal origins, or alternatively add some tokens to the world definer that will prevent the phenomenon. Similarly if he notices desirable manifestations he can set about eliminating unnecessary work which does not contribute to bringing about those manifestations.

6 Summary

We have defined what it means for one thing to cause another. In doing so we developed an ontology of time in which potential facts about the world are decomposed into materialized and ghost parts. To capture the actual physics of the world we presented a theory of causal conditionals. This introduced the idea of generating and clipping functions which suggest an organization for our causal knowledge. This organization, we claim, fits well with our intuitions about how people represent and reason about causality.

Bibliography

[Allen 84] Allen, J.F., *Towards a General Theory of Action and Time,* Artificial Intelligence, 23/2 July (1984), pp. 123–154.

[Barwise 85] Barwise, Jon, *The Situation in Logic-II: Conditionals and Conditional Information*, Technical Report CSLI-85-21, Stanford, 1985.

[Lerner 65] Lerner, D. (ed.), *Cause and effect,* Collier-Macmillan, Toronto, 1965.

[Mackey 74] Mackey J.L, *The Cement of the Universe: a Study of Causation,* Oxford University Press, 1974.

[McDermott 82] McDermott, D.V., *A Temporal Logic for Reasoning about Processes and Plans,* Cognitive Science, 6 (1982), pp. 101–155.

[Patil et al. 82] Patil, R.S., Szolovitz, P. and Schwartz, W.B, Causal Understanding of Patient Illness in Patient Diagnosis, *Proc. AAAI-82,* AAAI, 1982.

[Pople 82] Pople, H. E., Heuristic Methods for Imposing Structure on Ill-Structured Problems: The Structuring of Medical Diagnosis, Szolovits, P. ed., *Artificial Intelligence in Medicine*, Westview Press, 1982, pages 119–190.

[Quine 59] Quine, Willard Van Orman, *Methods of Logic,* Holt, Rinehard & Winston, 1959.

[Schank 75] Schank, R.C., The Structure of Episodes in Memory, Bobrow, D.G. and Collins, A. eds., *Representation and Understanding,* Acadmic Press, 1975, pages 237–272.

[Shoham 85] Shoham, Y., Time and Causality from the Standpoint of AI1985. In preparation.

[Simon 65] Simon, H. A., Causal Ordering and Identifiability, Lerner, D. ed., *Cause and Effect,* Collier-Macmillan, Toronto, 1965.

[Sosa 75] Sosa, E., *Causation and Conditionals,* Oxford University Press, 1975.

Story Telling and Generalization

Michael Lebowitz
Department of Computer Science -- Columbia University
New York, NY 10027

Abstract

The generation of extended plots for melodramatic fiction is an interesting Artificial Intelligence task -- one that requires the application of generalization techniques to carry out fully. UNIVERSE is a story-telling program that uses plan-like units, "plot fragments", to generate plot outlines. By using a rich library of plot fragments and a well-developed set of characters, UNIVERSE can create a wide range of plot outlines. In this paper, we illustrate how UNIVERSE's plot fragment library might be automatically extended using explanation-based generalization methods. Our methods are based on analysis of a television melodrama.

1 Introduction

In [Lebowitz 84] and [Lebowitz 85] we described how extended story telling of the sort used to create fictional serials, novel series, television melodrama ("soap operas"), and the like involves a wide range of interesting AI problems. We introduced a story-telling program, UNIVERSE, concentrating on its ability to create realistic characters [Lebowitz 84] and a scheme for generating plot outlines using plan-like units, "plot fragments" [Lebowitz 85]. In this paper, we briefly describe our model of story telling, showing how UNIVERSE generates a simple piece of a plot outline. We then discuss how the appropriate way to extend the program is for it to automatically expand its knowledge base using techniques closely allied to explanation-based learning methods (e.g., [DeJong 83]). This discussion is based on analysis of plot outlines from a television melodrama.

The story domain we have selected for UNIVERSE is that of interpersonal melodrama, as such stories provide excellent examples of narrative construction that are formulaic enough to generate by computer, and yet interesting enough to hold the interest of reader/viewers. Eventually, we expect the program to be able to generate connected stories in natural language form over a long period of time. For the moment, we are concentrating on generating plot outlines, and leaving problems of dialogue and low-level event generation for later. As an illustration of the kind of stories we would like to generate, consider the following synopsis of events from the television melodrama, *Days of Our Lives*:

> **STORY1** - Liz was married to Tony. Neither loved the other, and, indeed, Liz was in love with Neil. However, unknown to either Tony or Neil, Stephano, Tony's father, who wanted Liz to produce a grandson for him, threatened Liz that if she left Tony, he would kill Neil. Convinced he was serious by a bomb exploding near Neil, Liz told Neil that she did not love him, that she was still in love with Tony, and that he should forget about her. Eventually, Neil was convinced and he married Marie. So later, when Liz was finally free from Tony, Neil was not free to marry her and their troubles went on.

STORY1 exemplifies the kind of plot outline we wish to generate at this point in our

research as a precursor to full story generation. There are several important points to note about this example. First, the interactions among characters are quite complex. It is important that the behavior of all the characters make sense in terms of what we already know about them. On the other hand, it is not enough that we simply simulate the lives of these characters (as in TALE-SPIN [Meehan 76]), since it is unlikely that they would naturally do such interesting things or that their actions would interleave with each other so nicely. Both our generation scheme and plan for generalizing new plot fragments endeavor to create plot outlines that are believable and yet interesting.

2 The basic UNIVERSE story-telling model

Story telling in UNIVERSE is a plan-based activity (although, since we are generating extended stories, the program must "tell as it plans"). It is based around a set of "plot fragments", which are initially built-in by the program designer, that play the same role as standard plans in planning systems such as [Sacerdoti 77; Wilensky 83]. We will discuss in Section 3 how the library of plot fragments might be extended automatically. The plot fragments provide narrative methods to achieve goals. What is crucial is that the goals and plot fragments are not viewed as goals and plans of the characters (although these must also be monitored), but, instead, goals and plans of the *author* (or program, in our case). This allows actions that make sense but yet are not necessarily what a character, if an independent agent, would choose to do. This approach fits nicely with work in narrative theory such as [Barthes 77; Eco 79; Todorov 77], that has influenced the development of UNIVERSE (as has the AI work of [Meehan 76; Dehn 81; Yazdani 83]).

The plot fragments used by UNIVERSE can span a wide range of levels -- from very general, thematic plans that may take a long time to carry out, to plans for specific actions. They may include steps with the sole purpose of setting the stage for later events (e.g., *dropping-hints*). The more general plot fragments are much like the plot units in [Lehnert 81], such as *double-crossing*. Other plot fragments are more specific, such as *seduction* or *wild-fight*, but the range of the fragments, along with the use of many characters, keeps the plot outlines interesting. Before starting a new story, UNIVERSE builds up a story-telling universe of characters, keeping track of personality traits, interpersonal relationships and goals for each, as described in [Lebowitz 84]. This proves to be precisely the way that some authors work in developing novels: create a set of characters and develop the plot from there (see [Eco 84]).

Figure 1 shows the main features of a typical UNIVERSE plot fragment, *forced-marriage*. It involves a nasty parent (?parent) forcing his daughter-in-law (?her) to stay in an unhappy marriage (to ?husband), preventing her from being with the person she really loves (?him), roughly the plot of STORY1.

The first piece of information about the *forced-marriage* plot fragment is the goal that it can be used to achieve -- "churn", keep two lovers from being happy. Obviously this goal makes no sense from the point of view of the characters, but makes a great deal of sense for the author, and, indeed, is a staple of melodrama. The next pieces of information in Figure 1 are a list of the roles involved in the plot fragment and constraints upon them. (Generalization of the constraints will be an important part of expanding the plot fragment

```
PLOT FRAGMENT: forced-marriage

GOALS: (churn ?him ?her)  {prevent them from being happy}
TIME SCALE:  months
CHARACTERS:  ?him ?her ?husband ?parent
CONSTRAINTS: (has-husband ?her)          {the husband character}
             (has-parent ?husband)       {the parent character}
             (< (trait-value ?parent 'niceness) -5)
             (female-adult ?her)
             (male-adult ?him)
SUB-GOALS:  (do-threaten ?parent ?her "forget it") {threaten ?her}
            (dump-lover ?her ?him) {have ?her dump ?him}
            (worry-about ?him)     {have someone worry about ?him}
            (together * ?him)      {get ?him involved with someone else}
            (eliminate ?parent)    {get rid of ?parent (breaking threat)}
            (do-divorce ?him ?her) {end the unhappy marriage}
            (or (churn ?him ?her)      {either keep churning or}
                (together ?her ?him)) {try and get ?her and ?him back together}
```

Figure 1: A typical UNIVERSE plot fragment

library.) For the roles that are not determined by the goal, UNIVERSE will find characters that fit the constraints (or create characters, if need be).

Finally, Figure 1 shows the heart of the plot fragment -- a series of subgoals (actually, a partial ordering) that must be achieved to attain the fragment's goal. Often, a plot fragment will include actual plot actions to be generated, but since *forced-marriage* is a relatively high level plot fragment, it simply spins off the series of sub-goals shown. Each of the subgoals in Figure 1 can potentially be satisfied by a variety of different plot fragments, leading to a wide range of possible stories.

The basic UNIVERSE story-telling algorithm is relatively simple, relying on the richness of its plot fragment library and character set. UNIVERSE maintains a precedence graph indicating the prerequisites of pending author goals and plot fragments. The program selects to pursue an author goal with all its prerequisites satisfied and a plot fragment to use for it (trying to achieve extra author goals, if possible). It continues this process as long as unsatisfied goals remain, generating concrete actions when appropriate. This algorithm, much like the one used in TALE-SPIN [Meehan 76] or micro-TALE-SPIN [Schank and Riesbeck 81], but using author goals, is summarized in Figure 2. This algorithm can lead to dead ends -- branches of disjunctive plans that cannot be achieved -- but, interestingly, as long as this does not happen too often, it is not a problem, being much like the "red herrings" often introduced in television melodrama.

One of the positive side effects of this algorithm is that as multiple goals are pursued, various "plot lines", i.e., the pursuit of different high level plot fragments, will become interleaved. This is important in the production of intricate plot outlines.

Figure 3 shows a brief run of UNIVERSE. In this example, we have given the program the top-level goal of "churning" Liz and Neil's relationship (e.g., keeping them apart) and a secondary goal of getting Renee and Neil together.

The program trace in Figure 3 (with plot outline output indicated by ">>>") shows the expansion of the *forced-marriage* plot fragment for the churn goal. Once UNIVERSE has

Liz becomes available). It picks the *seduction* plot fragment to do this. Rather than picking a seductress at random from among the characters with acceptable characteristics, UNIVERSE was able to satisfy its second goal as a side-effect by selecting Renee as the seductress. This sort of opportunistic planning goes a long way towards achieving the intricate interconnections that exist in most popular melodrama.

3 The next step: Generalizing plot fragments

We feel the generation framework outlined in Section 2 and described more fully in [Lebowitz 85], which is able to generate moderately interesting plot outlines, provides a good basis for future work. To generate interesting stories over the long run, it will be necessary for the program to be able to automatically expand its plot fragment library by creating new plot fragments. We are studying this problem by looking at various plot outlines from *Days of Our Lives*. For example, consider STORY2:

> **STORY2** - Hope and Bo were very much in love. However, Bo was involved in some dangerous activities and was worried about Hope's safety. To protect her, he told Hope that he didn't love her, that he was living with Diane, a childhood friend, and that she should get on with her life. She was more or less convinced, and started spending a lot of time with Larry, the DA, who wanted to marry Hope for political purposes. Just as they were about to be married, Bo arrived, the danger past, and told Hope that he loved her. They ran off together. However, Maxwell (a new bad guy), who was interested in Larry's career, had his goons capture Hope (without Larry's knowledge) and tell her that she was to return to Larry or harm would come to Bo and her other friends. She did so (after some of Maxwell's goons beat up Bo), and in fact married Larry, convincing Bo she no longer cared for him. Then, Megan, a childhood love of Bo's appeared and told him she once bore his child. We assume that when Hope is finally free of Larry, Bo will be entangled with Megan.

STORY2 at first appears rather complex. However, if we look at it closely, we realize that it is really two sequential plot lines, each quite similar to STORY1, but with a few twists. This can be seen more clearly if we break down STORY1 and the two plot lines in STORY2 into sequences of events displayed in parallel. This is done in Figure 4.

Figure 4 makes clear the similarities among the three plot lines. For example, the first parts of STORY1 and STORY2B differ primarily in character substitutions. STORY1 and STORY2A, on the other hand, are similar at a somewhat more abstract level. So, for instance, the B events in STORY1 and STORY2B involve a marriage and an engagement, respectively, while STORY2A involves some sort of dangerous activity. To recognize the similarity among these three plot lines, we have to realize that all these events play the same role -- they create a reason to keep the lovers apart. The I events, again, have rather different actions playing the same role -- to further thwart the lovers even when the prime obstacle between them disappears.

Notice that while all of these three plot lines are quite similar, we cannot hope to generate STORY2A or STORY2B from the same rather specific plot fragment that we used to generate STORY1 (the *forced-marriage* plot fragment shown in Figure 1, or a somewhat more complex version of it). *forced-marriage* has rather specific constraints and calls for rather specific events. However, in STORY2B, instead of an overt threat from the woman's

Pick a goal with no missing pre-conditions

Pick a plot fragment for that goal
(achieving extra goals, if possible)

"Execute" the plot fragment
(generating plot actions and adding new goals to the goal graph)

Repeat

Figure 2: The basic UNIVERSE story-telling algorithm

```
*(tell '(((churn liz neil) (together renee neil))))

working on goal -- (CHURN LIZ NEIL)
Several plans to choose from FORCED-MARRIAGE LOVERS-FIGHT JOB-PROBLEM
-- using plan FORCED-MARRIAGE

working on goal -- (DO-THREATEN STEPHANO LIZ "forget it") -- using plan THREATEN

>>> STEPHANO threatens LIZ: "forget it"

working on goal -- (DUMP-LOVER LIZ NEIL) -- using plan BREAK-UP

>>> LIZ tells NEIL she doesn't love him

working on goal -- (WORRY-ABOUT NEIL) -- using plan BE-CONCERNED
Possible candidates -- MARLENA JULIE DOUG ROMAN DON CHRIS KAYLA
Using MARLENA for WORRIER

>>> MARLENA is worried about NEIL

working on goal -- (TOGETHER * NEIL)
Several plans to choose from SEDUCTION DRUNKEN-SNEAK-IN SYMPATHETIC-UNION JOB-TOGETHER
Possible candidates -- DAPHNE RENEE
Using RENEE for SEDUCER

>>> RENEE seduces NEIL

working on goal -- (ELIMINATE STEPHANO)
Several plans to choose from ATTEMPTED-MURDER EXPOSE
-- using plan ATTEMPTED-MURDER
Using ALEX for KILLER

>>> ALEX tries to kill STEPHANO

working on goal -- (DO-DIVORCE TONY LIZ) -- using plan DIVORCE

>>> LIZ and TONY got divorced

working on goal -- (TOGETHER LIZ NEIL)
no acceptable plans
```

Figure 3: Generating a simple plot outline

selected this plot fragment, each of the subgoals is pursued in turn. As it happens, in this example each subgoal can be achieved by a single plot fragment. In a larger example, of course, some of these subgoals would lead to further expansion of subgoals, which might intertwine with each other.

An interesting point in Figure 3 comes when UNIVERSE tries to satisfy the (together * neil) goal, e.g., get Neil involved with someone (so that he will be entangled romantically when

	STORY1	STORY2A	STORY2B
A	Liz loved Neil	Bo loved Hope	Hope loved Bo
B	Liz was married to Tony	Bo was involved in activity that could endanger Hope	Hope was engaged to marry Larry
C	Stephano wanted Liz married to Tony		Maxwell wanted Hope married to Larry
D	Stephano told Liz if she left Tony he would hurt Neil		Maxwell told Hope, indirectly, if she didn't marry Larry, he would hurt Bo
E	Stephano had Neil hurt in a bomb explosion		Maxwell had Bo beat up
F	Liz told Neil she didn't love him	Bo told Hope he didn't love her	Hope told Bo she didn't love him
G	Liz stayed with Tony	Bo lived with Diane	Hope married Larry
H	Neil believed her	Hope believed him	Bo believed her
I	Neil married Marie	Hope got involved with Larry and was about to marry him	Bo discovered Megan had had his child
J	Liz eventually got free of Stephano, but couldn't be with Neil because he was married to Marie		???
K	Still later, Liz was able to marry Neil	Bo's danger passed, and Bo and Hope ran off together	

Figure 4: Three similar plot outlines

father-in-law there is a covert threat from the danger Bo is in. We could, however, hope to derive from *forced-marriage* a more general plot fragment that would produce STORY2 .

Based on this sort of analysis of existing melodramatic plot outlines, we propose to automatically augment UNIVERSE's library of plot fragments by generalizing existing plot fragments.[1] In general terms, we can easily see how this could lead to the production of STORY2A and STORY2B. The *forced-marriage* plot fragment could be generalized into a "coerce into staying out of a relationship" plot fragment where event B involved a competing goal, step D a threat (optional), step G a competing relationship and so forth.

[1] An alternate, and ultimately quite similar, method would be to tell new stories by direct modification of stories told previously, without going through generalizing plot fragments. Generalizing new plot fragments fits better into the UNIVERSE scheme.

Then this plot fragment could be instantiated into either STORY2A or STORY2B (or back into STORY1).

The main process of generalization is simply a relaxation of the constraints upon the role fillers for a given plot fragment along with modification of the remainder of the plot fragment to accept any role filler within the generalization. So, for example, instead of requiring a married couple in *forced-marriage*, we might have UNIVERSE just look for any sort of a normally positive relationship. Various details will be added to the actions and sub-goals depending on how the plot fragment is instantiated.

While it is easy to simply generalize the role constraints (in fact, we have implemented such generalization), it is obviously not acceptable for such generalization to be arbitrary (or else we would just generalize all role fillers to "person"). We must consider two issues: 1) keeping the generalized plot fragments believable and 2) making sure they still generate interesting stories.

We feel that the maintenance of believability will be accomplished through processing much like the explanation-based learning methods of [DeJong 83].[2] The basic idea is that we would build up a causal analysis of a plot fragment, and then generalize it in various ways such that the analysis still holds, as done in [DeJong 83] to build up new story understanding schemata. Using the causal analysis, built up in ways such as those described in [Schank and Abelson 77], we can determine various ways that a plot fragment can be generalized while maintaining that same causal explanation. So, for example, in the *forced-marriage* plot fragment, we might be able to generalize the relation that one participant is being forced into from a marriage to any permanent relationship (e.g., "living together"), but not into a less personal relationship, like "roommates", as that would undermine the causal explanation for the plot fragment.

There are two primary ways in which our generalization methods differ from the explanation-based learning of [DeJong 83]. Both of these differences address the second of our concerns -- that the generalized plot fragments still yield interesting stories. We propose 1) that author goals be included in the explanations of the plot fragments and 2) that we not generalize plot fragments as far as possible all at once.

Inclusion of author goals in the causal analysis is crucial. In many cases, it will be impossible to analyze a plot fragment without considering such goals. For example, in *forced-marriage*, the only rationale for getting the jilted lover involved with another person is to satisfy an author goal of complicating the situation. The characters would certainly not want this to happen, particularly if they knew the original relationship would become feasible again. Even when it is possible to analyze a plot fragment based on character goals, it is likely that only the consideration of author goals will inhibit over-generalization. So, continuing our *forced-marriage* example, we might imagine a generalization that makes real-world sense if the main characters merely like, not love, each other. However, such a

---

[2]There has been considerable other work into explanation-based learning (see [Michalski et al. 83]). However, DeJong's story understanding domain is most closely related to our work.

plot fragment is unlikely to produce interesting stories, as it will not further author goals that revolve around intense relationships.

Our second point is that we do not wish to generalize plot fragments as far as possible. It is unlikely that we will be able to analyze plot fragments in enough detail to fully understand the "flavor" that they provide. We will maintain the flavor of the initial fragments by only generalizing some of the features of each plot fragment at a time. This will help both by simplifying the instantiation process (as there will be fewer degrees of freedom for the various role fillers) and will maintain most of the interesting aspects of the plot fragments. The richness of the stories that we are looking for comes from instantiating the generalized plot fragments with a variety of different role fillers. The choice of each role filler can then lead to interesting changes required to keep the story consistent (the determination of which, incidentally, can also make use of the causal analysis needed for generalization).

Even small generalizations of the plot fragments will add considerable richness to UNIVERSE's stories. Discovering that *forced-marriage* "works" if the outside threat comes to achieve political goals rather than desire for a grandchild, or that it will work if the threat is made to the man, not the woman, will expand the utility of the plot fragment. Such generalization is well within the scope of current explanation-based learning methods.

As a partial illustration of generalizing plot fragments, we have developed a simple module to generalize constraints without regard to underlying causality (which must, of course, be considered eventually). Figure 5 shows the generalized constraints for *forced-marriage*. We can see that the requirement for the female lover to have a husband with a nasty parent has been generalized into a requirement for one lover to have a spouse with a nasty parent. The requirement for opposite sex lovers has also been relaxed. This gives us a plot fragment that can describe pressure on either side of a opposite or same sex relationship (given proper definition of the predicate "has-spouse"). Note that the names of the role fillers (e.g., ?him, ?her) are now rather misleading. The level of nastiness needed by the parent has also been relaxed.

```
CONSTRAINTS: (has-spouse ?her)          {instead of has-husband}
             (has-parent ?husband)
             (< (trait-val ?parent 'niceness) 0)
             (adult ?her)               {instead of female-adult)
             (adult ?him)               {instead of male-adult}
```

Figure 5: Generalized constraints for *forced-marriage*

Figure 6 shows how the generalized version of *forced-marriage, forced-marriage0*, can be used by UNIVERSE. When we ask the program to "churn" a relationship between David and Renee, it can use the generalized plot fragment, since David's wife's father (Alex) is nasty enough to threaten David. The original *forced-marriage* would not be applicable since Renee does not have a nasty father-in-law. After the selection of *forced-marriage0*, processing proceeds much as in Figure 3. It is worth reiterating that the plausibility of *forced-marriage0* is somewhat fortuitous, since we have not implemented the causal analysis that will ultimately be necessary.

```
*(tell '(((churn david renee))))

working on goal -- (CHURN DAVID RENEE)
-- using plan FORCED-MARRIAGE0 (FORCED-MARRIAGE with generalized constraints)

working on goal -- (DO-THREATEN ALEX DAVID "forget it") -- using plan THREATEN

>>> ALEX threatens DAVID: "forget it"

working on goal -- (DUMP-LOVER DAVID RENEE) -- using plan BREAK-UP

>>> DAVID tells RENEE he doesn't love her

working on goal -- (WORRY-ABOUT RENEE) -- using plan BE-CONCERNED
Possible candidates -- MARLENA JULIE DOUG ROMAN DON CHRIS KAYLA
Using MARLENA for WORRIER

>>> MARLENA is worried about RENEE

working on goal -- (TOGETHER * RENEE) -- using plan SEDUCTION

>>> RENEE seduces ROMAN

working on goal -- (ELIMINATE ALEX)
Several plans to choose from ATTEMPTED-MURDER EXPOSE
Using STEPHANO for KILLER

>>> STEPHANO tries to kill ALEX

...
```

Figure 6: Using a plot fragment with generalized constraints

4 Conclusion

There are many areas left to explore before our generalization methods can be fully implemented -- when to generalize a plot fragment, deciding exactly how much to generalize, and using a casual explanation to adjust details of the plot fragment, for example. However, we feel that this kind of explanation-based generalization of plot fragments, in conjunction with the basic story-telling methods we have developed, will lead to dynamic systems that can generate wide ranges of interesting plot outlines. When coupled with the appropriate natural language generation techniques, such systems will produce interesting and exciting stories, as well as help us learn a great deal about many areas of cognitive processing.

References

[Barthes 77] Barthes, R. *Image -- Music -- Text*. Hill and Wang, New York, 1977.

[Dehn 81] Dehn, N. Memory in story invention. Proceedings of the Third Annual Conference of the Cognitive Science Society, Berkeley, California, 1981, pp. 213 - 215.

[DeJong 83] DeJong, G. F. An approach to learning from observation. Proceedings of the 1983 International Machine Learning Workshop, Champaign-Urbana, Illinois, 1983, pp. 171 - 176.

[Eco 79] Eco, U. *The Role of the Reader*. Indiana University Press, Bloomington, Indiana, 1979.

[Eco 84] Eco, U. *Postscript to The Name of the Rose*. Harcourt Brace Jovanovich, New York, 1984.

[Lebowitz 84] Lebowitz, M. "Creating characters in a story-telling universe." *Poetics 13*, 1984, pp. 171 - 194.

[Lebowitz 85] Lebowitz, M. Generating plot outlines. Columbia University Department of Computer Science, 1985.

[Lehnert 81] Lehnert, W. G. "Plot units and narrative summarization." *Cognitive Science 5*, 4, 1981, pp. 293 - 332.

[Meehan 76] Meehan, J. R. The metanovel: Writing stories by computer. Technical Report 74, Yale University Department of Computer Science, 1976.

[Michalski et al. 83] Michalski, R. S., Carbonell, J. G. and Mitchell, T. M., eds. *Proceedings of the 1983 Machine Learning Workshop*. University of Illinois, Urbana-Chanmpage, IL, 1983.

[Sacerdoti 77] Sacerdoti, E. *A Structure for Plans and Behavior*. Elsevier North-Holland, Amsterdam, 1977.

[Schank and Abelson 77] Schank, R. C. and Abelson, R. P. *Scripts, Plans, Goals and Understanding*. Lawrence Erlbaum Associates, Hillsdale, New Jersey, 1977.

[Schank and Riesbeck 81] Schank, R. C. and Riesbeck, C. K. *Inside Computer Understanding*. Lawrence Erlbaum Associates, Hillsdale, New Jersey, 1981.

[Todorov 77] Todorov, T. *The Poetics of Prose*. Cornell University Press, Ithica, New York, 1977.

[Wilensky 83] Wilensky, R. *Planning and Understanding*. Addison-Wesley, Reading, MA, 1983.

[Yazdani 83] Yazdani, M. Generating events in a fictional world of stories. Technical Report R-113, Computer Science Department, University of Exeter, 1983.

MULTIPAR: a Robust Entity-Oriented Parser[1]

Jill Fain, Jaime G. Carbonell, Philip J. Hayes and Steven N. Minton
Computer Science Department, Carnegie-Mellon University,
Pittsburgh, PA 15213, USA

1. Objectives

The phenomenon of human language comprehension has posed a considerable number of challenging problems for linguists, psychologists, philosophers, and artificial intelligence researchers alike. One particularly important aspect is the robust manner in which people are able to comprehend novel utterances, many of which deviate from the abstract notion of grammatical correctness. Whereas most linguists and philosophers focus on formulating *competence* theories of the idealized speaker, our objective is to model human-like *performance* in robust comprehension of naturally-occurring utterances. Within that general objective, this paper presents a concrete computational model of language interpretation capable of tolerating grammatical deviations, approximately to the extent exhibited in human language comprehension. Unlike the psychological approach that strives to model the fine-structure internal process of a cognitive task, we seek only to match observed human performance. Eliminating the constraint of strict adherence to human processing — to the extent that it can be measured or theorized — enables us to make much more rapid progress in developing an effective computational model.

Thus, our objectives can be summarized as follows:

- Understand the information processing requirements of robust natural language comprehension, including the integration of syntactic, semantic and pragmatic knowledge.
- Build an effective computational model for experimentation, refinement, validation, or refutation of our theoretical precepts. This requires the formulation of flexible control strategies and fairly rich knowledge representation formalisms.
- Apply the robust language interpreter to human-computer natural language interfaces, and investigate the extent to which such powerful, human-like abilities enhance the communication process.

In order to realize these objectives, we developed an experimental parsing system called MULTIPAR [4] based on earlier work at Carnegie-Mellon University, such as the CASPAR and DYPAR parsers [3, 9, 4, 5]. Unlike previous work on ATN-based robust parsing [14, 13, 11, 10], our approach is based on more flexible and interpretive case-frame instantiation methods. The case-frame approach enables us to integrate diverse syntactic and semantic knowledge into a universal data structure, which can be accessed by different parsing strategies. In essence, the computational model elaborated below consists of finding the best possible parse satisfying all the syntactic, semantic and pragmatic constraints and expectations. If one fails to find such a parse — perhaps due to missing function words, misspellings, other grammatical errors, semantic constraint violations, etc. — the constraints are gradually relaxed *in a principled manner* to find the best possible parse(s). A second theme interwoven with the progressive relaxation process, is the notion of invoking multiple parsing strategies, depending upon expectations of grammatical structure, semantic role, and suspected deviations from the correct grammar. The bulk of this paper presents the methods and knowledge sources that make robust parsing an effective computational process. The principles underlying our approach are:

- ***Grammatical tolerance levels***: The search space of possible interpretations of an utterance can grow prohibitively large if one allows compounding of all possible grammatical and semantic deviations. Therefore, the search is carried out using a least-deviant-interpretation-first strategy. This requires the postulation of additive *tolerance levels*. Each grammatical deviation is assigned a tolerance level, and multiple deviations in the same utterance correspond to the sum of the tolerance levels.
- ***Multiple strategies***: Each known deviation from grammatical well-formedness indexes one or more recovery strategies, as well as the required tolerance level increment. An agenda-based control structure allows strategies to post hypotheses of suspected deviations (and their associated recovery strategies), which are examined subsequently

[1]This research was sponsored in part by the Defense Advanced Research Projects Agency (DOD), ARPA Order No. 3597, monitored by the Air Force Avionics Laboratory under contract F33615-78-C-1551, and in part by the Air Force Office of Scientific Research under Contract F49620-79-C-0143. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of DARPA, the Air Force Office of Scientific Research or the US government.

only in the case that no fully grammatical, or less deviant interpretation is found.

- ***Entity-oriented parsing***: An entity is a generalized case-frame. In addition to storing the semantic roles, headers, and case markers, it may represent declaratively multiple syntactic mappings from the surface input to the semantic representation.

The power of the MULTIPAR system lies in the integration of these principles into a cohesive computational model. The central objective of the system is to comprehend less-than-perfect utterances much as humans would, rather than to hallucinate all possible nonsensical interpretations. The first stage is to develop the general computational model, and the second is to tune it to find only those interpretations corresponding to human-tolerable grammatical deviations. After outlining the continuation-passing control structure and the entity-based knowledge representation, we present in some detail one example of parsing a plausible but imperfect utterance.

The MULTIPAR approach differs from earlier inference-intensive systems that sought to complete a parse in the presence of an unknown word [7, 2] in that it provides a uniform computational framework to recover from all grammatical difficulties and deficiencies. Moreover, it differs from text-scanning approaches in that it does not resort to ignoring large segments of input it judges either incomprehensible or uninteresting [12, 6].

2. MULTIPAR Control Structure

Like most natural language parsers, MULTIPAR operates by searching through a virtual space of possible parses for a given input. The size of the search space depends on the number of local ambiguities that are encountered during the parsing process. Because MULTIPAR is expected to parse ungrammatical input, it is typically confronted with a search space that is much larger than that explored by conventional parsers. Unlike its more conventional predecessors, MULTIPAR cannot simply reject a partial parse when a grammatical constraint is violated. Instead, various recovery techniques are applied.

If the best parse is to be found in a timely manner, the exploration of alternative paths in the search space must be carefully controlled. For example, spelling correction should be tried before hypothesizing a missing word, and hypothesizing one word is preferable to hypothesizing five. Moreover, if a strategy fails to find a correct parse for some input, it does not mean that the recovery actions should be invoked immediately. It may be that some other strategy will find a grammatical and semantically consistent parse. If so, that other strategy should be given its chance before any recovery actions are attempted.

To control the exploration of the search space, competing alternatives within a strategy are explicitly specified using SPLIT statements. When a SPLIT statement is encountered, the computation divides into separate branches; each branch has a *flexibility increment* indicating the additional degree of ungrammaticality tolerated by the associated continuation. A partial parse is generated along each branch, and each computation may proceed from the SPLIT statement independently of the others.[2]

```
(Split (+0 continuationA)
       (+1 continuationB)
       (+3 continuationC)...)
```

Thus, a SPLIT statement such as the one above produces a three-way branch in the search tree. Continuation A does not result in a gain in flexibility of the associated parse (implying that this continuation requires no additional relaxation from grammatical correctness). Continuations B and C, if they are ever pursued, will result in gains of one and three respectively (implying that different degrees of additional relaxation from grammatical correctness are required to continue each parse).[3] All continuations in a SPLIT statement are posted in a global agenda, indexed by their global *flexibility level* (explained below). The structure of a global agenda looks like:

```
[Agenda: (0  Continuation-A  Continuation-B ...)
         (1  Continuation-C ...)
         (4  Continuation-D ...) ...]
```

---

[2] MULTIPAR is implemented in Common Lisp.

[3] The non-determinism provided by the SPLIT statement is also exploited for handling normal ambiguities. When a strategy finds that a parse is ambiguous, a separate branch is created for each alternative parse. +0 flexibility increments are used for each locally-ambiguous continuation.

MULTIPAR starts with an agenda containing one or more top-level strategies at the 0-flexibility level, and a global flexibility level set to 0 (i.e. full grammaticality). As each strategy is applied it can post additional continuations in the agenda at the 0 or higher flexibility levels. Continuations are posted when SPLIT statements are encountered, and the flexibility levels at which they are posted correspond to the sum of the global flexibility level and the flexibility increment of the continuation. When a continuation completes execution it is removed from the agenda.

MULTIPAR pursues all the continuations at the 0-flexibility level. Ideally, exactly one of these yields a complete and consistent parse, indicating that the input was grammatical and unambiguous. If more that one parse is generated, we have true ambiguity, which must be resolved by external means. If no parses are generated at the 0-flexibility level, the input is ungrammatical (or semantically inconsistent), and MULTIPAR goes on to the next-lowest flexibility level in the agenda, selecting this one as the new global flexibility level. The process is repeated at this flexibility level (perhaps resulting in more continuations being posted on the agenda), and if, again, no parses are found, the global flexibility level is once more augmented. In this fashion, the search for a possible parse continues, seeking the least deviant interpretations first. If the global flexibility level reaches some preset threshold, MULTIPAR judges the input to be incomprehensible and abandons its parsing attempts.

3. MULTIPAR Entities

MULTIPAR is an entity-oriented parser. As described in [8], entity-oriented parsing is an approach to restricted domain natural language processing in which the parser is driven by a set of definitions of domain entities (objects, operations, and states). The entities are defined at a high level of abstraction, primarily in terms of other component entities. This approach to parsing is well adapted to dealing with ungrammatical input; it allows a parser to interpret the abstract entity definitions in a variety of ways so that it can look for the entity components in places other than the syntactically correct ones.

A simplified example of an entity definition used by MULTIPAR is:

```
(EntityName     MoveCommand
 SemanticCases  (
        Object      (FileObjDesc or DirectoryObjDesc)
        Source      (DirectoryObjDesc or LogicalDeviceObjDesc)
        Destination (FileObjDesc or DirectoryObjDesc)
        Location    (DirectoryObjDesc or LogicalDeviceObjDesc))
 Constraints    (
        (Destination FileObjDesc > Object FileObjDesc)
        (Object DirectoryObjDesc > Source LogicalDeviceObjDesc) ...
        (Required Object Destination))
 SurfaceForms   (
        (SFName Icf-Canonical
         Head           <movehead>
         DirectObject   Object
         Cases          (
                (Preposition <sourcepreps>
                 Bind Source) ...)))
 InstanceTemplate (
        Action  'MOVE
        Deviations Deviations
        Source (
                IsA (IsA in Object)
                Name (Name in Object)
                Extension (Extension in Object)
                Directory (Directory in Object or ... or Directory in Location)
                LogicalDev (LogicalDev in Object or ... or LogicalDev in Location)
                ObjDesc  (Description in Object)
                SourceDesc (Description in Source)
                LocDesc (Description in Location))
        Destination (
                IsA (IsA in Destination)
                Name (Name in Object or Name in Destination)
                Extension (Extension in Object or Extension in Destination)
                Directory (Directory in Destination or Directory in Location)
                LogicalDev (LogicalDev in Destination or LogicalDev in Location)
                DestDesc (Description in Destination)
```

```
LocDesc (Description in Location))))
```

This defines the "move" command for an operating system interface. Like all entity definitions, it has four main parts:

- **SemanticCases**: this defines the basic structure of the entity in terms of the other entities that are its components. MoveCommand, for instance, has an object (the file[4] or directory to be moved), a source (the directory or logical device to move it from), a destination (the file or directory to move it to), and a location (the directory or logical device in the context of which the move takes place). These semantic cases (collectively constituting a case-frame) do not tell MULTIPAR how to recognize the entity in an input utterance, they just define what other entities may or must be found in the input as part of finding the entity defined. The general format is a list of case names and filler types.

- **Constraints**: For any specific instance of an entity, the constraints specify both relations that are required to hold between cases, and predicates on individual cases. The first constraint for MoveCommand says that if the Destination case is filled by a file, then the Object case must also be filled by a file. Many constraints of this kind may be needed to fully specify an entity. The final constraint says that the Object and Destination cases are required to be present for any instance of MoveCommand. MULTIPAR enforces constraints of both types insofar as it can. In other words, it prefers parses in which the constraints are satisfied, but will accept (as a deviation) inputs in which the constraints are violated.

- **SurfaceForms**: This component of an entity definition tells MULTIPAR how an entity can be described in English. That is, it tells where in the input to find words identifying the entity itself (e.g. "move" or "transfer" in our example), as well as where to find fillers for the SemanticCases. The information about where to look for the SemanticCases is implicit in the specific parsing strategies associated with the SurfaceForm name (Icf-Canonical, or "imperative case frame canonical," above). Although there is a 1-to-1 mapping between SurfaceForm and strategy in the current implementation, it is not necessary that this be the case. A strategy can know how to parse more than one surface form and a SurfaceForm can be used by more than one strategy.

  The SFName is the only attribute common to all SurfaceForms. The other attributes are specific to particular surface forms. In the above example, the Head attribute defines the imperative verb to be used to identify the MoveCommand, the DirectObject attribute indicates which SemanticCase is the syntactic direct object of the imperative verb, and the Cases attribute defines which prepositions are used to mark the other SemanticCases in English input. Symbols like <movehead> are non-terminals in a grammar used by DYPAR [3, 1] (our pattern-matcher) and expand in the course of computation (e.g. <movehead> -> move | transfer). SurfaceForm slot fillers that are not surrounded by <>'s are SemanticCase names and tell the strategy which SemanticCase to bind the information to. Consider the DirectObject case of the Icf-Canonical SurfaceForm in the MoveCommand. The strategy that knows how to parse this kind of surface form will be passed an instance of the MoveCommand entity definition and an input segment to work on. When it finds a noun phrase unmarked by prepositions, it will check the surface form to find out what SemanticCase it should be trying to fill. Since the Object SemanticCase can be filled by either a FileObjDesc or DirectoryObjDesc, it will try to parse the unmarked segment as each of these. This will result in calls to strategies that know how to handle the SurfaceForms in the entity definitions of FileObjDesc and DirectoryObjDesc respectively. In general, an entity may have more than one surface form corresponding to different forms of surface expression for the same underlying semantic cases. The entity definition for FileObjDesc, for example, has two SurfaceForms: Ncf-Canonical[5] and Ncf-System. The former is designed for recognition of input like "the files with extension lsp in directory [c410jf90]," while the latter is used for recognition of descriptions that include proper names, such as "foo.lsp in [c410jf90]".

- **InstanceTemplate**: This information is used when a strategy has finished parsing an entity. It tells MULTIPAR the final representation to use for the entity instance thus produced. It is essentially a method for reformatting, "canonicalizing," and pulling information out of subordinate entity instances to compose the current one. The slot fillers in the InstanceTemplate act as directions to a routine that uses the bindings to the SemanticCases produced by the strategy. A single word means the value of the slot is exactly what is bound to the SemanticCase. A list without "or" means the value of the slot is whatever is found in the slot with that name in the InstanceTemplate bound to the SemanticCase. (e.g. IsA in Object says look at the IsA field in whatever kind of entity is bound to Object — if the field is not found, the slot = nil). Finally, directions that have one or more "ors" act as deterministic disjuncts. Each

---

[4] FileObjDesc means file object description; other abbreviations are similar

[5] Ncf is an abbreviation for NominalCaseFrame

instruction is followed until a non-nil value is found. In this way, "move [c410jf90]foo to my directory" produces the same InstanceTemplate as "move foo from [c410jf90] to my directory." In the former, the directory name for the source of the move is found in the FileObjDesc InstanceTemplate bound to the Object SemanticCase. In the latter, the directory name for the source of the move is found in the DirectoryObjDesc InstanceTemplate bound to the Source SemanticCase. Finally, each InstanceTemplate has a special slot called "deviations" which has no corresponding SemanticCase. This slot acts as a repository of information about the recovery actions taken by strategies.

4. An Annotated Example

To make clear how all the components of MULTIPAR combine and interact to parse both well-formed and ungrammatical input, we present the following extended example. Readers are advised to refer frequently to the figures in this section, and to the entity definition shown in section 3. In the discussion that follows, function calls and strategy names are written in boldface type; entity slots and values are given in italics. The notation "$word_1 ... word_n$" refers to the portion of the input utterance starting with $word_1$ and ending with $word_n$.

Consider the behaviour of the system when the user types:

Move the accounts directory the file Data3.[6]

Step 1. Initialization.

Before any part of the input is examined, the control tree is initialized as in Figure 1, label (a)[7]. Here, MULTIPAR sets up a branch for each of the commands in its current vocabulary. Note that no flexibility increment is added to the initial value of zero because no deviation has occurred. The control chooses the first level 0 branch on the agenda for continuation.

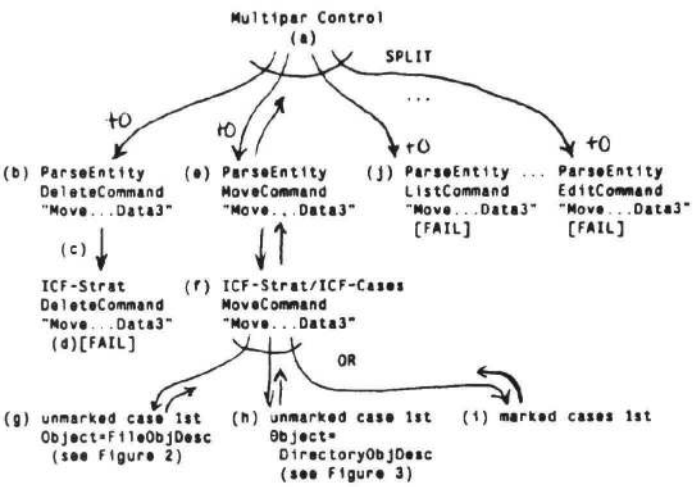


Figure 1.
Annotated Example, Steps 1 4

Step 2. Try to parse "Move...Data3" as a *DeleteCommand*.

ParseEntity (Figure 1,(b)) is a function that maps entity types to strategies. A request for a command entity could result in the trial of a number of different strategies. At present, only top-down versions of the parsing strategies exist, and calls to **ParseEntity** that look for commands are always mapped into calls to the imperative caseframe strategy (Figure 1,(c)). **ImperativeCaseFrame-Strat** will be unable to find an appropriate verb for the *DeleteCommand* entity and will fail without scheduling any alternate branches (i.e. this can be viewed as a non-recoverable error for the top-down strategy). Failure means that processing is continued by the control structure which eliminates this branch (Figure 1,(d)) and chooses another (Figure 1,(e)). Of course, the new branch also contains a call to **ParseEntity**; this call is mapped as above (Figure 1,(f)).

Step 3. Try to parse the input as a *MoveCommand*.

ImperativeCaseFrame-Strat knows how to use the *Icf-canonical SurfaceForm* to interpret the input. In the *MoveCommand* entity definition shown in section 3, this is the only *SurfaceForm*. As linguistic coverage is extended to include, for example, declaratives and interrogatives, new *SurfaceForms* must be defined. **ImperativeCaseFrame-Strat** could be expanded to interpret all top-level forms or each form could be provided with its own "expert".

The first action that **ImperativeCaseFrame-Strat** takes is to use the *Head* field in the *SurfaceForm* to search for a legal

[6] The grammatically correct version of this sentence is "Move to the accounts directory the file Data3."

[7] Hereafter, simply Figure 1,(a).

verb.[8] It finds "Move" and the unparsed segment is reduced to "the accounts directory the file Data3". The next step is to call a routine to fill the *SemanticCases* of the *MoveCommand*.

Step 4. Use **ImperativeCaseFrame-Cases** to fill *MoveCommand*'s *SemanticCases*.

ImperativeCaseFrame-Cases (Figure 1,(f)) is not a strategy itself but only a part of the top-down imperative caseframe strategy. The distinction is important because the responsibility of a strategy is to return an "instance list," i.e. one or more instantiated *InstanceTemplates*. **ImperativeCaseFrame-Cases** will return lists of consistent *SemanticCase* bindings which **ImperativeCaseFrame-Strat** will use to fill in *InstanceTemplates* when building its instance list. Even if **ImperativeCaseFrame-Cases** fails to fill any cases, **ImperativeCaseFrame-Strat** may return a non-empty instance list.

When filling cases we impose no order on their appearance in the utterance, nor do we fill required cases first (doing so would eliminate possible parses at flexibility levels greater than 0). As we try to expand a partial parse the still-unfilled cases may be constrained in the kinds of values they can take on by the values bound to those cases already filled. The *Constraints* field of the entity definition specifies the requirements. Of course, at this point no cases have been filled and no constraints apply. The unparsed segment, "the accounts directory the file Data3", is examined in two ways:

a. The first case we try to fill is the direct object which is unmarked. (Figure 1,(g) and (h))
b. The first case we try to fill is one of the marked cases. (Figure 1,(i))

Consider step 4.a. We are attempting to interpret some portion of the segment as the *Object SemanticCase* of the *MoveCommand*. Since the entity definition shows that an *Object* can be an instance of either a *FileObjDesc* or a *DirectoryObjDesc*, we will try each of these in turn (Figures 2,(a) and 3,(a)).

Step 5. Continue step 4.a.; try to parse *Object* as a *FileObjDesc*.

ImperativeCaseFrame-Cases wants to find a *FileObjDesc* in "the accounts directory the file Data3". To do so, it must call **ParseEntity** with a request for a noun phrase that can be interpreted as a *FileObjDesc*. Since there are two *SurfaceForms* for parsing nominal caseframes, each with its own associated strategy, this call to **ParseEntity** results in a SPLIT (Figure 2,(b)) We will examine only the path labelled **NominalCaseFrame-Strat-Canonical** (Figure 2,(c)).

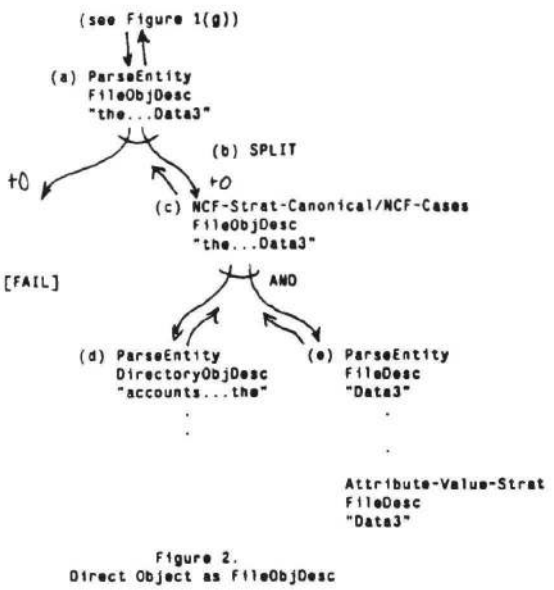


Figure 2.
Direct Object as FileObjDesc

Step 6. Find the *Head* and *Quantifier* of the noun phrase.

As with imperative caseframes, the first action taken to fill a nominal caseframe is to locate the *Head*. **NominalCaseFrame-Strat-Canonical** finds "file" as a possible head and breaks the remainder of the input into prenominal and postnominal segments. The strategy then looks for a quantifier or determiner at the left end of the prenominal segment and finds "the". "Accounts directory the" now constitutes the prenominal segment and "Data3," the postnominal segment. We call **NominalCaseFrame-Cases**, a sub-routine of **NominalCaseFrame-Strat-Canonical**, to begin filling cases from the prenominal segment (Figure 2,(c)).

Step 7. Parse the prenominal segment, "accounts directory the".

The only *FileObjDesc* case we will be able to fill from this segment of the input is *FileDirectory*, an instance of a *DirectoryObjDesc*. After a sequence of calls (Figure 2,(d)) a sub-invocation of **NominalCaseFrame-Strat-Canonical** will return an instance list with a single instance:

```
(IsA DIRECTORY
```

[8] The reader should keep in mind that at each step it is possible to have more than one interpretation of the input. Thus, if our input had been, "Move the file called transfer to dskb," the initial scan for a verb would have resulted in two partial parses -- one catching "move" and the other catching "transfer".

```
Name (accounts))[9]
```

Thus, we return to step 6 (Figure 2,(c)) with one case filled and the word "the" unused.

Step 8. Parse the postnominal segment, "Data3".

We are interested in extending any partial parses created in step 7 by using the postnominal segment to fill any unbound *SemanticCases*. Figure 2,(e) shows that a succession of calls results in filling the *FileName* using the Attribute-Value strategy.

Step 9. Return to step 4, Figure 1,(g).

Steps 7 and 8 produced one consistent set of bindings for two *SemanticCases* in the *MoveCommand* with one word leftover. Although there are unfilled cases, we have run out of input on the right. Thus, the instance list of *FileObjDescs* returned by the call to **ParseEntity** in Figure 2,(a) has only one element:

```
(IsA    FILE
 Name   (Data3)
 Directory  (accounts)
 Description (
       Quantifier (the)))
```

Referring to step 4.a. (Figure 1,(g)), the possible interpretations of the input such that the *Object* is a *FileObjDesc* have been exhausted. It remains to examine what happens when we look for an *Object* that is a *DirectoryObjDesc*. Again, we will call **ParseEntity**, split and suspend one of the calls, and examine the branch labelled **NominalCaseFrame-Strat-Canonical** (Figure 3,(a) through (c)).

Step 10. Continue step 4.a.; try to parse the *Object* as a *DirectoryObjDesc*.

We pick up the *head*, "directory," and the quantifier/determiner case as in step 6. This leaves the word "accounts" in the prenominal segment and "the file Data3" in the postnominal segment. The Attribute-Value strategy finds "accounts" as the *Name*. No other *DirectoryObjDesc* cases can be filled, so this step returns:

```
(IsA DIRECTORY
 Name (accounts)
 Description (
       Quantifier (the)))
```

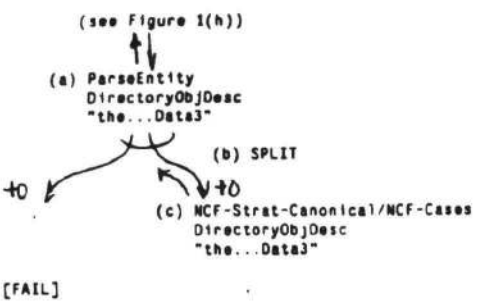


Figure 3.
Direct Object as DirectoryObjDesc

Step 11. Return to step 4.

Consider Figure 4,(a) which corresponds to the "OR" in Figure 1,(f). The computations shown in Figures 2 and 3 have given us two partial parses with the direct object filled; once by a *FileObjDesc* with the word "the" leftover and no input left to fill the other cases of the *MoveCommand* (Figure 4,(b)), and once by a *DirectoryObjDesc* with "the file Data3" leftover (Figure 4,(c)). As we try to extend this second partial parse, we have only marked cases remaining in the *SurfaceForm*. Since there is no marker at the beginning of the remaining segment we have encountered our first violated expectation. The recovery action associated with this failure is to hypothesize the existence of the missing case marker. Thus, for each of the three remaining *SemanticCases* we schedule a continuation that charges three flexibility points for the deviation (Figure 4,(d)). Note that if some other branch of the search tree with cumulative flexibility less than three succeeds in consuming the entire input segment, the branches just spawned will never be reactivated.

---

[9] The *InstanceTemplate* has many more fields in it; only those with a non-nil value are shown.

Step 12. Return to step 4.b. (Figure 4.(e))

Consider what happens when we try filling the marked cases of the *MoveCommand* first (Figure 1.(i) corresponding to Figure 4.(e)); the situation is identical to the one just outlined. For all cases other than the direct object, the segment "the accounts directory the file Data3" must have a left marker. Since it has none, we hypothesize a branch for each marked case at the current flexibility level plus three (Figure 4.(e)).

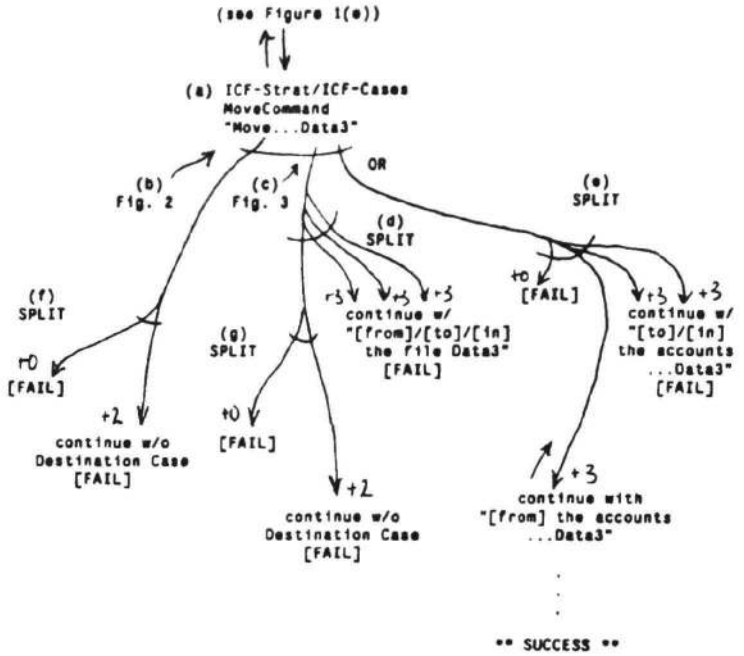


Figure 4.
Continuation of Figure 1 After
Processing Completed by Figures 2 and 3

Step 13. All the possibilities for the *MoveCommand* have been examined.

We have succeeded in finding two ways to fill the cases of the *MoveCommand*. Before we can return the partial parses from **ImperativeCaseFrame-Cases** to **ImperativeCaseFrame-Strat** we must check whether the *Required* cases, as specified by the *Constraints* field, have been filled. Indeed, each of the partial parses is missing the required *Destination* case, a violated expectation. The recovery action associated with this error is to suspend each of these partial parses and charge two flexibility points per missing required case for their continuation (Figure 4,(f) and (g)). Since no parses had all the required cases, the level 0 continuation of **ImperativeCaseFrame-Cases** returns a failure signal to **ImperativeCaseFrame-Strat.**

ImperativeCaseFrame-Strat returns an instance list with a single instance whose *Source* and *Destination* fields are nil. This signifies that the only part of the strategy that succeeded at level 0 was finding the verb. Since there is unused input, the top-level of MULTIPAR interprets this instance as a failure and signals this to the control.

Step 14. Exhaust level 0 of the agenda looking for a non-deviating parse.

The control structure takes over and continues in turn each branch suspended at level 0. Those containing requests for imperatives (Figure 1,(j)) fail immediately as in step 2. The other level 0 branches were left suspended by the SPLITs in Figures 2 and 3; these also fail.

Step 15. Exhaust levels 1 and 2 of the agenda.

Having tried all the branches in level 0 without success, the flexibility level is incremented and the control structure tries to choose a path suspended at level 1. Our example did not spawn any level 1 branches (single spelling corrections), so the flexibility level is incremented again. There are two branches at level 2, both in the same predicament (Figure 4, (f) and (g)); each has a missing required case and leftover input. If there had been no leftover input (as in "Move foo"), they would have succeeded at this level.[10] However, since no further recovery actions apply, each of these branches fails without adding to the control tree.

Step 16. The control increments the flexibility level to 3.

There are two sets of branches at this level:

a. The *Object* case is filled and the missing marker has been hypothesized before "the file Data3". (Figure 4,(d))
b. No cases are filled and the missing marker has been hypothesized before "the accounts directory the file Data3". (Figure 4,(e))

[10] Although consuming the entire input would have guaranteed success, note that if some branch with cumulative flexibility of 0 or 1 had succeeded, these branches would never have been retried.

Step 17. Continue 16.a. (Figure 4.(d))

The *Object* case of the *MoveCommand* has been bound to a *DirectoryObjDesc* with the *Name* field bound to "(accounts)". Hypothesizing the appropriate kind of marker for each of the remaining cases gives:

a. *Source*: Move the accounts directory [from] the file Data3.
b. *Destination*: Move the accounts directory [to] the file Data3.
c. *Location*: Move the accounts directory [in] the file Data3.

The single recovery action of hypothesizing a missing marker is not enough for any of these branches to succeed. Each would be rescheduled at least one more time. If MULTIPAR allowed the control structure to search the space indefinitely, 17.a. would eventually succeed at flexibility level 8 (3 points for a missing marker, 3 points for a constraint violation[11], and 2 points for a missing required case). 17.b. would eventually succeed at level 6 (1 missing marker, 1 constraint violation) and 17.c. at level 8 (1 missing marker, 1 missing required case and 1 constraint violation).

Step 18. Continue 16.b. (Figure 4.(e))

There are three branches remaining, one for each of the marked cases in the *MoveCommand*, with no cases yet filled. Allowing indefinite expansion, the following would occur:

Source:

a. Move [from] the accounts directory the file Data3.
   eventually succeeds at +5; 3 for missing marker, 2 for missing case
b. Move [from] the accounts directory [to] the file Data3.
   eventually succeeds at +8; 2 missing markers, 1 required case
c. Move [from] the accounts directory [in] the file Data3.
   eventually succeeds at +13; 2 missing case markers, 1 constraint violation and 2 missing required cases

Destination:

a. Move [to] the accounts directory the file Data3.
   *** SUCCEEDS at this level (level 3) ***
b. Move [to] the accounts directory [from] the file Data3.
   eventually succeeds at +8; 2 missing case markers and 1 missing required case
c. Move [to] the accounts directory [on] the file Data3.
   eventually succeeds at +13; same as *Source* c.

Location:

a. Move [in] the accounts directory the file Data3.
   eventually succeeds at +5, 1 missing marker and 1 missing case
b. Move [in] the accounts directory [to] the file Data3.
   eventually succeeds at +8, 2 markers, 1 case
c. Move [in] the accounts directory [in] the file Data3.
   eventually succeeds at +13, same as *Source* c.

Step 19. A successful path is found at level 3.

Hypothesizing the existence of a marker for the *Destination* enables **ImperativeCaseFrame-Strat** to continue the second branch of Figure 4.(e). Now "the accounts directory" can be picked up as the *Destination* and "the file Data3" as the unmarked direct object. Since both required cases are bound and no input remains, MULTIPAR return the following instance as its representation of the input:

```
(Action MOVE
 Deviations (MissingMarker Destination)
 Source (
       IsA FILE
       Name (Data3)
       Description (
              Quantifier (the)))
```

[11] In *Constraints:(Object DirectoryObjDesc > Source LogicalDeviceObjDesc)*.

```
Destination (
        IsA FILE
        Name (Data3)
        Directory (accounts)
        Description (
                Quantifier (the))))[12]
```

5. Conclusion

MULTIPAR is not a detailed cognitive model of human language processing. It is an attempt to emulate the performance of humans in comprehending natural language utterances that deviate from strict grammatical standards. MULTIPAR uses multiple parsing strategies, and is driven by a "grammar" of descriptions of entities relevant to the domain of discourse. The multiple strategies are able to interpret the entity definitions in a variety of ways. Some of the ways depend on the surface language constituents being in the grammatically correct place. Other ways, though more computationally expensive, relax the grammatical constraints, and are thus able to handle grammatically deviant input. To control the potential for exponential growth in the search space of a parser that accepts ungrammatical as well as grammatical input, MULTIPAR incorporates a control mechanism that allows possible parses to be explored in order of increasing degree of ungrammaticality. All of these features of MULTIPAR are essential to its performance as a robust natural language parser.

References

1. Boggs, W. M. and Carbonell, J. G., Monarch, I., and Kee, M., "The DYPAR-I Tutorial and Reference Manual," Tech. report, Carnegie-Mellon University, Computer Science Department, 1985.
2. Carbonell, J. G., "Towards a Self-Extending Parser," *Proceedings of the 17th Meeting of the Association for Computational Linguistics*, 1979, pp. 3-7.
3. Carbonell, J. G. and Hayes, P. J., "Dynamic Strategy Selection in Flexible Parsing," *Proceedings of the 19th Meeting of the Association for Computational Linguistics*, 1981.
4. Carbonell, J. G. and Hayes, P. J., "Recovery Strategies for Parsing Extragrammatical Language," *American Journal of Computational Linguistics*, Vol. 9, No. 3-4, 1983, pp. 123-146.
5. Carbonell, J. G. and Hayes, P. H., "Robust Parsing Using Multiple Construction-Specific Strategies," in *Natural Language Parsing Systems*, L. Bolc, ed., Springer-Verlag Publishers, 1985.
6. Dejong, J. F., *Skimming Stories in Real Time: An Experiment in Integrated Understanding*, PhD dissertation, Yale University, May 1979.
7. Granger, R., "FOUL-UP: A Program that Figures Out Meanings of Words from Context," *Proceedings of IJCAI-77*, 1977, pp. 172-178.
8. Hayes, P. J., "Entity-Oriented Parsing," *COLING84*, Stanford University, July 1984.
9. Hayes, P. J. and Carbonell, J. G., "Multi-Strategy Parsing and its Role in Robust Man-Machine Communication," Tech. report CMU-CS-81-118, Carnegie-Mellon University, Computer Science Department, May 1981.
10. Kwasny, S. C. and Sondheimer, N. K., "Ungrammaticality and Extra-Grammaticality in Natural Language Understanding Systems," *Proc. of 17th Annual Meeting of the Assoc. for Comput. Ling.*, August 1979, pp. 19-23.
11. Kwasny, S. C. and Sondheimer, N. K., "Relaxation Techniques for Parsing Grammatically Ill-Formed Input in Natural Language Understanding Systems," *American Journal of Computational Linguistics*, Vol. 7, No. 2, 1981, pp. 99-108.
12. Lebowitz, M., *Generalization and Memory in an Integrated Understanding System*, PhD dissertation, Yale University, Oct. 1980.
13. Weischedel, R. M. and Sondheimer, N. K., "Meta-Rules as a Basis for Processing Ill-formed Input," *Computational Linguistics*, Vol. 10, 1984.
14. Weischedel, R. M. and Black, J., "Responding to Potentially Unparseable Sentences," *American Journal of Computational Linguistics*, Vol. 6, 1980, pp. 97-109.

[12] The reader may have noticed that the directions for filling in the *MoveCommand*'s *InstanceTemplate* would give different values for the *Destination* fields. The *InstanceTemplate* shown in section 3 has been simplified for illustrative purposes. The real *InstanceTemplate* has considerably more complex directions for filling the fields; from those directions (which include conditionals that test other fields) the instance shown above would be constructed.

TOWARDS A COMPUTATIONAL THEORY OF HUMAN DAYDREAMING*

Erik T. Mueller
Michael G. Dyer

Artificial Intelligence Laboratory
Computer Science Department
University of California
Los Angeles, CA 90024

ABSTRACT

This paper examines the phenomenon of *daydreaming:* spontaneously recalling or imagining personal or vicarious experiences in the past or future. The following important roles of daydreaming in human cognition are postulated: plan preparation and rehearsal, learning from failures and successes, support for processes of creativity, emotion regulation, and motivation.

A computational theory of daydreaming and its implementation as the program DAYDREAMER are presented. DAYDREAMER consists of 1) a scenario generator based on relaxed planning, 2) a dynamic episodic memory of experiences used by the scenario generator, 3) a collection of personal goals and control goals which guide the scenario generator, 4) an emotion component in which daydreams initiate, and are initiated by, emotional states arising from goal outcomes, and 5) domain knowledge of interpersonal relations and common everyday occurrences.

The role of emotions and control goals in daydreaming is discussed. Four control goals commonly used in guiding daydreaming are presented: rationalization, failure/success reversal, revenge, and preparation. The role of episodic memory in daydreaming is considered, including how daydreamed information is incorporated into memory and later used. An initial version of DAYDREAMER which produces several daydreams (in English) is currently running.

1. INTRODUCTION

Daydreaming is the spontaneous human activity of recalling or imagining personal or vicarious experiences in the past or future. Although sometimes viewed as a useless distraction from the task at hand, we postulate the following important roles of daydreaming in human cognition:**

- Daydreaming supports *planning for the future.* The anticipation of possible future situations allows the *formation of desirable responses* to those situations in advance and thus *improves efficiency.* By assessing the consequences of alternative courses of action in advance, daydreaming assists in *decision-making.* Future daydreaming also provides a *rehearsal* function to increase accessibility of responses and the skill with which they can be performed.
- Daydreaming supports *learning from successes and failures.* Examination of alternative actions in a success or failure experience allows one to *learn planning strategies* to be remembered for use in future similar situations. In addition, daydreaming allows the ongoing *reinterpretation of past experiences* in light of new information or if there was insufficient time to digest an experience when it occurred.
- Daydreaming supports processes of *creativity.* The *generation of fanciful possibilities* can lead to the discovery of new and useful solutions to a problem. While daydreaming about one thing, it is possible to stumble into a solution to another problem, i.e., *fortuitous recognition of analogies* among problems is possible. Daydreaming occurs in the context of an episodic memory which is constantly subject to *revision.* Each time a problem is examined, new information may be available that will enable a better, different, or more creative solution. Ideas generated while daydreaming often provide the initial *inspiration* for a creative work and further daydreaming, e.g., of success or praise from

* The work reported here was made possible in part by a grant from the Keck Foundation, with matching funds from the UCLA School of Engineering and Applied Sciences. The first author was also supported in part by an Atlantic Richfield Doctoral Fellowship and the second author by an IBM Faculty Development Award.

** See Mueller and Dyer (1985) for a more detailed discussion of the functions of daydreaming, both for humans and intelligent computer systems.

others, may increase the *motivation* to realize it.

- Daydreaming supports *emotion regulation*. For example, upon a failure experience, daydreaming enables one to *feel better* or *feel worse* depending on success or failure in rationalizing that experience. Fear associated with a future event may be reduced if one daydreams about effective plans to succeed in that event, or increased if daydreams of likely failure result.

1.1 Obtaining Daydreaming Protocols

Daydreaming is unlike most other human activities which artificial intelligence researchers attempt to model, since it is a behavior which is not manifested externally. The fact that there is no apparent "I/O" while a person is daydreaming presents several problems for us: How do we know when a person is daydreaming? Given that a person is daydreaming, how can we find out what that person is daydreaming about?

Several methods for gathering data about daydreaming have been used in the past. Varendonck (1921, pp. 215-216) used a form of *retrospective report* in which he first recalled the final portion of a daydream and then worked his way backward. Klinger (1971) and Pope (1978) used the technique of *thinking aloud* in which subjects would verbalize their stream of thought as it occurred. In the method of *thought sampling* used by Klinger (1978), subjects would carry a beeper with them. When the beeper sounded at a random time, subjects would fill out a questionnaire asking them to describe their most recent thoughts.

Retrospective reports are generated by recalling a memory trace of the stream of thought which is laid down in episodic memory (Tulving, 1972) during daydreaming. Although this memory trace fades with time, immediate transcription will produce fairly accurate recall. Ericsson and Simon (1984) argue that for cognitive processes of intermediate duration, retrospective reports will give the same information as thinking-aloud protocols, provided certain criteria are met. However, experiments conducted by Pope (1978) have suggested that thinking-aloud protocols of daydreaming may inhibit the subject or slow down and thus modify the stream of thought.

1.2 Example Episode and Resulting Daydreams

In order to investigate the phenomenon of daydreaming, transcripts of daydreams have been obtained from a variety of sources. Wherever possible, we have sought immediate retrospective reports. The discussions which follow will refer to the episode below and resulting daydreams, which are an edited composite of several daydream transcripts:

NUART-EPISODE

Last night I went alone to the Nuart Theater to see a film. This actress whom I've always had a crush on also came alone and happened to sit down near me. I recognized her and started a conversation. We talked quite a while. At one point, she mentioned the obviously well-known director of the film we were about to see and I didn't know the name. I was embarrassed. I finally asked her if she'd like to go have a drink after the film. She said she had other plans. I was disappointed.

NUART-DAYDREAM1

I feel embarrassed for not knowing who the director was. I should've pretended to know who the director was. In the future I'll read up on the film before I go.

NUART-DAYDREAM2

I'm disappointed that she didn't want to go out with me. She might change her mind if I were to ask her out again. I would call her up, but I don't have her telephone number. I should've asked for her telephone number when I had the chance. I imagine that I asked for her telephone number and she gave it to me. Then later I call her up and she accepts my offer for a date. I regret not having asked for her number. But she's a movie star and out of my league. I feel better because no matter what I might have done, she wouldn't have gone out with me.

NUART-DAYDREAM3

I'm disappointed that she didn't want to go out with me. I imagine that she accepted my offer and we soon become a pair. I help her when she has to rehearse her lines. I go to the studio to watch her work. When she has to do a film in France, I drop my work and travel there with her. As we continue to travel, I begin to miss my work. I become unhappy and feel unfulfilled. She loses interest in me, because I have nothing to offer her. It's good that I didn't get involved with her, because it would've led to disaster. I feel less disappointed that she didn't accept my offer.

NUART-DAYDREAM4

I'm angry that she didn't accept my offer to go have a drink. I feel rejected. I imagine that I pursue an acting career and become a star even more famous than she is. She remembers

meeting me a long time ago in a movie theater and calls me up. I'm glad she admits she was wrong about me. I go out with her, but now she has to compete with many other women for my attention. I eventually dump her.

1.3 The DAYDREAMER Program

DAYDREAMER is a computer program designed to implement and test a computational theory of daydreaming. The program operates in two modes: *daydreaming mode* and *performance mode*. In daydreaming mode, the program daydreams continuously until interrupted. Performance mode allows the program to demonstrate that it has learned from daydreaming. DAYDREAMER takes as input simple situational descriptions, such as `accidentally meeting a movie star`, or `being fired from one's job`, and produces as output 1) actions that it would perform in the given situation, and 2) daydreams, all in English. DAYDREAMER learns as it daydreams by indexing daydreams, planning strategies, and future plans into memory for future use.

DAYDREAMER is composed of:

- **a scenario generator** consisting of a *planner* (Fikes & Nilsson, 1971; Meehan, 1976) and *relaxation rules*,
- **a dynamic episodic memory** (Tulving, 1972; Kolodner, 1984) of experiences used by the scenario generator,
- a collection of *personal goals* (Maslow, 1943; Schank & Abelson, 1977) and **control goals** which guide the scenario generator,
- **an emotion component** in which daydreams initiate, and are initiated by, emotional states arising from goal outcomes, and
- **domain knowledge** of interpersonal relations and common everyday occurrences.

In the sections which follow, we describe the above components. We should note that our theory is *not* intended to account for 1) mental imagery or the quasi-sensory experiences which are often a part of daydreaming (Singer & Antrobus, 1972; Singer, 1975), 2) the altered state of consciousness (called "fore-consciousness" by Varendonck, 1921) which often accompanies daydreaming, and 3) the subjective "feeling" of consciousness; see, for example, Nagel (1974) and Dennett (1978) for a discussion of the philosophical problems with such an endeavor.

2. THE ROLE OF EMOTIONS AND CONTROL GOALS IN DAYDREAMING

How and when are daydreams triggered? Given that many differing daydream sequences are possible at any given moment, what determines the way in which a daydream unfolds? We postulate a set of goals, called *control goals*, which are activated in part by emotions, and which both help trigger and direct daydreaming. Once a control goal is activated, the *scenario generator*, discussed in Section 3, generates a sequence of events according to that control goal. The purpose of control goals is generally to provide helpful modification of emotional state in the short run and to help achieve personal goals in the long run. Not all control goals serve both functions. Some in fact serve one function while ignoring or even harming the other.

The study of daydreaming is interrelated with the study of emotion. Previous work on simulation of human emotions has involved modeling the comprehension by a reader of the emotional reactions of story characters (Dyer, 1983b) or modeling emotional responses to real-world situations and the influence of emotional state on subsequent behavior (Pfeifer, 1982). However, a complete model of human emotions must not only account for the relationship between emotions and events in the external world, but also between emotions and *internal events*, i.e., imagined or expected outcomes (Abelson, 1981), or daydreams. Instead of simply modeling the single emotional response to an event, one must model the entire *sequence* of responses and daydreams resulting from an event, perhaps at last resting on a final emotion.

In DAYDREAMER, emotions activate control goals which result in daydreaming. Daydreaming in turn results in modification of emotional state (in part as a result of the success or failure of control goals), directing the future course of daydreaming by causing the activation of other control goals. Thus we have a feedback loop in which 1) emotions trigger daydreams and 2) daydreams modify existing emotions and trigger new emotions, which trigger new daydreams, and so on. An important part of our research involves specifying an intuitive set of control goals and their interactions with scenario generation and emotion processes. In this section, we discuss four control goals which commonly appear in daydreaming:

- **Rationalisation:** generating reasons for why an outcome is satisfactory to the daydreamer in order to reduce negative emotions and maintain self esteem.
- **Revenge:** reducing negative emotions through imagined retaliations after a goal has been thwarted

by another.

- **Failure/Success Reversal:** altering reality by imagining scenarios in which failures were prevented or in which successes failed to come about in order to learn future planning strategies.
- **Preparation:** generating hypothetical future scenarios in order to learn planning strategies and/or specific actions to be used in possible future situations.

2.1 Rationalisation

In DAYDREAMER, recall of a goal success produces a positive emotion, while recall of a failure produces a negative emotion (Dyer, 1983b). Negative emotions resulting from recalled failures activate *rationalization* control goals. Rationalization, which involves finding a reason why a particular negative outcome is satisfactory, serves to reduce the intensity of a negative emotion associated with that outcome. In effect, failures result in a form of "cognitive dissonance" (Festinger, 1957) which must somehow be reduced. The scenario generator realizes a rationalization control goal through planning. There are many ways of rationalizing a failure.

In NUART-DAYDREAM3, the method for rationalization is to imagine that the goal in question succeeded instead of failed, and then imagine that this goal success in fact leads to a worse goal failure. Another method of rationalizing failures involves playing out the consequences of a goal failure and discovering that a more important goal success may be achieved.

Another method for rationalization is suggested by attribution theory (Heider, 1958; Weiner & Kukla, 1970). An *attribution* is the cause or causes that one attributes to a past success or failure. Attributions have an impact on emotional state. For failures, finding an *external attribution*, i.e., attributing the failure to another person, lack of luck, environmental factors, lack of ability, lack of effort, or fatigue, will often reduce a negative emotion resulting from blaming oneself for a failure. Thus external attribution is another possible plan for rationalization.

2.2 Revenge

The emotion of ANGER results when someone else causes DAYDREAMER a goal failure. The *revenge* control goal is activated upon presence of ANGER and serves the function of substituting a positive emotion for the negative emotion associated with the failure. An example is NUART-DAYDREAM4, where DAYDREAMER imagines that the actress now pursues him but he rejects her. Activation of control goals such as rationalization and revenge in response to an emotion, rather than to an abstract situation, is supported by experiments by Weiner (1980), which found that the presence of the emotion was required.

2.3 Failure/Success Reversal

Another control goal which is initiated upon a negative emotion resulting from a recalled failure is that of *failure reversal*, whose objective is to generate a means by which the failure could have been prevented and a success achieved. This control goal enables learning through the abstraction of generated alternatives to planning strategies for use in future similar situations.

In NUART-EPISODE, there are two failures: the failure of a social regard preservation goal resulting in the emotion of EMBARRASSMENT, and the failure of the goal to go out with the movie star resulting in emotions of DISAPPOINTMENT and REJECTION. Several failure reversal daydreams result. NUART-DAYDREAM1 involves having pretended to know the name of the director. NUART-DAYDREAM2 involves having asked the star for her telephone number and having received it.

Positive and negative emotional response to goal success and failure occur during imagined episodes just as during real episodes. Thus when DAYDREAMER imagines a way of avoiding a failure and achieving a success, a positive emotion results initially. However, it is often followed immediately by the renewal and intensification of the negative emotion associated with the failure. Thus failure reversal actually has a negative emotional function: for the sake of learning, it allows a negative emotion to be intensified. As a consequence, DAYDREAMER generates: "I regret not having asked for her number" in NUART-DAYDREAM2.

A *success reversal* control goal is sometimes pursued upon a recalled success. Why do people imagine failures as well as successes in their daydreams? It is well known that people learn from actual failures; see, for example: (Schank, 1982; Dyer, 1983a; Dolan and Dyer, 1985). It is reasonable to expect that it is possible to learn from imagined failures as well. By noting the causes of daydreamed failures in memory, one may avoid similar failures in the future.

2.4 Preparation

The *preparation* control goal is activated when thinking about a possible future situation or upcoming event which is emotionally charged. This control goal serves the following functions. First, it al-

lows one to be prepared, i.e., it increases the chances of success in the future event. Second, it serves to reduce negative emotions such as anxiousness if effective plans which lead to success are found. An example of plan preparation and rehearsal occurs in NUART-DAYDREAM2, where DAYDREAMER rehearses the plan of asking the actress out and spots the planning error of not having asked for her number.

A striking example of the plan rehearsal aspect of daydreaming occurred in a case where a friend (F) of the second author was stung by a bee, whereupon F experienced a strong allergic reaction to the bee venom. F was rushed to the hospital and treated in time to avoid anaphylactic shock. F stated that, for an entire year after this experience, F would spontaneously daydream about being stung. These daydreams were not repetitions of the original event, but began by imagining being stung in various situations, e.g., by a pool, at the beach, while at a party, while biking, while alone at home, etc. Each daydream consisted of imagining what F would do in case F's bee sting kit was inaccessible, in case the phone was out of order, in case the car broke down, and so on. F claims now to have rehearsed plans for a large number of hypothetical circumstances, e.g., using ice from the refrigerator, knowing where a hospital is and driving at breakneck speed to a hospital before collapsing, etc. This case is rather dramatic because the original goal threat was to a very high-priority goal, i.e., a health preservation goal. The function of daydreaming in this case was to rehearse and examine plans in imagined situations. Clearly, daydreaming here provides an advantage over planning systems which only initiate planning when posed with the actual occurrence of a goal threat or goal failure. Preparation-based daydreams tend to be triggered by emotions of fear. Thus, generating this class of daydreams involves specifying a process model of what might be called "worrying" when observed in people.

3. REALIZING A DAYDREAM THROUGH SCENARIO GENERATION

How are the various imaginative sequences of events which make up daydreaming generated? In DAYDREAMER, the *scenario generator* generates daydreams in response to and under the guidance of control goals. We propose that the basic mechanism for scenario generation is *planning* (Fikes & Nilsson, 1971; Sacerdoti, 1974; Meehan, 1976), i.e., generating a sequence of actions necessary to achieve a goal. However, the scenario generator differs from traditional planning mechanisms in several fundamental ways. First, instead of relentlessly pursuing a given goal, the scenario generator operates as an ongoing process under numerous, often conflicting, personal goals. It is possible to start planning for one goal only to abandon that goal in pursuit of another. Second, the scenario generator incorporates relaxations which enable it to generate scenarios which are fanciful, i.e., non-realistic solutions to problems. Third, the scenario generator incorporates an episodic memory of experiences which influences the planning process 1) by providing the experiences which are the subject matter of daydreaming and which trigger activities such as attempting to learn from a failure or rationalization, and 2) by providing a source of knowledge for use by the scenario generator in generating possible events or sequences of events. In this section, we discuss the first two above aspects of the scenario generator. The third aspect, the role of episodic memory, is discussed in Section 4.

3.1 Daydreaming Is Influenced by Multiple Personal Goals

In a wish-fulfillment daydream (Freud, 1908; Varendonck, 1921), the course that a daydream takes is determined by a wish of the daydreamer. Where do wishes come from? In addition to control goals, DAYDREAMER has a large number of *personal goals*, including: health, food, sex, friendship, love, possessions, self esteem, social esteem, enjoyment, and achievement. How does the scenario generator choose which goals or wishes to focus upon at any given point? Minsky (1977) espoused a computational theory of the mind as a society of intercommunicating and conflicting entities. He described a child playing blocks: Internally, the WRECKER in the child wants to destroy the tower being built by the BUILDER. Meanwhile, the I'M-GETTING-HUNGRY entity is growing in strength. As the control of the BUILDER weakens, the child destroys the tower and gets up to go home and eat. Schank and Abelson (1977) constructed an elaborate taxonomy of goal states and interactions while Wilensky (1978) showed how knowledge of goal competition and goal conflict is needed to understand stories involving multiple, interacting narrative characters.

In DAYDREAMER, goals are organized into a *goal tree* (Carbonell, 1980) which specifies the relative importance of each of the goals at any point in time. When confronted with several competing goals, the scenario generator pursues a course of action which leads to the satisfaction of the more important goal. For instance, in NUART-DAYDREAM3, DAYDREAMER imagines that success in the relationship leads to loss of his job. Suppose, however, that he were beginning to grow tired of his job. In this case, daydreaming that he goes to France might more likely lead to imagining that he finds a better job and begins an entirely new career.

3.2 Relaxation of Constraints

A collection of relaxation rules allow the generation of fanciful scenarios. In particular, the following constraints may be relaxed in planning:

- **Behavior of others:** DAYDREAMER imagines that the movie star accepts his offer in NUART-DAYDREAM3.
- **Self attributes:** one might imagine being an olympic athlete or being a famous movie star.
- **Physical constraints:** one might imagine being invisible or being able to fly.
- **Social constraints:** one might imagine starting a food fight at a fancy restaurant.

The scenario generator does not always employ relaxation rules. The level of relaxation can be varied depending on what control goals are currently active. For example, if the daydreamer is pursuing a failure-reversal control goal whose objective is to generate realistic ways of having prevented a failure (e.g., NUART-DAYDREAM1), then the relaxation level is set on LOW. If, however, DAYDREAMER is pursuing a revenge control goal whose objective is to generate a fanciful retaliation (e.g., NUART-DAYDREAM4), then the relaxation level is set on HIGH. It is also important to *assess* the level of relaxation which led to a given scenario in order to avoid raising expectations, since raised and failed expectations lead to negative emotions. The result of such assessment can be seen in NUART-DAYDREAM2, where DAYDREAMER imagines "But she's a movie star and out of my league ... no matter what I might have done, she wouldn't have gone out with me."

In standard planning systems such as ABSTRIPS (Sacerdoti, 1974), relaxation of operator preconditions is employed to reduce the amount of searching needed to solve a problem. However, in daydreaming, relaxation rules may actually *increase* the amount of searching that is done, since such rules enable the generation of numerous imaginative situations which are not realistic methods of achieving current personal goals. Still, such explorations may prove useful in the future, may enable the fortuitous discovery of realistic solutions, or may simply serve some other function such as emotion regulation.

4. THE ROLE OF EPISODIC MEMORY IN DAYDREAMING

After a daydream, what remains in memory? How is information which is incorporated into memory during daydreaming used in the future? That is, how does one *learn* from daydreaming? First of all, one may be skeptical of the ability for humans to remember their daydreams and thus doubt that it is truly possible to learn from them. However, the mere fact that it is possible to obtain transcripts of daydreams from people proves that some daydreams are remembered, at least for a short while. Moreover, many subjects have reported, long after the fact, daydreams that they remember having had (Singer & Antrobus, 1972). Varendonck (1921, p. 327) reports near complete recall of his daydreams during writing: "when I am composing letters I often afterwards write them almost exactly as I worded them in my phantasy." Even subtle memory modifications may occur during daydreaming. Neisser's (1982) analysis of the testimony of John Dean shows that he often remembered conversations in terms of his own fantasy about how those conversations *should* have been, rather than how they actually were.

The *dynamic episodic memory* (Tulving, 1972; Schank, 1982) of DAYDREAMER is its long-term memory of personal or vicarious experiences and daydreams. This memory is called *dynamic* because it is constantly being modified during daydreaming. Thus not only are actual experiences available for use at any point, but so are previously daydreamed ones. This gives rise to a dynamic behavior dependent on previous external *and internal* experiences.

DAYDREAMER incorporates the following information into memory as it daydreams:

- entire daydreams,
- future plans or actions formed during daydreaming, and
- planning strategies formed during daydreaming.

How is information indexed in episodic memory so that it is available at an appropriate time in the future? Once information is retrieved, how is it applied to the current situation? Our work on mechanisms for storage, organization, retrieval, generalization, and application of experiences in episodic memory builds on previous work by Kolodner (1984), Schank (1982), and Anderson (1983). Episodes (both daydreams and personal or vicarious experiences) are organized in and retrieved from dynamic episodic memory according to surface-level similarities as well as Plot Units (Lehnert, 1982), emotions and abstract themes (Dyer, 1983a). The decision to organize episodes by emotions is also partly supported by the work of Varendonck (1921, p. 192) and Bower and Cohen (1983).

4.1 Indexing Entire Daydreams in Episodic Memory

How are entire daydreams indexed in memory for later use? Suppose that at some point in the past, DAYDREAMER had been interviewed for a job he wanted, was turned down, and had the following daydream:

JOB-DAYDREAM

I'm angry at the interviewer for not hiring me. I imagine that years from now I am president of a large company. The interviewer has lost his job and comes to me for employment. I tell him that he doesn't have the necessary qualifications.

In order to recall this daydream in the future, DAYDREAMER indexes it under the RETALIATION Plot Unit,* which represents the abstract situation of a person A achieving a positive outcome by causing a negative outcome for B, where B had originally caused a negative outcome for A. DAYDREAMER also indexes the daydream under REJECTION, which is a negative emotion of a person A resulting from the failure of A's goal to activate a positive relationship R with a person (or group) B, where that failure is caused by B.

4.2 Recalling and Using Recalled Episodes

Daydreams have a certain amount of coherence to them. We believe that this coherence is partially provided by abstract knowledge structures such as Plot Units. Once a control goal is activated, the scenario generator must select and execute a plan to achieve it. Plans for achieving control goals may often be captured by Plot Units, e.g., one plan for the revenge control goal is RETALIATION. Realization of plans for achieving control goals expressed as Plot Units may be simplified through reminding of appropriate episodes indexed by those Plot Units.

Plans for achieving the rationalization control goal may also be expressed in terms of Plot Units. One plan is expressed by the MIXED BLESSING Plot Unit, employed in NUART-DAYDREAM3, which represents the abstract situation of a goal success leading to the failure of another goal. Here the MIXED BLESSING Plot Unit is realized as a fantasy in which dating the actress leads to eventual failure of the relationship and DAYDREAMER's career. Another plan for rationalization is captured by the SUCCESS BORN OF ADVERSITY Plot Unit, which represents the abstract situation of a goal failure leading inadvertently to the success of another goal. Here we could imagine a daydream in which, as a result of being turned down, DAYDREAMER imagines drowning his sorrows at a bar and by chance meeting a more beautiful actress.

Once JOB-DAYDREAM is indexed in memory, when might it later be recalled? Consider the situation of NUART-EPISODE in which DAYDREAMER feels REJECTION and ANGER toward the movie star. The ANGER activates a revenge control goal. Next, the scenario generator will activate the RETALIATION Plot Unit in attempting to realize this control goal. JOB-DAYDREAM, which is indexed under RETALIATION and REJECTION, will then be recalled. The scenario generator now adapts this daydream to the situation of NUART-EPISODE in order to achieve the active revenge control goal and the following daydream is produced:

NUART-DAYDREAM5

I'm angry that the actress didn't want to go out with me. I imagine that years from now I am an influential director. The actress is having trouble finding work and comes to read for a part in my next film. I tell her that I can't use her.

A recalled experience or daydream is adapted to the current situation through *analogy*. The major correspondences which make up the analogy may be identified by the abstract structures used to recall the episode. In our example, the interviewer may be seen to be analogous to the actress, since they occupy the same roles in their respective REJECTION and RETALIATION structures. Similarly, the EMPLOYER-EMPLOYEE relationship of JOB-DAYDREAM can be seen to be analogous to the FRIENDS relationship of NUART-EPISODE. The scenario generator must now complete a scenario analogous to JOB-DAYDREAM in the NUART-EPISODE context. Analogous details are filled in using the same planning knowledge that the scenario generator would need to generate NUART-DAYDREAM5 from scratch. However, in this case, less effort is required (Carbonell, 1983).

The use of recalled episodes is not limited to the generation of entire daydream sequences. Episodic memory is also useful for suggesting possible continuations, or next events, of an ongoing scenario. For example, when DAYDREAMER imagines he is going out with a movie star in NUART-DAYDREAM3, he is reminded of the time he helped an actor friend rehearse for a play. By analogy DAYDREAMER produces the following scenario event in NUART-DAYDREAM3: "I help her when she has to rehearse her lines." Similarly, a reminding of a recent magazine article about an actress going to France to shoot a film leads DAYDREAMER to imagine: "When she has to do a film in France, I drop my work and travel there with her."

---

* Plot Units were developed by Lehnert (1982) to represent narrative plots. Plot Units consist of abstract configurations of positive and negative outcomes linked to mental states by initiation, termination and coreference links. By combining Plot Units, larger plot structures can be created dynamically.

5. CURRENT STATUS

An initial version of the DAYDREAMER program has been constructed using GATE (Mueller & Zernik, 1984), an integrated set of graphical artificial intelligence development tools for the T language (Rees, Adams, & Meehan, 1984), a dialect of Scheme running on Apollo Domain workstations. GATE includes a graphical knowledge representation system, a demon-based programming language, and a logic programming system.

DAYDREAMER currently 1) participates in NUART-EPISODE by receiving and performing actions in response to input phrases such as `You are near Debra Winger` and `She tells you that she does not want her and you to go out on a date`, 2) indexes the episode into memory under the DENIED REQUEST Plot Unit, 3) generates versions of NUART-DAYDREAM2, NUART-DAYDREAM3, and NUART-DAYDREAM4 in English produced by a simple recursive descent generator, indexing the daydreams into memory via Plot Units and emotions, and indexing into memory the planning strategy formed during NUART-DAYDREAM2, i.e., not to forget to ask for someone's telephone number, and 4) participates in another episode in which it demonstrates that it has learned the above planning strategy. An abbreviated trace of DAYDREAMER as of June 1985 is provided in Appendix A.

6. FUTURE WORK AND CONCLUSIONS

DAYDREAMER is being extended to enable it to 1) participate in a larger variety of input episodes, 2) generate more daydreams and possible scenarios in response to each episode, 3) incorporate many daydreams, planning strategies, and future plans into memory, and 4) demonstrate the use of recalled daydreams, planning strategies, and future plans in future internal (daydreaming) and external behavior. The use of analogy and episodic memory in scenario generation and learning of planning strategies will be investigated in greater detail. We intend for DAYDREAMER to be able to daydream continuously, stopping only to receive new experiences.

Numerous problems and issues have faced us in the ongoing implementation and design of DAYDREAMER. These include:

(1) The representation of knowledge in the interpersonal domain, e.g., representing conceptualizations for: "having a crush" on someone, "rehearsing lines", "getting involved with someone", "admitting being wrong" about someone, and so on.

(2) The interaction of processes of scenario generation, relaxed planning, emotional triggers, control goal activation, Plot Unit selection, and episodic memory remindings.

(3) The separation of personal episodes experienced by DAYDREAMER, from vicarious episodes, from imagined episodes already daydreamed and stored in episodic memory.

How and to what extent is reality separated from imagination? Mentally healthy individuals rarely confuse what they have imagined from what they have experienced. How easy is it for our daydreams and imaginings to subtly alter our interpretation of past events? This issue will become all the more pressing as DAYDREAMER's memory of input and daydreamed experiences grow and as we try to make DAYDREAMER's dreams available to systems concerned with such creative tasks as conversation, invention, and story invention.

Through the continuing design and implementation of DAYDREAMER we have been exploring a computational theory underlying daydreaming. This theory is based on a process model which specifies how emotions, scenario generation, planning, plot structures, themes, personal goals, control goals, episodic memory, and analogy processes interact with one another. We have argued that, far from being a useless epiphenomenon, daydreaming serves an important cognitive function in plan preparation and rehearsal, learning from failures and successes, support for processes of creativity, emotion regulation, and motivation. Truly intelligent computers should not be left in a "diddle loop" or turned off when unused, but engaged in daydreaming like ourselves.

REFERENCES

Abelson, R. P. (1981). Constraint, construal, and cognitive science. In *Proceedings of the Third Annual Conference of the Cognitive Science Society*. Berkeley, CA.

Anderson, J. R. (1983). *The architecture of cognition*. Cambridge, Mass.: Harvard University Press.

Bower, G. H., & Cohen, P. R. (1982). Emotional influences in memory and thinking: data and theory. In Clark, M. S., & Fiske, S. T. (Eds.) *Affect and Cognition: The 17th Annual Carnegie Symposium on Cognition*. Hillsdale, NJ: Lawrence Erlbaum.

Carbonell, J. G. (1983). Learning by analogy: Formulating and generalizing plans from past experience. In R. S. Michalski, J. G. Carbonell, and T. M. Mitchell (Eds.). *Machine learning*. Palo Alto, CA.: Tioga.

Carbonell, J. (1980). Towards a process model of human personality traits. *Artificial Intelligence 15*, 49-74.

Dennett, D. C. (1978). *Brainstorms*. Cambridge, Mass.: MIT Press.

Dolan, C., & Dyer, M. G. (1985). Learning planning heuristics through observation. *Proceedings of the Ninth International Joint Conference on Artificial Intelligence*. University of California, Los Angeles. August 18-24, 1985.

Dyer, M. G. (1983a). *In-depth understanding*. Cambridge, Mass.: MIT Press.

Dyer, M. G. (1983b). The role of affect in narratives. *Cognitive Science. 7,* 211-242.

Ericsson, K. A., & Simon, H. A. (1984). *Protocol analysis: Verbal reports as data*. Cambridge, Mass.: MIT Press.

Festinger, L. (1957). *A theory of cognitive dissonance*. Evanston, Ill.: Row, Peterson.

Fikes, R. E., & Nilsson, N. J. (1971). STRIPS: A new approach to the application of theorem proving to problem solving. *Artificial Intelligence. 2,* 189-208.

Freud, S. (1908). Creative writers and day-dreaming. In Freud, S. *The standard edition of the complete psychological works*. Vol. IX. London: Hogarth, 1962.

Heider, F. (1958). *The psychology of interpersonal relations*. New York: Wiley.

Klinger, E. (1971). *The structure and function of fantasy*. New York: John Wiley & Sons.

Klinger, E. (1978). Modes of normal conscious flow. In K. S. Pope and J. L. Singer (Eds.). *The stream of consciousness*. New York: Plenum.

Kolodner, J. L. (1984). *Retrieval and organizational strategies in conceptual memory: A computer model*. Hillsdale, NJ: Lawrence Erlbaum.

Lehnert, W. G. (1982). Plot units: A narrative summarization strategy. In W. G. Lehnert & M. H. Ringle (Eds.). *Strategies for natural language processing*. Hillsdale, NJ: Lawrence Erlbaum Associates.

Maslow, A. H. (1943). A theory of human motivation. *Psychological Review. 50,* 370-396.

Meehan, J. (1976). *The metanovel: Writing stories by computer*. (Research Report #74). Computer Science Department, Yale University, New Haven, CT.

Minsky, M. L. (1977). Plain talk about neurodevelopmental epistemology. *Proceedings of the Second International Joint Conference on Artificial Intelligence*. MIT, Cambridge, Mass. 1083-1092.

Mueller, E. T., & Dyer, M. G. (1985). Daydreaming in humans and computers. *Proceedings of the Ninth International Joint Conference on Artificial Intelligence*. University of California, Los Angeles. August 18-24, 1985.

Mueller, E. T., & Zernik, U. (1984). *GATE reference manual* (Technical Report UCLA-AI-84-5). Artificial Intelligence Laboratory, Computer Science Department, University of California, Los Angeles.

Nagel, T. (1974). What is it like to be a bat? *The Philosophical Review*. October 1974.

Neisser, U. (1982). John Dean's memory: A case study. In Neisser, U. (Ed.). *Memory observed: Remembering in natural contexts*. San Francisco: W. H. Freeman.

Pfeifer, R. (1982). *Cognition and emotion: An information processing approach* (CIP Working Paper No. 436). Department of Psychology, Carnegie-Mellon University, Pittsburgh, PA.

Pope, K. S. (1978). How gender, solitude, and posture influence the stream of consciousness. In Pope, K. S., and Singer, J. L. (Eds.). *The stream of consciousness*. New York: Plenum.

Rees, J. A., Adams, N. I., & Meehan, J. R. (1984). *The T manual*. Computer Science Department, Yale University, New Haven, CT.

Sacerdoti, E. D. (1974). Planning in a hierarchy of abstraction spaces. *Artificial Intelligence. 5.* 115-135.

Schank, R. C. (1982). *Dynamic memory*. Cambridge: Cambridge University Press.

Schank, R. C., & Abelson, R. P. (1977). *Scripts, plans, goals, and understanding*. Hillsdale, NJ: Lawrence Erlbaum.

Singer, J. L., & Antrobus, J. S., (1972). Daydreaming, imaginal processes and personality: A normative study. In P. W. Sheehan (Ed.). *The function and nature of imagery*. New York: Academic Press.

Singer, J. L. (1975). *The inner world of daydreaming*, New York: Harper & Row.

Tulving, E. (1972). Episodic and semantic memory. In E. Tulving and W. Donaldson (Eds.). *Organization of memory*. NY: Academic Press.

Varendonck, J. (1921). *The psychology of day-dreams*. London: George Allen & Unwin Ltd.

Weiner, B. (1980). A cognitive (attribution)-emotion-action model of motivated behavior: An analysis of judgments of help-giving. *Journal of Personality and Social Psychology. 39,* 186-200.

Weiner, B., & Kukla, A. (1970). An attributional analysis of achievement motivation. *Journal of Personality and Social Psychology. 15,* 1-20.

Wilensky, R. (1978). *Understanding goal-based stories* (Technical Report #140). Computer Science Department, Yale University, New Haven, CT.

APPENDIX A: ABBREVIATED TRACE OF CURRENT DAYDREAMER

Below is an abbreviated trace of DAYDREAMER producing fragments of NUART-DAYDREAM2, NUART-DAYDREAM3, and NUART-DAYDREAM4. The entire trace showing all goals, subgoals, preconditions, plans, relaxations, control goals, and emotions, is over 16 pages long, so only a suggestive segment with much information removed can be shown here.

```
-----------------------------------------
DAYDREAMER [GATE/T/Apollo of 6/8/85] ...
Input? You are near Debra Winger.
Input?
Mode? performance
I tell Debra that I want her and me to
go out on a date.
Input? She tells you that she does not
Input? want her and you to go out on a
Input? date.
Input?
Mode? daydreaming
I fail at going out with Debra.
TERMINATE PLANNING FOR [^WEB.1631: GOAL FAILED ME (IPT-LOVERS)]
-----------------------------------
 IF   self goal failure
 THEN activate negative affect
-----------------------------------
ADD TO WM [^WEB.1632: AFFECT NEG ME]
I feel displeased.
EPISODE INDEX: Plot Unit [^WEB.1633: PU-DENIED-REQUEST DEBRA ME]
-------------------------------------------------
 IF   negative emotion associated with failure
 THEN activate control goal to rationalize failure
-------------------------------------------------
ADD TO WM [^WEB.1635: GOAL ACTIVE (RATIONALIZATION) ME CONTROL]
I want to rationalize failing at going
out with Debra.
START PLANNING FOR [^WEB.1635: GOAL ACTIVE (RATIONALIZATION) ME]
GOAL [^WEB.1635: GOAL ACTIVE (RATIONALIZATION) ME CONTROL]
 INTENDS PLAN [^WEB.1674: PLAN ME]
ADD TO WM [^WEB.1674: PLAN ACTIVE ME]
EXECUTE CODED PLAN [^WEB.1674: PLAN ACTIVE ME]
ADD TO WM [^WEB.1677: GOAL SUCCEEDED (IPT-LOVERS) ME DELTA]
I succeed at going out with Debra.
ADD TO WM [^WEB.1500: IPT-LOVERS DEBRA ME]
-------------------------------------------------
 IF   active indices match episode indices
 THEN retrieve episode and analogize to current situation
-------------------------------------------------
ADD TO WM [^WEB.118: EPISODE JODIE]
ADD TO WM [^WEB.1678: M-ACT DEBRA]
The time Jodie Foster the actress acted
in Paris.  Debra acts in Paris.
PROVE ALL (#AND# [^WEB.780: IPT-LOVERS ME] ...) [^WEB.782: #PROX#]
PROVED (#AND# [^WEB.780: IPT-LOVERS ME] ...) 1 WAY(S)
PROVED [^WEB.782: #PROX#] 1 WAY(S)
-------------------------------------------------
 IF   positive interpersonal thema with person and
      requirement violated
 THEN activate preservation goal
      on the interpersonal thema
-------------------------------------------------
ADD TO WM [^WEB.1855: GOAL ACTIVE (IPT-LOVERS) ME PRESERVATION]
I want to continue to be going out with
Debra.
START PLANNING FOR [^WEB.1855: GOAL ACTIVE (IPT-LOVERS) ME]
GOAL [^WEB.1855: GOAL ACTIVE (IPT-LOVERS) ME PRESERVATION]
 INTENDS PLAN [^WEB.1861: PLAN ME]
ADD TO WM [^WEB.1861: PLAN ACTIVE ME]
PERFORM ACTION [^WEB.1862: PTRANS ME (POLITY)]
ADD TO WM [^WEB.1862: PTRANS ME (POLITY)]
I go to Paris.
ADD TO WM [^WEB.1856: IPT-LOVERS DEBRA ME]
TERMINATE PLANNING FOR [^WEB.1867: GOAL SUCCEEDED ME (IPT-LOVERS)]
PROVE (#AND# [^WEB.828: M-JOB ME] ...)
PROVED (#AND# [^WEB.828: M-JOB ME] ...) 1 WAY(S)
-----------------------------------
 IF   have a job and requirement violated
 THEN activate preservation goal on the job
-----------------------------------
ADD TO WM [^WEB.1868: GOAL ACTIVE (M-JOB) ME PRESERVATION]
I want to continue to have a job.
START PLANNING FOR [^WEB.1868: GOAL ACTIVE (M-JOB) ME PRESERVATION]
GOAL [^WEB.1868: GOAL ACTIVE (M-JOB) ME PRESERVATION]
 INTENDS PLAN [^WEB.1874: PLAN ME]
ADD TO WM [^WEB.1874: PLAN ACTIVE ME]
CHECKING PLAN PRECONDITIONS [^WEB.1874: PLAN ACTIVE ME]
-------------------------------------------------
 IF   working on a goal causes a p-goal to be activated
 THEN resolve conflict
-------------------------------------------------
ADD TO WM [^WEB.1876: GOAL FAILED PRESERVATION ME (M-JOB)]
I fail at having a job.
-----------------------------------------
 IF   failure of goal G and success of G leads
      to major goal failure G1
 THEN failure of goal G is rationalized
-----------------------------------------
EPISODE INDEX: Plot Unit [^WEB.1877: PU-MIXED-BLESSING ME]
ADD TO WM [^WEB.1879: RATIONALIZATION]
I rationalize failing at going out with
Debra by the fact that succeeding at
going out with her leads to failing at
having a job.
-------------------------------------------------
 IF   control goal to rationalize goal failure succeeds
 THEN reduce scale of affect associated with goal failure
-------------------------------------------------
REPLENISH ACTIVATION OF [^WEB.1632: AFFECT LESS-THAN-NORM NEG ME]
I feel a bit displeased.
ACTIVATION FALLS BELOW LIMIT [^DEBRA: PERSON RT-ACTOR FEMALE]
REMOVE FROM WM [^DEBRA: PERSON RT-ACTOR FEMALE]
ACTIVATION FALLS BELOW LIMIT [^WEB.1497: PTRANS ME (ORGANIZATION)
REMOVE FROM WM [^WEB.1497: PTRANS ME (ORGANIZATION)]
-----------------------------------
 IF   person caused a self goal failure
 THEN activate anger toward person
-----------------------------------
ADD TO WM [^WEB.1882: AFFECT DEBRA NEG ME]
I am angry at Debra.
-----------------------------------------
 IF   negative affect directed toward person
 THEN activate control goal to gain revenge
-----------------------------------------
ADD TO WM [^WEB.1883: GOAL ACTIVE (REVENGE) ME CONTROL]
I want to gain revenge for Debra tel-
ling me that she does not want her and
me to go out on a date. I want Debra to
like me. I study to be an actor. I tell
Debra that I do not want her and me to
go out on a date.  I feel pleased.
-----------------------------------------
 IF   recalled goal failure
      and negative emotion not strong
 THEN activate control goal to reverse failure
-----------------------------------------
ADD TO WM [^WEB.2305: GOAL ACTIVE (FAILURE-REVERSAL) ME CONTROL]
I want to reverse failing at going out
with Debra. I want to be going out with
Debra. I want Debra and me to be in
touch. I want to know Debra's telephone
number. I want Debra and me to be in
touch.
START PLANNING FOR [^WEB.2249: GOAL ACTIVE (VPROX) ME]
GOAL FAILURE [^WEB.2249: GOAL ACTIVE (VPROX) ME]. PLANNING LOOP
ADD TO WM [^WEB.2252: GOAL FAILED ME (VPROX)]
TERMINATE PLANNING FOR [^WEB.2260: GOAL FAILED ME (IPT-LOVERS)]
-----------------------------------------
 IF   top-level goal failure from planning loop
 THEN index new conditional plan precondition
-----------------------------------------
INDEXING UNDER GOAL [^WEB.1284: GOAL (IPT-LOVERS) ACTIVE DELTA]
 PLAN [^WEB.1286: PLAN]
 CONDITION [^WEB.2261: VPROX]
 PRECONDITION [^WEB.2262: KNOW (PHONENUMBER)]
INTERRUPT...
Input?  You are near Debra Winger.
Input?
Mode? performance
I tell Debra that I want to know her
telephone number.
Input?
-----------------------------------------
```

Integrating Marker-Passing and Problem Solving

James A. Hendler
Department of Computer Science
Brown University[†]

In this paper we describe how an efficient underlying mechanism, a parallel, spreading activation algorithm, can be used during problem solving. We present this mechanism, chosen due to its demonstrated usefulness for several other cognitive tasks, and show how it can be used to guide the problem solver in choosing correct plans, rejecting plans that violate constraints, or modifying plans as they are generated. Examples of how this technique is used are given, and an implementation of such a system, integrating a marker-passer with a problem solver, is described. The paper discusses some of the desiderata in designing such systems and some of the issues that arise. Some future directions for the work are also described.

1. Introduction

A major problem faced by problem-solving systems today is that of making a choice in the presence of multiple alternatives. It is often the case that the system has access to knowledge that would lead to the optimal solution, or that could avoid a conflict, but this knowledge is not used. Consider the case of a system trying to solve a goal such as *Satisfy hunger*. It must make a choice between alternatives such as *Go to restaurant* and *Eat at home*. In the absence of other knowledge it may not matter which path is chosen, but what if we had already asserted *You have no money* to this system? At this point there is a conflict down the *restaurant* path, but none down the *eat at home* path. Thus the system should choose the latter. Unfortunately, most present day systems are unable to take advantage of this sort of information, and would make the choice at random. At a later point in the problem solving it would encounter the error (assuming it took the *restaurant* path) and be forced to backtrack. Avoiding backtracking has been a primary goal of problem-solving researchers.

In this paper we present a new approach to the issue of choosing paths during problem solving. We propose that a suitably structured "marker-passer", a parallel, spreading activation mechanism, can be used to aid the choice mechanism used by a problem-solver. In this paper we will describe an implementation of such a mechanism, as well as discussing why such a mechanism is desirable.

2. Integrating Marker-Passing and Problem Solving

Present day problem solvers work by using information found in Memory to generate plans. These plans are then subjected to some form of plan Evaluation, either in the form of demons (cf Sussman, 1975; Sacerdoti, 1977) or meta-rules (cf Wilensky, 1983) which critique the plans and if necessary, return them to the problem-solver for reworking.

Most problem solvers work by making step-wise refinements on subplans until primitive actions are reached. Thus *Restaurant* would be broken down into *enter, order, eat, pay,* and *leave. Order* could be broken down into *read menu, decide,* and *tell waitress*, etc. Steps such as *read menu* would be primitive and thus the final plan would be comprised of such steps.

It is this step-wise refinement that causes the problem described in the introduction. At the time we decide which plan to use we do not yet know what steps it will eventually decompose into. However, it is when we try to perform these primitives that the conflicts will most often manifest themselves.

[†] This research was supported by Office of Naval Research under contract N00014-79-C-0592

It is clear that if a mechanism could be developed that would examine all these primitive actions looking for possible conflicts and identifying them prior to the making of a choice then the problem solver would benefit. This mechanism would need to find all possible bindings for the variables being passed through the various levels of substeps checking for conflicts.

Unfortunately, such a mechanism does not presently exist. To be computationally viable this mechanism would need to perform deduction extremely quickly. Further, since all possible bindings of any variables need to be examined, large numbers (susceptible to combinatorial explosion) of low-level deductions[†] would need to be done. Parallelism would improve the situation, but not solve it, due to inherent limitations on the efficiency of such deductions and the combinatorics of multiple possibilities of variable bindings.

It is possible, however, to approximate this mechanism with a system that can examine the primitive actions quickly while looking for special features. A Marker-passing system is one such mechanism. It is our contention that by merging a marker-passer with a problem solver we can provide substantial improvement. Our system (Figure 1) has the marker-passer integrated with the problem-solver and plan evaluator. At this point we will discuss exactly what we mean by marker-passing and why we prefer this type of system. Following that we will show some examples of how the marker-passer is integrated into the problem solver.

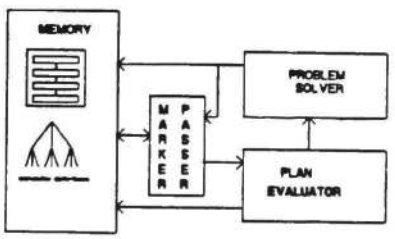


Figure 1: Adding a marker-passer to a Problem Solver.

2.1. What is Marker-Passing?

Marker-passing (cf Quillian, 1966; Fahlman, 1979; Charniak, 1983), a computational model of spreading activation, can be thought of as the marking of nodes adjacent to some node in memory, and then marking all nodes adjacent to those, etc. Thus, for example, in a traditional semantic net system if we marked *CAR* we would then mark *WHEELS*, *VEHICLE*, and all those concepts directly related to *CAR*. Following this we would then mark *RUBBER*, *ROUND* and all those concepts relating to *WHEEL*, as well as *TRANSPORTATION* and those concepts relating to *VEHICLE*. This process would proceed until either nothing was left to mark or the process was called to a halt.

By marking first one node and then another we can see how they are related by examining those concepts marked by both. Thus, in this example from Charniak (1983):

John wanted to commit suicide. He got a rope.

We would start passing markers at *suicide* and then pass markers starting at *rope*. This would find an intersection at the node *noose* which would be found both from *suicide* (the instrument of a *hang*) and from *rope* (as a material relation).

The primary difference between the marker-passer defined by Charniak (1983; based on Quillian, 1966) and the parallel model proposed by Fahlman (1979) is that the former will return the path found, rather than reporting the node of intersection (as would the latter). Thus in the example above the marker-passer would return:

SUICIDE -> to do suicide KILL self -> KILL -> HANG is a type of KILL -> HANG -> instrument of HANG is NOOSE -> NOOSE -> NOOSE is made of ROPE -> ROPE

[†] Depending on the type of problem-solver being used these computations could involve unification, pattern-matching, script instantiation, etc. In this paper all of these types of tasks will be referred to as "low-level deductions" for want of a better term

Notice that the marker-passer does not perform any deductions during its run. If we had asserted that the agent and patient of a *hang* had to be different this path would still be found. This would not be a valid path since the definition of *suicide* requires that the agent and patient must be the same. It is this "blind" behaviour that enables us to implement marker-passing efficiently, at the expense of finding some false paths. The issues of evaluating false paths and avoiding their generation are discussed later in this paper (section 4.2).

2.2. Why Marker-Passing?

In considering an underlying mechanism to use for problem-solving it was our desire to find a system with several properties: it must be able to examine many possibilities efficiently, it must return information that the problem-solver can use for making choices, and it would be desirable to start from an existing system, being used for other cognitive problems. Marker-passing fits all three of these criteria.

Earlier, we discussed how the choice mechanism needs to be able to access the primitive actions of our problem-solver. As the marker-passer proceeds from a plan it marks the substeps, the substeps of these, etc. until it reaches the primitives. Since it does not do deduction as it marks it is able to reach these nodes efficiently. Further, existing implementations of marker-passing (Fahlman, 1979; Charniak, Gavin, Hendler, 1983) are formulated so as to be computed in parallel[†] thus gaining efficiency. In section 3 we will show examples of how the problem-solver can take advantage of the marker-passer. Thus, we satisfy our first two criteria.

Our system is not, by any means, the first to propose the use of marker-passing as an underlying mechanism. Quillian (1966) described a system, TLC, in which spreading activation was used to analyze sentence fragments such as "lawyer's client." As these phrases were analyzed the system would "learn" their meaning and thus be able to build to an understanding of more complex sentences. Collins and Loftus (1975) expanded some of Quillian's ideas, and described the psychological relevance of the spreading activation idea.

Fahlman (1979) described a system in which a marker-passing like scheme was used as the basis for a deductive system. In his system, NETL, links between nodes had properties that determine how marks pass over these links. By allowing this feature, many paths that a blind marker-passer would find are avoided. For example, if we wanted to encode the information that "Clyde was a *pink* elephant" we would have a regular link from *Clyde* to *elephant* and another link, called a cancellation link, from *Clyde* to the area where we record that *the color of an elephant is grey*. Fahlman described an implementation scheme for doing many standard deductive inferences using these sorts of links.

Charniak (1983) described the use of the marker-passer as a device for aiding in the process of story comprehension. The marker-passer is used to find possible paths through the frame knowledge database that relate two frames together. These paths enable the system to build up a model of the frames being instantiated. Thus, in the example of section 2.1, the path from *suicide* to *rope* is found by the marker-passer. It returns this information to a higher level mechanism which uses the information for instantiating the *hang* frame and for building a representation of the events in the story.

Hirst (1983) used Charniak's model of marker-passing for word sense disambiguation. Part of his "Polaroid words" system would pass markers between the senses of words being examined. Thus, in the case of the sentence:

The farmer bought the straw.

the marker-passer would find the path between *farmer* and the *hay* meaning of *straw* thus preferring this meaning over the *drinking straw* meaning for which no significant connection would be found. Further, his system explains the difficulties that would be found in disambiguating a sentence such as:

[†]Due to hardware limitations these systems have been implemented as "psuedo-parallel." That is, although running on serial machines they violate no constraints of locality or temporality that a massively parallel system would need.

The astronomer married the star.

Granger, Eiselt, and Holbrook (1984) developed a model of parsing based on the spreading activation model. Their program, ATLAST, models the integration of lexical, syntactic, and pragmatic processes during natural language comprehension. A spreading activation model of memory is used to propose paths that are then evaluated by a filtering mechanism. Those paths found to be significant are pursued by the parser.

3. Examples

The flow of control when we integrate the marker-passer with the problem-solver was shown in figure 1. In general the following occurs: To start, the problem-solver is invoked. In attempting to solve problems it passes markers throughout the knowledge base. When these markers encounter other markers an intersection is found. The paths meeting at this intersection are then returned to the plan evaluation mechanism. This mechanism checks this path for information which can be used to help guide the problem-solver. If no such information is found, the path is ignored.

The system checks to see if this path proposes a solution to an existing problem (see section 3.1 below), causes a conflict (sections 3.2 and 3.3), or suggests a modification to an existing plan (section 3.4). If one of these is identified the system takes appropriate action choosing, rejecting, or modifying the plan as necessary. In this section we will show examples of these situations and informally describe how the system would work. In later sections we will discuss the specifics of this implementation in more detail.

3.1. Example 1

The first example we will examine is one in which the marker-passer can help us identify a correct solution to a problem with many alternatives. In this example we consider the case of

Trying to commit suicide while holding a gun.

Figure 2 shows a simplified view of the knowledge base used in this example.

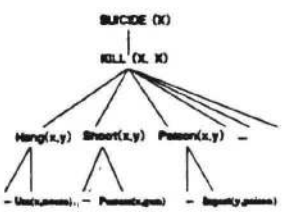


Figure 2.

The system needs to choose between each of the possible ways of commiting suicide shown. It starts by passing markers to each of the alternatives, these in turn mark their subplans, etc. While this goes on we are also passing markers starting with the frame[†] *gun*. Since this marks *gun*, and since *shoot* has as a precondition the possession of a *gun*, we find an intersection. The path

SUICIDE -> to do SUICIDE(x) KILL (x,x) -> KILL -> method of KILL(y,x) is SHOOT(y,x) ->
SHOOT -> precondition of SHOOT(y,x) is possess (y,GUN) -> GUN

is reported to the plan evaluator. Since this path presents a potential solution to the task which has been posed, committing suicide, the plan evaluator can cause the problem solver to prefer the *shoot* plan for achieving the *suicide* goal.

[†]We will use frames here as a generic word for a planning structure. The user may replace it with "script", "schema", "plan" or other such term.

3.2. Example 2

We can also use the marker-passer to reject plans that violate externally generated constraints. Consider, for example, the following situation:

> You are told at some time that the Air Traffic Controllers are on strike. At some later time you are told to plan a trip to California.

Figure 3 shows a simplified view of the knowledge we have in the database.

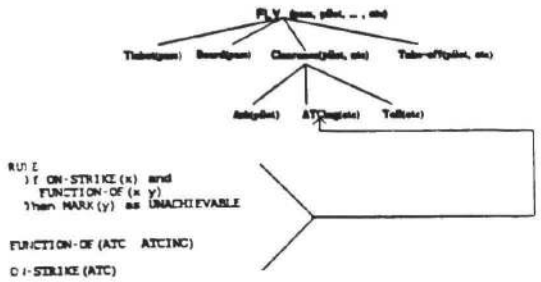


Figure 3.

It is important for our problem-solver to realize that the first statement (the ATC strike) is in conflict with our primary choice for the method of getting to California. When the system is given the first sentence it wishes to add it to the database. The system enters this information. It also notices a forward inference rule of the form *IF someone is on strike AND that person has some function THEN that function might not be able to happen.* Since the Air Traffic Controllers have as their function *ATCing* this would be marked.

We would now ask the system to plan a trip to California. Among the options to choose from will be the *FLY* frame. We will now pass markers on our alternatives. This will mark the *Ticket*, *Board*, *Clearance*, and *Take-off* frames, followed by each of the substeps of each of these frames. Since *ATCing* is one of the substeps of *clearance* it will be marked. At this point the marker-passer would recognise that an intersection has been found, and will return the path

> TRIP -> method trip FLY -> FLY -> step of FLY: CLEARANCE -> CLEARANCE -> step of CLEARANCE: ATCING -> ATCING -> ATCING (marked as unachievable by FACT of strike)

This information is passed to the plan evaluator which examines the path. Since ATCING has been marked as unachievable the system asserts that it cannot be done. The system now invokes a rule of the form *IF I cannot achieve a step in some plan, THEN I cannot achieve that plan.* Since ATCING is a step in the plan for FLY we can rule out the choice of flying as a method of transport for the problem solver.

3.3. Example 3

Another use of this mechanism is to identify conflicts between parts of plans. As an example consider the case of

> Buying a cleaver while on a business trip.

Figure 4 shows a simplified view of the information we use to process this example.

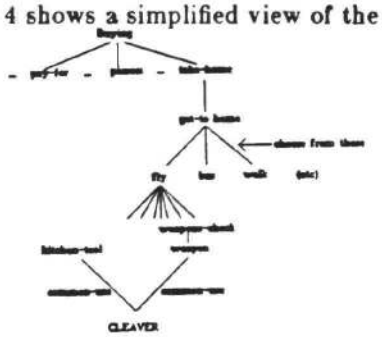


Figure 4.

The system starts by attempting to plan the buying of the cleaver. As such it passes markers down the paths for *Cleaver* and for *Buying*. From *cleaver* it use the knowledge that cleavers are used as *kitchen tools* and as *weapons* so it marks these nodes.

The system also marks the information about *buying* which includes the step of taking home that which is bought. Via the substeps of substeps the marker-passer eventually notices that one way to go home is to *fly* and that within *flying* one must pass through a *weapons check*. However, *weapons check* marks *weapon* and finds that it is already marked. Thus, an intersection has been reached and the information is passed to the plan evaluator.

The plan evaluator examines this path and discovers that the intersection involves carrying a weapon through a weapons check, which leads to being arrested, an undesirable state.

The plan evaluator can now use a heuristic which states that *IF a path causes one to reach an undesirable state THEN rule out the plan during which this state is enocuntered.* Since *Flying* is the plan which dominates the *weapons check* we can rule it out as the method of getting the cleaver home. It can then do replanning and either reject the purchase or amend the plan so as to avoid the illegality (and thus check the cleaver in as luggage).

3.4. Example 4

In the final example the marker-passer is used to find an inference rule that causes a modification of an existing plan. Take for example the following (from Wilensky, 1983):

The planner is given the task of fetching a newspaper from outside during a rainstorm.

Figure 5 shows the necessary information, simplified as usual.

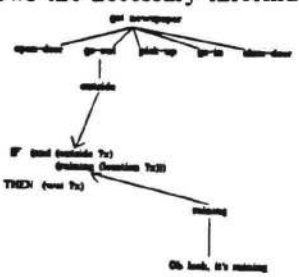


Figure 5

As before we pass markers from both parts of this problem. We thus mark the elements of the *get newspaper* plan and the *raining* statement. In this case we find the intersection not at one of the nodes of the plan, but at an inference rule in the system. This rule, *If it's raining AND if someone is outside THEN an umbrella should be used by that person*, is now examined by the plan evaluator. Since the person in question will can be bound with the agent of *get newspaper*, the plan will be modified so that that agent takes an umbrella.

4. SCRAPS: An Implementation

We have implemented a system, called SCRAPS, that combines a marker-passer and a problem-solver as described so far. In this section we will describe this implementation and discuss how we avoid some of the potential problems inherent in the design of such a system.

4.1. Implementation Details

SCRAPS is implemented using the problem solver NASL (McDermott, 1978; Charniak, 1982) on a logic-based semantics system, FRAIL (Charniak, Gavin, Hendler, 1983). Built into FRAIL is an implementation of a marker-passer. FRAIL and NASL are both implemented in the NISP dialect of Lisp (McDermott, 1983) and run on several different types of machines.

One important aspect of a marker-passing system has to do with the overall organization of the frame knowledge database being searched. For the purposes of performing marker passing, this database should be viewable as a large network of independent processors, each representing

a frame, that are interrelated via links along which markers may be passed. Each frame must be capable of both receiving and passing markers to neighboring frames through these links, which correspond to directed arcs relating frames to each other.

The FRAIL language provides such a system. Frames are translated into logical predicates which are stored in a LISP database as properties on atoms. These atoms serve as frames and the logical predicates serve as links. Thus, FRAIL provides both a logical language for deduction and a semantic-net representation useful for marker-passing.

The NASL problem solver uses FRAIL to retrieve the possible plans usable for satisfying a goal. When more than one plan can be applicable FRAIL uses a simple choice mechanism that checks for plans which are ruled out or ruled in. SCRAPS uses this mechanism as the primary communication between the plan evaluator and the problem-solver. When the marker-passer finds a path that the plan evaluator recognizes as a potential solution, that path is ruled in and all other paths are ruled out. Similarly, when the plan evaluator recognizes a problem it simply rules out that plan which would cause the contradiction. Synchronization is provided by having the problem-solver's choice mechanism wait until a message is received from the plan evaluator allowing it to proceed. Modifications to plans are done by having the plan evaluator make assertions directly into the FRAIL database.

As an example, consider once again the restaurant example of section 1. The system is started with the assertion that no money is available, and thus the *paying* frame is marked as unachievable via a forward chaining rule. We then ask the problem solver to solve the task of *eating* with some specific agent, *agent1*. NASL calls upon FRAIL to retrieve the potential plans and finds both *restaurant* and *eat at home*. As FRAIL retrieves these plans markers are passed. The path from *restaurant* to *money* is found and reported to the Plan Evaluator. The Plan Evaluator recognizes the conflict in the restaurant plan and therefore generates a *rule out* statement. [†] It then signals NASL to continue. NASL now uses its choice mechanism to examine the possibilities. Since the restaurant frame has been ruled out it uses the only other possibility, eat at home, and continues on.

4.2. Implementation Issues

For our system to be efficient it must be the case that the added efficiency of avoiding backtracking outweighs the overhead time spent in evaluating the returns from the marker-passer. If the plan evaluator takes longer to reject false paths than it would take the problem solver to find the correct path then the addition of a marker-passer loses its value. The plan evaluator must be designed so that it can quickly reject those paths with nothing to offer.

In the SCRAPS system the plan evaluator is composed of a set of *ad hoc* heuristics that can examine a path and see if any useful information can be gleaned. These rules serve two purposes: first, we wish to quickly reject paths which are not going to yield useful information, and second, we wish to be able to extract information from those paths which are useful. This is done by passing the path through a series of rules in a cascaded manner. The early rules are designed to reject paths known not to be useful, the later rules are designed to quickly check for a certain feature and pass the rule on to the next heuristic if it is not found.

The early rules work by first checking the node of intersection of a path. If the node of intersection is found to be a "promiscuous" node, one with a very high out-branching factor, the path is probably not useful and can therefore be rejected. If the node of intersection is not of this form, the path is checked to see if certain features can be detected that will cause the rule to be rejected. For example, a rule checks to see if a path is an *ISA Plateau*. This would be the situation if we passed markers from, for example, *dog* and *cat* and found an intersection at *animal*. These sorts of paths are useful for the word sense disambiguation use of marker-passing, but not for problem solving. Eugene Charniak (fortcoming) has been examining the paths returned by the marker-passer and trying to formalize path checking as a resolution proof procedure. As this

[†] Readers wanting more details about the format of the rules used by the choice mechanism in FRAIL and NASL are directed to Charniak, Gavin, and Hendler (1983).

work becomes practical our system will be changed to use this method for the rejecting of false paths.

Once the obviously useless paths have been rejected the Plan Evaluator then uses the second type of heuristic to see if the path yields valuable data. This is done by having each heuristic designed to quickly examine the path for some feature. If that feature is not found the path evaluator immediately passes this path on to the next heuristic. If none of these heuristics find such a feature then the path is considered uninteresting and rejected. If, however, such a feature is found the path is subjected to closer inspection to see if useful information is yielded. Due to the underlying logical formalism of FRAIL unification is used as a first test in these situations. If items in the path do not unify it can be rejected.

As an example of these heuristics consider what the Plan Evaluator would do with the *restaurant* example. With variables included, the path found is of the form:

> GOAL EAT (*me*) -> EAT -> To-do EAT(x) use RESTAURANT(x) -> RESTAURANT -> step-of RESTAURANT(x) is PAYING (x y) -> PAYING -> PAYING(*me*) is Marked as unachievable

We are also informed that the place where the marks intersect is on the node *Paying*. First we examine the node *Paying* to see if it is inherently uninteresting or "promiscuous." Since it isn't we go on to check the form of the path to see if we have found an ISA plateau (a very efficient heuristic). Again, this test fails, so we do not reject this path. We now move on to the second type of heuristic, those designed to extract information. The first heuristic to fire might be the heuristic checking for meta-rules of the type found in example 3.4. This rule would check the path to see if any of the links was of the form *Meta-rule*. Since none of the links in the path above have this form it is passed on to the next heuristic. This rule checks to see if either of the end-points of the path is marked as unachievable. Since this is the case, the rule is now subjected to closer scrutiny. The unification algorithm is used to see if variables in the path match up. Since, in this case, the person with the *Eat* goal is the same as the person trying to do the *Paying* the unifier reports the match. We now can use this rule to rule out using the restaurant plan as described in section 4.1.

Although, the key factor in designing these heuristics is to make the tests for path rejection as efficient as possible, it is still essential that the number of false paths generated can be held down. If too many paths are found by the marker-passer we lose efficiency since the evaluator must examine all those paths which are reported. If too few paths are reported the correct information can be missed. The marker-passer must be designed so as to constrain the number of paths reported.

We are presently exploring some of the issues involved in the design of such constraints. The marker-passer in FRAIL uses a propagation limitation on markers to help keep down the number of false paths found. Marker-passing is invoked with a certain strength at a starting node. That node divides the strength by the number of nodes it has as neighbors, and passes a marker to each of them with this new strength. Each of these nodes proceeds in turn marking its neighbors with ever decreasing strength. Once strength falls below a certain limit marker-passing stops along that path. Thus, long thin trails are preferred to short multiply branching ones.

Another means of limiting the number of false paths is the removal of marks during processing. At present we have developed a model which degrades marker strength over time, depending on the original strength of the marking. When the strength is sufficiently decremented the mark is removed. Recent work (Granger, Holbrook, 1983), however, has suggested that the removal of marks is a more complex process than the one we have implemented. We are presently examining this issue.

5. Research Directions

Recent work (Wilensky, 1983) has been examining the issues involved when a single agent has multiple goals, or when multiple agents, each with their own goals, are involved in a situation. This work has concentrated on recognizing, and resolving, conflicts between these goals. In section 3.3 we showed an example of how a system like SCRAPS could be used to detect problems

caused by a conflict between subgoals during a problem-solving task. Our recent research has been directed at exploring the relationship between these sorts of subgoal conflicts and the multiple agent goal conflicts being examined by Wilensky. It is our belief that SCRAPS can be extended to allow the plan evaluator to work on such tasks.

Take the following example (based on an example from Wilensky, 1978):

> Bill wants to get the treasure, but the hoard was guarded by a dragon.

SCRAPS would be invoked to solve two tasks one in which Bill attempts to obtain a treasure, and one in which a dragon attempts to guard a hoard. The marker-passer finds the path from Bill to the dragon via the path relating the treasure to the hoard. The plan evaluator can now examine this path. Since, in this simple case, the variables can be bound consistently the plan evaluator can recognize a potential planning conflict. If it had been the case that Bill had decided that the treasure he wanted was not the hoard but the giant pearl being guarded by the huge octopus, the plan evaluator would fail to match the dragon's treasure and the goal of Bill's quest, and thus the path would be regarded as false and therefore rejected.

The primary problem with extending SCRAPS to handle multiple agent planning is the growth in the number of false paths reported when common objects are used in many of the plans. Consider the following situation:

> John wishes to buy a farm. Mary, no relation to John, wants to get a job so she can earn more money. Bill, no relation to either of the others, wishes to go to a restaurant.

Our marker-passer will return paths connecting all of these as potential planning conflicts dealing with *money*. Similar growth in paths are caused by items like *cars*, *clothes*, and any other item common to many plans.

The single-agent version of SCRAPS is able to use a heuristic which rejects paths which meet at common objects. Thus, in the cleaver example of section 3.3 a false path such as:

> Cleaver -> ... -> weapons-check -> AGENT of Weapons-check is PERSON ->
> -> PERSON -> AGENT of buying is PERSON -> BUYING

can be rejected since the node of intersection, *person*, is involved in so many frames. In the multi-agent cases, however, these false paths cannot be rejected as simply. It now requires using our knowledge base to recognize that, for example, Bill's money, Mary's money, and John's money are all different.[†] We are presently working on redesigning the plan evaluator to handle such cases.

6. Conclusions

Integrating marker-passing with problem solving enables us to avoid backtracking in many problem solving tasks. An efficient underlying mechanism, the marker-passer, is used to choose correct plans, reject plans violating constraints, or to modify existing plans. The marker-passing mechanism is chosen due to its demonstrated usefulness during cognitive tasks.

We have shown that careful attention must be paid to the design of the marker-passer and the plan evaluator for the problem solving process to gain efficiency. At present SCRAPS is able to provide this efficiency boost in several situations. We are presently examining other problem solving areas to see if other classes of problem solving behaviours can benefit from this technique.

7. Acknowledgements

The author wishes to acknowledge Eugene Charniak for much aid in the design and critiquing of the work described in this report. Several of the examples used in this paper are based on his work. We also wish to acknowledge Doug Wong's aid in helping to formalize some of the pitfalls inherent in the design of a marker-passer.

[†] To convince oneself that this is true, consider the case where John and Mary are married and share a common bank account.

8. References

Charniak, E. *Micro-Nasl Reference Manual* Dept. of Computer Science, Brown University, 1982.

Charniak, E. "Passing markers: A theory of contextual influence in language comprehension." *Cognitive Science* 7(*3*), July - Sept. 1983.

Charniak, E. *A Neat Theory of Marker-Passing* Dept. of Computer Science, Brown University, forthcoming.

Charniak, E., Gavin, M.K. and Hendler, J.A. *The FRAIL/NASL Reference Manual.* Brown University Dept. of Computer Science Technical Report No. CS-83-06, Feb. 1983.

Collins, A.M. and Loftus, E.F. "A Spreading-activation theory of semantic processing" *Psychological Review* 82(*6*) pp. 407-428, 1975.

Fahlman, S.E. *NETL: A system for representing and using real-world knowledge* MIT Press, 1979.

Granger, R.H., Eiselt, K.P., and Holbrook, J.K. *Parsing with Parallelism: A Spreading Activation Model of Inference Processing During Text Understanding* University of California at Irvine Artificial Intelligence Project Technical Report #228, Sept. 1984.

Granger, R.H. and Holbrook, J.K. *Perseverers, Recencies, and Deferrers: New experimental Evidence for Multiple Inference Strategies in Understanding.* University of California at Irvine Artificial Intelligence Project Technical Report #195, May 1983.

Hirst. G.J. *Semantic Interpretation Against Ambiguity* Brown University Computer Science Technical Report CS-83-25, Dec. 1983.

McDermott, D.V. "Planning and acting" *Cognitive Science* **2** pp. 71-109, April 1978.

McDermott, D.V. *The NISP Manual* Yale University Computer Science Tech Report U/DCS/RR No. 274, June 1983.

Quillian, M.R. *Semantic Memory* (Scientific Report No. 2) Bolt, Beranek, and Newman, 1966.

Sacerdoti, E.C. *A Structure for Plans and Behavior* Elsevier, 1977.

Sussman, G.J. *A Computer Model of Skill Acquisition* Elsevier, 1975.

Wilensky, R. *Understanding Goal-Based Stories* Yale University Computer Science Research Report No. 140, Sept. 1978

Wilensky, R. *Planning and Understanding* Addison-Wesley, 1983.

The Evolution of Knowledge Representations with Increasing Expertise in Using Systems

Dana S. Kay

Yale University

John B. Black

Teachers College, Columbia University

The kinds of knowledge that people have about a computer system and the form in which that knowledge is represented and stored in memory changes as people progress from being beginners to being experts at using the system. By combining a series of studies that we have conducted over the last few years, we have concluded that the knowledge representations that people have about a system progress through four phases as they become increasingly expert. In this paper, we describe these four phases of knowledge evolution during learning and justify each phase using the results of our studies.

The novice/expert paradigm has recently been given a great deal of attention. Studies have examined the differences between novices and experts in domains such as algebra (Lewis, 1981), physics (Larkin, 1981) and computer programming (e.g., Ehrlich and Soloway, 1984; McKeithen, Reitman, Rueter and Hirtle, 1981; Adelson, 1981). Although this paradigm provides insight into the conceptual and performance differences of novices and experts, it does not address the learning process that underlies the transition from novice to expert. To understand what makes an expert expert, it is necessary to examine the learning process. One method for accomplishing this goal is to investigate the organization of the domain-relevant information at various stages in learning. The assumption behind this method is that as learning progresses, the content and organization of the knowledge changes to accommodate the acquired knowledge. Because of the recent interest in computers and how to make them easier for people to learn, we chose to study learning in the computer domain by examining the development of knowledge representations appropriate for this domain.

Others have studied the differences in the knowledge representations of expert and novice programmers (e.g., Ehrlich and Soloway, 1984; McKeithen, Reitman, Rueter and Hirtle, 1981; Adelson, 1981) and the knowledge and performance of expert users of command languages (e.g., Card, Moran and Newell, 1983), but there have been no previous studies of how people's knowledge representations change as they learn to use command language systems. Our research has focused on how people learn to use text editors and, in particular, how their knowledge representations change as they become increasingly expert at using text editors. We are interested in understanding the order in which the components of the knowledge representations are acquired and how this acquisition process is reflected in the conceptualization of the text-editing knowledge. The model that we will present is an incremental model in which the user builds a complex knowledge representation by acquiring different types of knowledge at different phases in learning.

In our discussion here we will draw upon three studies to make our points: the Sebrechts, Black, Galambos, Wagner, Deck and Wikler (1984) study of people learning to use the IBM Displaywriter and the UCSD p-system text editor; the Kay and Black (1984) study of the differences in people at two different expertise levels with a local Yale text editor, and the Robertson and Black (1983) study of the timing changes between keystrokes as people learned to use a simple experimental text editor. Taken together these studies use a variety of research methods to provide converging evidence for our learning phases. Specifically, the Sebrechts, et al. study examined the changes in people's perceptions of command similarity that result from training on two commercial text editors, Kay and Black examined the differences in perceptions of similarity in different people who naturally acquired different levels of expertise with a text editor, while Robertson and Black examined inter-keystroke time evidence for how

people conceptualize a simple artificial text editor as they become practiced in using it.

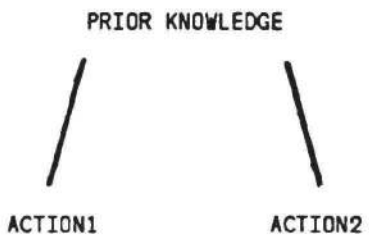
The Four Phases of Learning

The results of these studies have led us to conclude that the knowledge representations of users evolve through four phases as they become expert with a system. In particular, in the beginning user knowledge is determined by preconceptions based on the terminology used for the system, but with a little experience this changes to knowledge of what system commands are relevant to accomplishing various tasks with the system. However, with even more experience the form of the knowledge changes again to form complete plans (i.e., sequences of commands) for accomplishing goals with the system and with increasing expertise the plans become more complex and the user learns when each of these plans is most appropriately selected to accomplish the goal. In the following, we explain each of these phases in detail and describe the evidence supporting each.

Phase One: Preconceptions

Phase One represents the completely naive user who has had no experience using a text-editor. At this stage of learning, users have preconceptions about the terminology that will later refer to editing commands. As a result of prior experience with the terminology to be used in text-editing, users come to the text-editing domain with a knowledge representation that may or may not correspond to the knowledge representation that will develop as text-editing experience increases. Figure 1 presents an example of the type of knowledge structures that exist before any learning has taken place. In this structure, there are two actions (e.g. CENTER and BALANCE) that are related by prior knowledge (e.g. these are actions that make something even).

FIGURE 1: Preconception Knowledge Structures



Evidence of this phase was found for the Displaywriter and the UCSD systems in the Sebrechts et al. study. In this study, naive subjects were given a six hour training session in which they learned to use either the IBM Displaywriter or the UCSD p-system. Before and after training, the subjects were asked to rate the similarity of non-identical pairs of the commands used in each system. To analyze these ratings, Sebrechts et al., used three multivariate analysis techniques. From the results of a hierarchical clustering of the ratings before the training session, they found that initially the commands for both systems clustered based upon prior knowledge definitions of the commands. For example, in the results of the clustering for the Displaywriter commands, CODE and MESSAGE and CANCEL and DELETE were clustered together. CODE and MESSAGE are related in that one can use a code to convey a message. CANCEL and DELETE are related in that they both suggest the elimination of something. In both cases, the similarity of the command names reflects prior knowledge associations between the actions. Similar types of clusters were found in the UCSD system such as FIND and GET and HEADER and MARGIN.

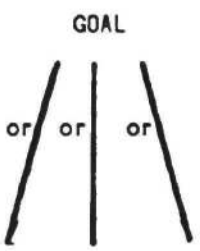
Although there may be some correspondence between the prior knowledge representations of the commands and the text-editing representations, in most situations, the two representations are quite different. Therefore, users in Phase One of the learning process are confronted with the task of overcoming a bias toward interpreting the commands in terms of their prior knowledge associations. To

accomplish this task and achieve any level of expertise, the previously existing knowledge representations must be reorganized to accommodate the acquisition of text-editing knowledge.

Phase Two: Initial Learning

Initial text-editing knowledge can be acquired from a manual, a class, self-teaching or any combination of these. No matter what the learning method, the main goal that the user has is to overcome the prior knowledge bias that exists for the text-editing commands. We believe that the accomplishment of this goal entails (1) learning the goals relevant to text-editing and (2) learning the commands that can be used to accomplish these goals. The first part of the learning process takes place as soon as the user begins to edit and is exposed to various editing tasks. Knowledge of general goals allows for the generation of high-level goal structures that can be used to organize the editing commands. For example, if a user types the word "thee" instead of "the", then the goal or task that is instantiated is to erase the extra "e". However, before this goal can be accomplished the user must learn the actual editing command(s) that are used. That is, the user must learn the functions of the commands and selectively choose the most appropriate command(s). To continue the previous example, a user might learn that the command BACKSPACE serves to get rid of unwanted text and decide to use BACKSPACE to remove the unwanted letter. At this time, the user learns that one of the possible goals to be found in text-editing is to remove unwanted text and one command that accomplishes this goal is BACKSPACE. As a result of this learning, the user develops knowledge structures that link specific goals and commands. Figure 2a presents the knowledge structures that exist at this phase of learning. Here the goal is linked to the actions by OR links because one can use ACTION1 or ACTION2 or ACTION3 to accomplish the goal. For our previous example, the GOAL might be GET-RID-OF-TEXT and ACTION1 might refer to BACKSPACE. As more commands that get rid of text, such as DELETE and SPACE, are learned, they are added to the representation with OR links because the user knows that if you want to GET-RID-OF-TEXT, you can use BACKSPACE or DELETE or SPACE.

FIGURE 2a: Initial Learning Knowledge Structure

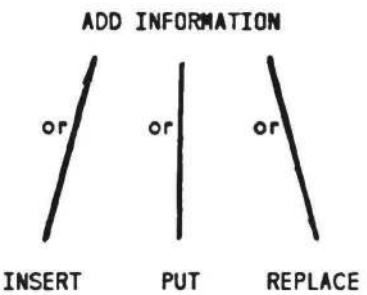


Kay and Black (1984) used a methodology similar to Sebrechts et al., but with a cross-sectional design. In this study, they examined the differences in the knowledge structures of novices and experts who naturally acquired text editing knowledge. They asked both types of users to rate the similarity of pairs of commands used in a local Yale editor and analyzed these ratings using multivariate techniques. In the results of the hierarchical clustering for the novice users, commands were clustered based upon similarity of function in the editor. For example, INSERT, PUT and REPLACE clustered together because all three commands are used to accomplish the goal of adding information to a text. This structure is represented in Figure 2b. Each of these actions could be use to accomplish the goal of adding information. Therefore, all the commands are linked to the goal by OR links. That is, if one wants to add information, INSERT or PUT or REPLACE can be used. These results will later be contrasted with the expert results in our discussion of Phase Three of the learning process.

It is important to note that although these users have acquired some text-editing knowledge, the

organization of this knowledge is based upon only the "result" of the commands and not the procedure that leads to this result. Because of this narrow focus, the users do not possess the knowledge necessary to develop more complex structures such as plans. In particular, while the users know that PUT can be used to ADD INFORMATION, they do not readily know that PUT is merely the main action in a multi-action plan to accomplish this goal.

FIGURE 2b: Knowledge Structure for Adding Text



In addition to providing evidence for Phase One, the results of the hierarchical clustering in Sebrechts et al. also illustrate the transition from Phase One to Phase Two. After six hours training, users no longer clustered commands by prior knowledge associations. The clusters changed to be based more on the function of the system. For example, DELETE which was initially clustered with CANCEL, was clustered with BACKSPACE after training. In the Displaywriter system, either DELETE or BACKSPACE can be used to erase a piece of text.

Because novice users have not refined their definitions of commands, several commands are often linked to a single goal by OR links. At this level of understanding, the user will employ any one of the actions linked to the goal to accomplish the goal. For example, in the knowledge structure presented in Figure 2b, PUT, INSERT and REPLACE are conceptualized as similar because they all accomplish the goal of adding information to the text. With experience, the user will develop more complex representations of these commands that include knowledge of the processes (or plans) that lead to the results of the commands and learn that although these commands all add information to the text, the added information comes from different sources. Evidence for this change in knowledge organization will be presented in Phase Three. For example, PUT uses information from the buffer and needs PICK or DELETE to put the information into the buffer. However, INSERT needs only the information that is typed by the user.

In addition to using hierarchical clustering to examine the knowledge representations of users, Kay and Black and Sebrechts et al. also used multidimensional scaling to study the overall organization of the commands. The results of this analysis were the same in both studies. The commands were organized along (a) a dimension that differentiated system and editor commands (b) a dimension that distinguished formatting from non-formatting commands and (c) a dimension that differentiated commands to begin a sequence from commands to end a sequence. These dimensions suggest the development of a goal space for the commands. This goal space helps the user in accessing the correct goals during an editing session. For example, if one is formatting a document, it is more appropriate to have the goal of centering the text (and accessing the CENTER command) than it is to have the goal of changing a misspelled word. The third dimension (begin/end of sequence) is particularly interesting because it suggests that although users do not organize the commands by specific sequences (plans), they do understand that commands are used in sequences.

Card, Moran and Newell (1983) proposed the GOMS model to account for the text-editing behavior of experts performing routine tasks. In this model, the expert knowledge representation consisted of four components Goals, Operators, Methods and Selection rules. Using our account of initial learning,

we propose that it is the Goals and Operators components of the GOMS model that are acquired first. That is, novices are able to understand the general goals involved in text-editing and the individual commands that are related to these goals. It is reasonable that the goal/action link would be the first link to be formed. During the initial learning of a system, the user is introduced to numerous command names and definitions with little reference made as to when each command should be used. If, instead of forming links between the commands and the goals that they accomplish, the first links to be formed were between commands, then the user would never know when to use each command and therefore, would have to resort to a trial-and-error method of achieving a goal. Thus, the linking of the command to a goal provides the user with some aid in using the correct command in the correct situation.

Novices seem to conceptualize the commands merely by what goals they are relevant for accomplishing. Because of this level of specificity, they have not yet acquired the procedures or plans that are associated with text-editing and thus, each text-editing task becomes a problem-solving task in which they must actively search through their representations of the commands and find the set of commands necessary to accomplish the task.

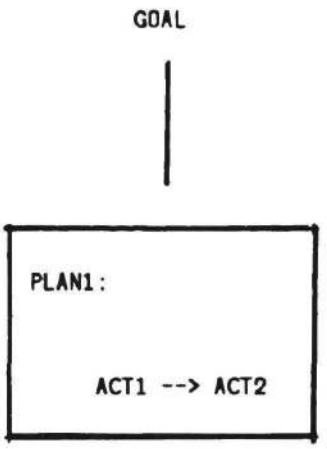
Phase Three: Plan Development

Once the users have acquired the basic editing commands and goals, they learn that there are combinations of commands that are often used together to accomplish a goal. In Phase Three, users develop the ability to form plans by combining the actions that were organized separately in Phase Two. These plans correspond to the Methods of the GOMS model. There are various ways that the transition from Phase Two to Phase Three takes place. For example, with the system that Kay and Black studied, users realize the inefficiency in repeating a command numerous times. To overcome this inefficiency they learn to use the ARGUMENT command that automatically repeats another command for the number of times specified as the argument. Once they have this knowledge, users begin to notice that there are other commands that can be used together in a sequence. This realization leads to a reorganization of the knowledge representation to accommodate the command sequences or plans that are used to accomplish goals.

This reorganization process entails the modification of the goal/action links that were formed in Phase Two. In this third phase, two types of links are formed. At one level, links are formed between the commands or actions that are used in a plan. At a higher level, links are formed between the goals and the plans that are used to achieve the goal. It is also possible to represent this knowledge as productions and explain the transition from Phase Two to Phase Three using the composition process proposed by Anderson (1983). However, since we are primarily interested in specifying the knowledge that is acquired at each phase of learning, we chose to use a network representation in which we trace the development of the links in the network because we think the networks are more perspicuous than productions.

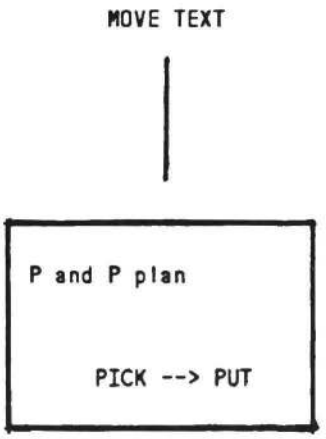
Figure 3a presents the type of knowledge structures that exist at Phase Three of the learning process. In these structures, just as in the Phase Two structures, the goals will guide the use of the commands. However, this guidance is provided by the instantiation of a plan that consists of actions, rather than invoking each action individually.

FIGURE 3a: Beginning Expertise Knowledge Structure



Again, both Kay and Black and Sebrechts et al. provide evidence for the development of Phase Three knowledge structures using the results of a hierarchical clustering analysis. As previously mentioned, the novice users organized commands as individual actions related to goals. On the other hand, expert users appeared to use a more sequence-oriented organization. That is, the perceived similarity between the command pairs was based upon the use of the commands in a plan to accomplish a given goal. For example, in the Kay and Black study, expert users clustered PICK and PUT together because these two commands are used together when a user want to accomplish the goal of moving a piece of text. This example, when contrasted with the Phase Two knowledge representation example in which novice users clustered PUT with INSERT and REPLACE because each of individual commands are used to accomplish the goal of adding information, illustrates the change in the representation of the commands from goal/action links to goal/plan links. Figure 3b presents a graphic representation for the knowledge structure that underlies this result.

FIGURE 3b: Move Text Knowledge Structure



In Phase Two we found that in addition to having specific goal/action knowledge structures, users also organized the commands in a goal space. Given the knowledge structure changes that occur from Phase Two to Phase Three, it is interesting to look at whether or not these changes influence the goal space of the users. The multidimensional scaling results for the experienced users in Kay and Black suggest that there is only one primary change in the goal space. Recall that in the initial learning phase, the dimensions of the goal space are editor/system, formatting/non-formatting and begin/end sequence. Of these three dimensions the former two were also present in the multidimensional scaling

for the experienced users. However, the begin/end sequence dimension is no longer used. This result suggests that experienced users continue to organize the commands in a goal space, but now they realize that interactions with the system don't occur in as rigid an order as they originally thought so the begin/end dimension disappears. They still have sequence knowledge, but it is at the more local plan level, instead of the global begin/end sequence level.

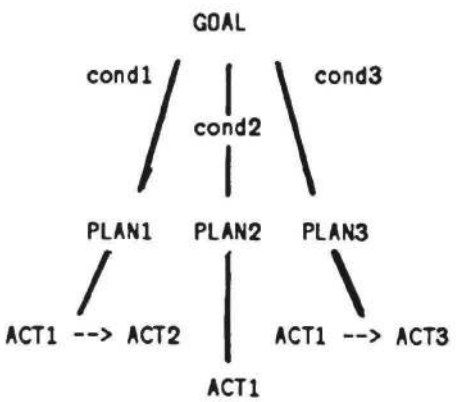
Because users begin to develop plan representations for their editing knowledge, we believe that it is during this phase in learning that users begin to mold the system to suit their own editing style by implementing these plans as macros. This molding makes their performance more efficient because using plan-macros, they can accomplish a goal with one command rather than several. We are currently testing this hypothesis by examining the number and content of macros created by users at various levels of expertise.

Phase Four: Increasing Expertise

Although the formation of simple plans results in some expertise in text editing it is not until compound plans are formed that one can accomplish more advanced tasks. Phase Four of our model accounts for this ability and represents the completion of the acquisition that results in knowledge representations similar to those proposed in the GOMS model of expert performance. In this phase, users (a) combine simple plans into more compound plans to accomplish major goals and (b) develop rules for selecting the best plan to achieve a given goal in a given situation. Once again, we observe a reorganization of the knowledge that results in the development of new links between the components of the representation. In this phase, we see a change from a one-to-one correspondence between goal and plan to a one-to-several correspondence: that is, in Phase Three each plan or sequence of actions is directly linked to a specific goal (e.g. move text) while in Phase Four, there are multiple plans that can be linked to each goal. However, because there are multiple plans, the links that connect these plans must have selection rules that tell the user under what conditions to access the plan to accomplish the goal.

Figure 4a shows the type of knowledge structures possessed by users in Phase Four. In this type of structure, there are several plans (sequences of actions) that may be instantiated to achieve a given goal. To be sure that the correct plan is chosen from the set of applicable plans, the links that connect these plans to the goal are conditions or selection rules that must be met before a given plan is chosen. These conditions can be based upon any distinguishing feature of the plans. Thus, at the highest level of expertise, goals are linked to plans using the conditions under which these plans are invoked, whereas, goals were linked to simple plans in Phase Three and actions in Phase Two.

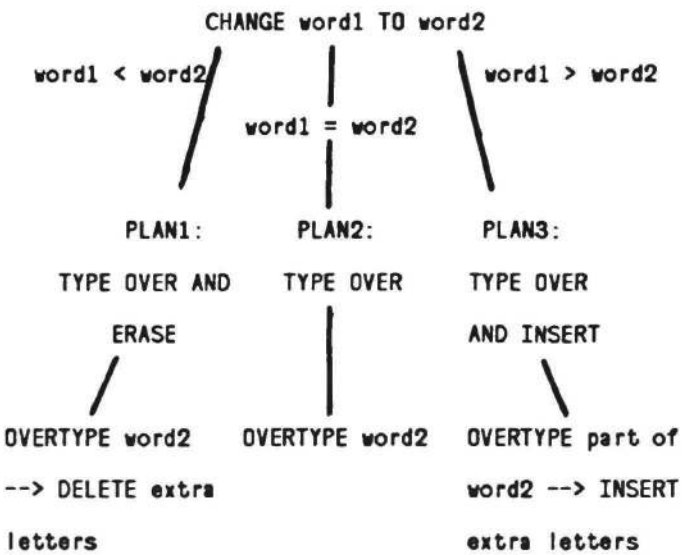
FIGURE 4a: Expert Knowledge Structures



Robertson and Black present evidence for the evolution of compound plans. With increased experience, the time spent pausing between simple plans decreased suggesting that users had combined

the simple plans that they had learned to form compound plans. In addition, the decision time to initiate a compound plan during which subjects choose what they think is the most appropriate plan decreased as the editing session progressed suggesting the acquisition of selection rules that facilitate accessing the most appropriate plan. Figure 4b presents a concrete example of a Phase Four knowledge structure. In the editor Robertson and Black used, to change one word to another word, there are three possible plans (sequences of actions) that may be invoked. The links connecting these three plans to the goal are conditions that must be met before the plan is selected. In the structure depicted, the conditions are based upon the relationship in length between the old word and the new word.

FIGURE 4b: Knowledge Structures for Changing a Word



In addition to providing evidence for the compound plans of Phase Four, Robertson and Black also found evidence for the development of selection rules. In the beginning of the training trials, when users had to change one word to another word, the majority of the users (approximately two thirds) would DELETE the old word then INSERT the new word. However, as experience increased, these same users stopped using the DELETE-INSERT plan and began to OVERTYPE the old word with the new word. In this example, users appeared to develop a selection rule that can be stated as "If you want to change one word to another, use the OVERTYPE plan." This type of selection rule organizes the plans according to the priority that they have in efficiently accomplishing the goal.

The Learning Process

Our four learning phases describe the evolution in user behavior from problem-solving to plan instantiation. That is, initially users must interpret each action in the editing task and monitor the success or failure of the action in terms of the final goal state. However, once the user is familiar with text-editing tasks, the commands are combined into plans that are applied when appropriate. It is the latter process that accounts for the plan-boundry pauses found in the Robertson and Black study, because deciding which plan to use takes longer than the transition from one plan action to another plan action when applying the plan.

Throughout this discussion of our four phase model, we have made references to the GOMS model and the order in which the components of this model are acquired. Although we are not able to distinguish whether the Goals are acquired before, after, or simultaneously with the Operators, we are able to conclude that the Goals and Operators are acquired before the plans that are acquired before the Selection rules. In addition, we noted that as the user becomes more experienced, the components build

upon one another so that the chunks of knowledge represented using these components increase in size. Thus, we were able to trace the evolution of text-editing knowledge structures and extract an acquisition process that in the end results in knowledge representations similar to those found in the GOMS model.

Why do these learning phases occur: that is, what is the learning process that causes the user of a system to progress through these phases of learning? The initial, preconception phase is clearly necessary, because new users have only their prior knowledge to help them understand and use a system in the beginning. The final, increasing-expertise phase is also clearly necessary, because compound plans for accomplishing major goals and the selection rules for choosing plans at the most appropriate times are both necessary for skilled performance (Card, Moran and Newell, 1983). But, why are the two intermediate phases necessary?

Users cannot progress directly from phase one to phase four representations, because transforming phase one representations to phase four representations requires bringing more information together at one time than human working memory is capable of holding simultaneously. Thus, the two intermediate knowledge representations allow the user to progress from phase one to phase four in bite-size chunks that correspond to the limits of human working memory. For example, to progress from phase one to phase two, the users need only learn the links between the commands and the goals that can be accomplished, they do not need to simultaneously learn the sequencing links between the commands that are needed to combine the commands into plans. Then once the phase two command-goal links are mastered, the users can progress to phase three by learning the sequencing links that combine the relevant commands into simple plans for accomplishing elementary goals. Finally, once the simple plans are well learned, the user can progress to phase four by combining these simple-plan chunks into more compound plans and learning the conditions that determine when the various plans are most appropriately selected for accomplishing the goals.

However, while an intermediate step between phases one and four is necessary, phases two and three may not both be necessary. The simple plans of phase three are clearly a necessary precondition for the compound plans and selection rules of phase four, but phase two might be an artifact of the way that systems are currently taught to new users. In particular, current instruction manuals emphasize descriptions of individual commands at the expense of describing how these commands are combined into plans. We are currently investigating whether plan-based instruction manuals will allow new users to skip phase two and progress directly to phase three, thus significantly speeding up the process of learning new systems.

We are also pursuing the model's generality and testing its predictions. In particular, we are testing the the generalization of our model to other domains. The most immediate extension from the text-editing domain is to computer programming. We have begun some preliminary work applying our model to this domain and plan to carry out a full scale longitudinal study in the near future. However, in the more distant future, we plan to try extending the model to non-computer domains (e.g. learning physics, algebra and the operation of devices).

To test the model's predictions, we are using it to guide training in text-editing. Since our model was extrapolated from our observations of the natural evolution of knowledge representations for text-editing information, we believe that using this model to train users might facilitate the learning process. We are currently designing a study to test this hypothesis by comparing the performance of users whose learning was guided by the phases in our model to the performance of users whose learning was guided by commercial training materials in which the emphasis is on the kinds of commands that can be used in the system.

Acknowledgement: This research is supported by grants from IBM, but represents the views of the authors and not necessarily the views of IBM.

REFERENCES

Adelson, B. (1981) Problem solving and the development of abstract categories in programming languages. *Memory and Cognition*, **9**, 422-433.

Anderson, J.R. (1983). *The architecture of cognition*: Cambridge, MA: Harvard University Press.

Card, S.K., Moran, T.P., and Newell, A. (1983). *The psychology of human-computer interaction*. Hillsdale, New Jersey: Lawrence Erlbaum Assoc.

Ehrlich, K. and Soloway, E. (1984). An empirical investigation of the tacit plan knowledge in programming. in Thomas, J. and Schneider, M. (Eds.) *Human Factors in Computer Systems*, Norwood, New Jersey: Ablex Publishing Corp.

Kay, D.S. and Black, J.B. (1984). The changes in knowledge representations of computer systems with experience. *Proceedings of the Human Factors Society 28th Annual Meeting 1984*. San Antonio, TX.

Larkin, J.H. (1981). Enriching formal knowledge: A model for learning to solve textbook physics problems. In J.R. Anderson (Ed.) *Cognitive Skills and their Acquisition*, Hillsdale, NJ: Erlbaum.

Lewis, C. (1981). Skill in algebra. In J.R. Anderson (Ed.) *Cognitive Skills and their Acquisition*, Hillsdale, NJ: Erlbaum.

McKeithen, K.B., Reitman, J.S., Rueter, H.H., Hirtle, S.C. (1981). Knowledge organization and skill differences in computer programmers. *Cognitive Psychology*, **13**, 307-325.

Robertson, S.P. and Black, J.B. (1983). Planning in text editing behavior. *Proceeding of the Chi'83 Conference on Human Factors in Computing Systems*. Boston, MA.

Sebrechts, M.M., Black J.B., Galambos. J.A., Wagner, R.K., Deck, J.A., and Wikler, E.A. (1983). The effects of diagrams on learning to use a system. (Report No. 2). Learning and Using Systems, Yale University, New Haven, CT.

PURPOSE-DIRECTED ANALOGY

Smadar Kedar-Cabelli

Department of Computer Science
Rutgers University
New Brunswick, NJ 08903

Abstract

Recent artificial intelligence models of analogical reasoning are based on mapping some underlying causal network of relations between analogous situations. However, causal relations relevant for the purpose of one analogy may be irrelevant for another. We describe here a technique which uses an explicit representation of the purpose of the analogy to automatically create the relevant causal network. We illustrate the technique with two case studies in which concepts of everyday artifacts are learned by analogy*.

[A]ny two things which are from one point of view similar may be dissimilar from another point of view.

-K. Popper, The Logic of Scientific Discovery

I Introduction

A recent development in artificial intelligence (AI) research on analogical reasoning has been to recognize that analogy involves mapping some underlying causal network of relations between analogous situations [2, 3, 5, 26]. Often, however, there are numerous causal networks describing the situations. Which are the relevant ones to map when performing a particular analogy? Causal relations relevant for the purpose of one analogy may be irrelevant for another. A more robust model of analogical reasoning cannot always reason from predefined causal networks, but needs the ability to automatically generate the appropriate network based on the purpose of the analogy being performed.

This paper describes a new technique, *Purpose-Directed Analogy*, which is designed to address the above limitation using a specialized notion of 'purpose' to automatically generate the relevant causal network. In particular, we are developing a system to learn concepts of everyday artifacts by reasoning analogically from a known example of an artifact to an unknown example. The specialized notion of 'purpose' is the purpose for which these artifacts will be used.

*This research is supported by GTE Laboratories, under Contract No. GTE840917.

Artifacts can be viewed as objects designed to enable people to perform certain actions (chairs to sit on, pens to write with, and so on). If the goal of an agent is to perform an action, often the agent may need to recognize an artifact which will enable him to perform that action (or a plan of actions leading to the goal). One way to recognize such an artifact is by reasoning analogically from a known example of the artifact to an unknown example. The central idea is that:

> *Two examples will be considered analogous if they share a network of relations which demonstrates how both can be used for the same purpose.*

Thus, performing an analogy can be viewed as a subprocess of a more global problem solving process (as in [11]): to enable an agent to proceed with an action he desires to perform by being able to recognize objects that facilitate that action.

For example, suppose an agent is thirsty, and would like to drink hot liquids. Assume that as a result, the agent wants to learn the concept HOT-CUP: objects whose purpose is to enable the drinking of hot liquids. One way to learn the concept is to be able to determine if a new example (a styrofoam cup, say) is analogous to a known, prototypical example (a ceramic mug) in ways relevant for the purpose of a HOT-CUP. If a cup were needed for a different purpose (ornamental or religious, say), a different network of relations would be relevant.

Section II presents a unifying framework for concept learning by analogy in order to compare existing models and point to a key limitation. Section III describes the Purpose-Directed Analogy technique. Section IV illustrates the technique with two case studies. One case study involves learning the concept of a cup for the purpose of drinking hot liquids. The second case study involves learning the concept of a vehicle in the context of identifying vehicles violating the legal statute "A vehicle is prohibited in a public park". We conclude in section V with a discussion of limitations of the technique, future work, and a summary.

II Related Research and a Limitation

A. Discussion

A common view is that analogy is powerful because it allows us to learn about an unfamiliar situation by

mapping over many aspects of a familiar situation with a dramatic savings in reasoning. To highlight the directionality in the mapping, the familiar situation is often referred to as the *base situation* from which aspects are mapped over to the unfamiliar, or *target situation*, [5]. Thus in the analogy "Science is like a jigsaw puzzle", the less well-understood process of scientific discovery is likened to the working out of a jigsaw puzzle--a more familiar activity. As a result, many of the properties of scientific inquiry are highlighted, without needing separate explanation.

We present in this section a simple, four-stage unifying framework that describes existing AI models of concept learning by analogy. (In fact, this framework encompasses other forms of analogical reasoning such as problem-solving by analogy and metaphor comprehension [8].) We then discuss three such models [5, 2, 26] from this common perspective. We examine the limitation which we address in this paper: the inability of these models to automatically generate a causal network relevant for the purpose of a particular analogy.

B. Concept Learning by Analogy: Unifying Framework

The problem of concept learning by analogy, and the four-stage unifying framework for solving it, is stated in figure II-1.

We illustrate each stage by the analogy "The hydrogen atom is like our solar system" from [5]. The framework we present is slightly more general than the models it describes: most of these models simplify the reasoning by supplying the base example instead of retrieving it. In this analogy, the potentially analogous base concept 'solar system', is provided as input, rather than *retrieved*.

First, independent relations and causal networks of relations describing the base concept are *derived*. By a *causal network of relations*, we mean a set of relations related by any higher order relations such as 'physical-cause(ri,rj)', 'logically-implies(ri,rj)', 'enables(ri,rj)' and so on. (This is a broader sense of 'causal' than is sometimes used [5].) *Independent relations* are those not belonging to a causal network. The causal network of relations in this example describes that 'the sun attracting the planets *causes* the planets to orbit the sun'. Next, the causal network is *mapped* from the base concept over to the target, to explain why the electrons orbit the nucleus of the atom. Finally, the correctness of the mapping is *justified*: that in fact 'the nucleus attracting the electrons causes the electrons to revolve around the nucleus.'

The unifying framework does not perform any concept learning, in the sense that it does not modify the the system's representation of the target concept in any way. In order to model concept learning following the analogical reasoning, this framework is used in conjunction with (possibly) three subsequent stages. First, the concept may be learned by simply *retaining* the causal structure which was mapped to the target concept. For instance, more is learned about the atom

Figure II-1: Unifying Framework for Concept Learning by Analogy

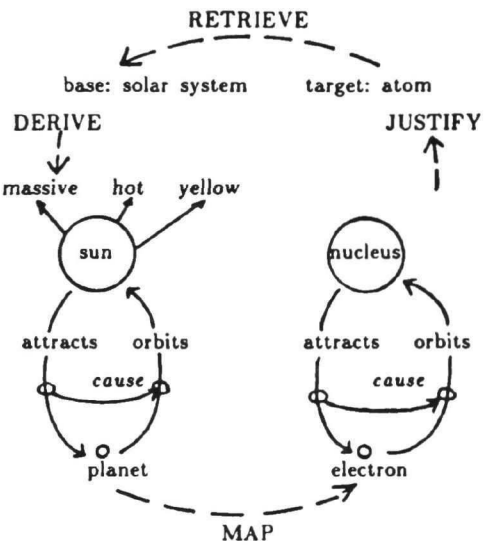
Given:

- a new, target concept, (e.g. the atom)

Find:

- a familiar, base concept, (e.g. the solar system)
- causal networks of relations of the base concept, and
- causal networks of relations of the target concept derived from the base concept

Process:



by retaining the causal structure describing why the electrons orbit the nucleus. In addition, concept learning might involve forming a *generalization* of the target and base, as in [26]. A generalized concept of 'attractive force' may be learned as a result of the above analogy. Furthermore, learning could involve *debugging* or refining a 'faulty' causal structure or generalization, by repeated analogical reasoning with the same, or different, base concepts [2, 28]. For example, the description of the atom's physical mechanisms may only be partially correct, and may be revised by analogy to other concepts.

Given the above framework, we can now discuss three recent models of concept learning by analogy [5, 2, 26]. (See [6, 8] for surveys of other work on analogy.)

C. Gentner's Domain-Independence Relevance Criterion

The central idea in Gentner's structure-mapping theory [5] is that a syntactic (domain-independent) principle can be used to select the relevant aspects of

situations for any analogy. This *systematicity principle* states that, in general, causal networks of relations are relevant to the analogy between situations, while independent relations are not. The justification is that analogy is defined as a reasoning process which maps over a "...system of connected knowledge, not a mere assortment of independent facts" [5, p.162]. Thus, as we saw earlier, in the analogy "The hydrogen atom is like our solar system" more is understood about the atom by mapping over causal relations. Specifically, the causal network describing why the planets orbit the sun is mapped to explain why the electrons orbit the nucleus of the atom. Note that the analogy is not intended to teach us that the nucleus of the atom is 'yellow, hot or massive' like the sun. These independent relations, not involved in the causal network, are considered irrelevant to the analogy.

Gentner's model assumes that the *relevant* causal network is given. For a different purpose, a different causal network may be relevant. Consider, for example, a different analogy with the sun: the metaphor "Juliet is the sun", from Shakespere's *Romeo and Juliet* (also discussed in [5]). We know the context in which this metaphor is conveyed: that Juliet is a woman and Romeo loves her. The purpose of this metaphor is to analogically *convey positive qualities about Juliet*, not to *convey anything about physical mechanisms!* Thus the causal network about the sun which was supplied for the previous analogy is no longer relevant.

D. Burstein's Automatic Indexing Into the Relevant Network

Burstein's model [2] also relies on the domain-independent criterion stated above. However, his model is a step closer to automatically selecting the relevant causal network among many candidate networks: it is provided with a relation used to index into the relevant causal network. One specific analogy he uses to illustrate his work is: "A variable is like a box, in that numbers can be inside variables in some ways similar to the way objects can be inside boxes". [2]. The action by which 'variable' is analogous to 'box' is explicitly supplied: they are analogous by the fact that things can be 'put inside' them. This eliminates considering many irrelevant actions involving boxes (such as stacking boxes, playing with boxes, etc.). Given the 'put-inside' action, the system is able to automatically retrieve the relevant goal/plan structure related to it: the 'store' plan is retrieved, which describes related actions such as putting things in boxes, taking things out of boxes, etc. (Actions can be thought of as relations, and the plan structure which connects actions in a higher order 'enable' relation can be viewed as the causal network of relations.) This goal/plan structure is then mapped to 'variable', to learn about storing things in variables, taking things out of variables, and so on, by analogy to boxes.

If the relevant action were not supplied, however, many actions and goal/plan structures associated with 'box' could be considered when trying to understand the analogy. Consider a student trying to understand this analogy. He will immediately eliminate many of these as being irrelevant. He is not likely to infer that variables can be 'stacked' like boxes, or that variables can be 'played with' like boxes. Why is that? A student learning about variables knows the purpose of the analogy: *to learn a command in a computer language*, and commands in a computer language enable the computer to manipulate numbers and symbols. Given several goal/plan structures, the student might dismiss 'play' or 'stack' as irrelevant for the purpose of the analogy. 'Put inside' might finally be focused on as the relevant action, and 'store' as the related goal/plan structure. So although Burstein's model is provided with an action 'put-inside' which can be used to automatically index into the relevant goal/plan structure 'store', it is supplied with exactly the relevant action, and cannot reason from the purpose of the analogy to select that action automatically.

E. Winston's Learning from Precedents and Exercises

The main scenario for learning and reasoning in Winston's work on analogy is one of guided learning (e.g. [26]). Here, a teacher supplies the system with a precedent. For instance, the system is provided with part of the Macbeth plot, describing Macbeth's relationship to Lady Macbeth, and what causes him to aspire to become king. The system is also given an exercise which describes personalities and relationships among some people. The task is to show that in the exercise 'the noble may want to be king,' by analogy to the precedent. This is accomplished by mapping a portion of the causal network shared by the precedent and the exercise. If in the precedent these relations are causally connected to the relation 'Macbeth may want to be king', then it can be (plausibly) concluded that in the exercise 'the noble may want to be king'.

Although Winston admits that "...the way things are matched depends on purpose as well as on experience" [25, p.6] currently just the appropriate causal structure needed to make the analogy was supplied. If, however, an analogy between the Macbeth story and the exercise were performed not for the purpose of *understanding Macbeth's motives*, but rather to *to understand Lady Macbeth's motives*, say, different causal relations would be considered important.

III Purpose-Directed Analogy

A. Discussion

We have argued above that Gentner's systematicity principle, Burstein's indexing into the relevant causal network, and Winston's analogies between precedents and exercises are all limited in their ability to automatically generate the network relevant for the purpose of a particular analogy, since explicit knowledge of purpose is not supplied as an input in these models.

Purpose-Directed Analogy attempts to overcome this limitation by making a specialized notion of 'purpose' an explicit input to the analogy. It uses this 'purpose' to automatically generate the relevant causal network for learning concepts by analogy. In this section we present the statement of the general problem, and the technique introduced to solve it. Section IV illustrates the technique by solving this problem in two case

studies of learning concepts of everyday artifacts. We are illustrating an initial design and partial implementation, not a fully implemented system. We have recently begun an implementation of a prototype system in PROLOG, a logic programming language [12]. (A PROLOG program consists of a set of horn clauses, a subclass of logical implications. The computation is based on resolution theorem-proving.)

B. Statement of the General Problem

We first introduce some terminology. A *concept* is a set of elements. The *goal concept* is the concept currently being learned by analogy. A *concept definition* provides a specification of logically necessary and sufficient conditions for being an element of the set, while a *sufficient concept definition* provides sufficient conditions only. An *example* of a concept is defined as an element of the set. The *domain theory* consists of default IF-THEN rules (axioms) and action operators which represent what is typically true in a real world domain. An *explanation* of how an example is a member of a concept is a proof that the example is an element of the set. The explanation can be viewed as a causal network of relations, consisting of domain-theory rules which link properties of examples, actions, and goals with the relation 'enables(ri,rj)' and 'logically-implies(ri,rj)'. An explanation *relevant for a particular purpose* can be viewed as a causal network of relations all of whose relations are related, either directly or by transitivity, to relations representing the purpose.

Concept learning by analogy as considered here differs slightly from that studied by Gentner, Burstein, or Winston. The analogy is *not* made between base and target *concepts*, but rather between base and target *examples* of the concept.

The problem, and the four-stage technique for solving it, is stated in figure III-1.

The system first *retrieves* a known, base example of the goal concept. The system then *explains* to itself how this example satisfies the purpose of the concept using the domain theory. (We make the simplifying assumption that there is a single purpose, which is given.) More precisely, using AI planning terminology, if the purpose of an artifact is to *enable an agent to perform a goal action*, then the artifact will satisfy the purpose if its structural features *enable a plan of actions* leading to the goal. It will enable a plan of actions if it *satisfies those preconditions of the actions in which it is involved*. So for example, a ceramic mug will enable an agent to drink hot liquids if it enables those preconditions of actions in a plan leading to DRINK in which it is involved: that is, if it enables the agent to PUTIN the hot liquids (i.e. pour), KEEP the hot liquid in the cup for some interval of time, GRASP the cup with the hot liquids in order to PICKUP, and finally if it enables the agent to DRINK the hot liquids. The prototypical ceramic mug clearly satisfies these preconditions with its open concavity, its non-porous, insulating material, its flat bottom, handle, and light weight.

The styrofoam cup will be considered analogous to

Figure III-1: Purpose-Directed Analogy

Given:

- goal concept (e.g. HOT-CUP)
- purpose of goal concept (e.g. enable an agent to drink hot liquids)
- domain theory (e.g. axioms such as '∀x has-part(x,handle) ⇒ graspable(x)')
- a new, target example (e.g. styrofoam-cup1)

Find:

- a familiar, base example (e.g. ceramic-mug1),
- an explanation of how the base example is a member of the goal concept (e.g. how ceramic-mug1 is a HOT-CUP), and
- an explanation of the target example is a member of the goal concept derived from the explanation of the base example (e.g. how styrofoam-cup1 is a HOT-CUP)

Process:



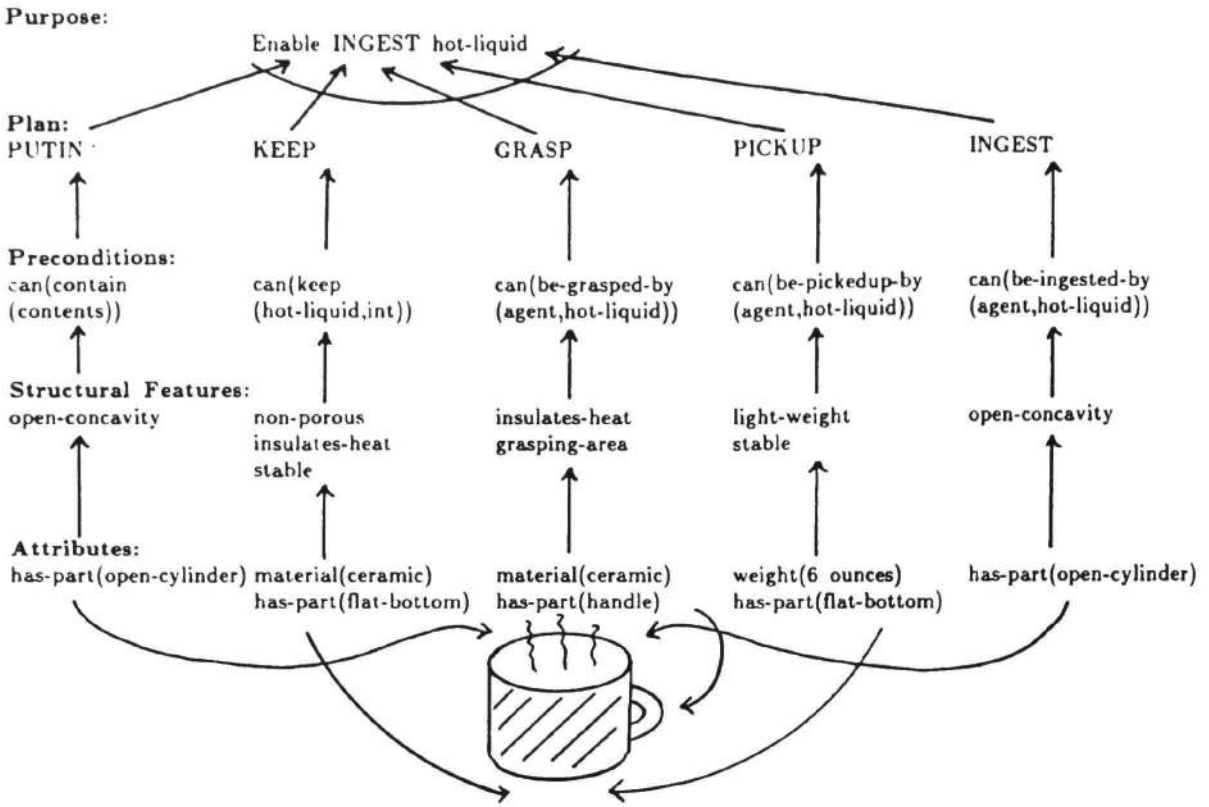
---

the ceramic mug if it too can be used for the stated purpose. To show that, the system *maps* the explanation derived for the ceramic mug, and attempts to *justify* that it is satisfied by this example. The styrofoam cup satisfies the explanation, although with slightly different structural characteristics. It differs structurally in that the styrofoam, not ceramic material, provides insulation; and the conical shape, rather than the handle, makes it graspable.

C. Relationship to Explanation-Based Generalization

The research described here adapts recent techniques for performing goal-directed and explanation-based generalization [4, 11, 14, 17, 18, 19, 27]. One key feature of these techniques is that the relevant aspects of a single example can be extracted by generating an explanation of how the example satisfies a particular goal, or purpose.

Figure IV-1: Explain How the Ceramic Mug is a HOT-CUP



In adapting these techniques to analogy, the distinction between analogy and generalization has somewhat blurred. While in analogy the explanation is mapped from a known example and modified to fit the new example; in generalization, the explanation is generated anew for each example. Is there an advantage to modifying explanations rather than generating them each time? Although it seems plausible that modifying explanations is computationally more efficient, we do not yet have experimental data to support this. One can argue, however, that observing multiple examples and modifying the explanation slightly each time provides a principled way of learning alternate ways of satisfying a particular goal or purpose (see also [13]). Current generalization techniques which analyze a *single example* do not have this capability.

The work described here is most closely related to Winston's [27], where the relevant *structural* features of an example of an artifact are extracted by explaining how the example satisfies some pre-defined *functional* features. We extend this work by providing the ability to automatically derive relevant *structural and functional features* from an explicitly given *purpose*.

IV Case Studies

A. Discussion

In this section we illustrate the technique by two case studies. Our case studies illustrate the problem of refining concepts of artifacts, by analogy, based on the specialized purpose for which these artifacts are to be used. Often when learning a concept, some notion of the concept is already known, and the task is to modify it slightly as it is used in a different context. To simplify our technique conceptually, we assume that the known purpose of the artifact, constrained by the specialized purpose for which it is intended, is the 'purpose' input to the system. For example, if the system is to learn the concept of 'vehicle' in the context of prohibiting vehicles from being driven in the park, we assume that in the known purpose of vehicles (to enable transportation), constrained by the context (interfering with park use) is the 'purpose' input to the system: i.e. vehicles that 'enable transportation but interfere with park use'.

B. Case Study 1: A Cup for Drinking Hot Liquids

In the case study described below, a system for performing Purpose-Directed Analogy takes as input the goal concept (HOT-CUP), its purpose (to enable an agent to drink hot liquids), a target example (a styrofoam cup), and domain theory (typical actions an agent can perform, a structural and functional model of the artifact). Then, by analogical reasoning to a known base example of a HOT-CUP (a ceramic mug), the system determines how the target example (styrofoam cup) is a member of the concept (HOT-CUP), derived from the explanation of how the base example is a member of the concept.

We now detail each step of the technique.

1. RETRIEVE step

Given the goal concept, this step retrieves a prototypical base example of the goal concept. Specifically, a prototypical example of a HOT-CUP (a ceramic mug) is retrieved. To simplify the problem, we assume that a prototypical example is known, and stored in such a way that it can be easily retrieved (as an instance of the general concept in an instance/class hierarchy).

2. EXPLAIN step

The next step uses the domain theory to explain how the base example (ceramic mug satisfies the purpose (to enable an agent to drink hot liquids) (see figure IV-1). This is the crux of Purpose-Directed Analogy: in this step the relevant explanation is automatically derived, given explicit purpose for which the artifact is used.

The explanation step consist of two parts: first, *derive a general explanation* of how an example can satisfy the purpose of the goal concept; second, *recognize* that features of the example in fact satisfy the explanation.

Derive a General Explanation: First, given the purpose of the goal concept and domain theory, a general explanation of how an example satisfies the purpose of the goal concept is derived. The purpose of HOT-CUP can be stated in a PROLOG-like representation as follows:

purpose(object,
  enable-action(object,Ingest(agent, hot-liquid, object)))
⇐ hot-cup(object)

In words: if something is a HOT-CUP, its purpose is to enable an agent to 'ingest' hot liquids. If the purpose of a HOT-CUP is to enable ingesting hot liquids, then an example of a HOT-CUP will satisfy the purpose if it enables a plan of action leading to the goal. A planner (as in [12]) generates a prototypical plan (or 'script') which leads to the goal action 'ingest': The plan is:

Putin(agent, liquid, object)

Keep(agent, liquid, object, time-interval)

Grasp(agent, object, liquid)

Pickup(agent, object, liquid)

Ingest(agent, liquid, object)

Each action has a list of preconditions which must be true in order to enable the action. The *object preconditions* are those preconditions which must be true of the *object* in order to enable the action. For an artifact to enable the plan of actions, it is expected to satisfy the object preconditions of each of the actions in the plan. The object preconditions are:

object preconditions for 'Putin':
  can(contain(object, contents))

object preconditions for 'Keep':
  can(keep(object, hot-liquid, time-interval))

object preconditions for 'Grasp':
  can(be-grasped-by(object, agent, hot-liquid))

object preconditions for 'Pickup':
  can(be-pickedup-by(object, agent, hot-liquid))

object preconditions for 'Ingest':
  can(be-ingested-with(object, agent, hot-liquid))

In general, the preconditions are collected together by a method such as goal regression [22] which collects only those preconditions not directly enabled by previous actions, and keeps track of other constraints among the preconditions.

The output of this first part is a general explanation of the preconditions which an example is expected to satisfy in order to fulfill the purpose of a HOT-CUP.

Recognize Example as Satisfying the Explanation: Given the general explanation, the base example, and domain theory, this step verifies that, in fact, the base example (a ceramic mug) satisfies the explanation for membership in HOT-CUP (closely related to plan recognition [23].) An artifact can satisfy these preconditions, or *functional requirements*, by certain structural characteristics. These can be satisfied, in turn, by particular attributes of the artifact.

The example is represented as a frame, with attributes represented by slots and values. A frame in a PROLOG-like representation is a list of binary predicates [12]. The frame describing ceramic-mug1 is:

manufacturer(ceramic-mug1, abc-co))
serial-number(ceramic-mug1, 72118))
color(ceramic-mug1, blue)
material(ceramic-mug1, ceramic)
weight(ceramic-mug1, 6-ounces)
has-part(ceramic-mug1, flat-bottom)
has-part(ceramic-mug1, open-cylinder)
has-part(ceramic-mug1, handle)

The domain theory contains 'default' rules, representing typical structural and functional characteristics of an artifact. An *enable structure* rule expresses how a general structural attribute can typically be satisfied by a particular attribute of an object. For instance:

structure(object, open-concavity) ⇐
  has-part(object,open-cylinder)

The *enable function* rule expresses how a functional requirement can typically be satisfied by a structural attribute. For example:

enable-function(object,
  can(contain(object, contents)) ⇐
  structure(object, open-concavity)

Recognition proceeds as a search for rules to generate a proof that attributes of the ceramic mug satisfy the functional requirements. The resulting explanation is as follows: (see Figure IV-1 for an illustration): Since the shape of the mug is an open cylinder, it has an open concavity which allows hot liquids to be PUTIN it (i.e. poured in). The ceramic material of the mug provides a non-porous material which also insulates the heat, and the flat shape of its bottom makes it stable--all which enable the cup to KEEP the hot liquid for some interval. Its handle and insulating material makes it GRASPable. Its weight (6 oz.) makes it light-weight, and that, along with its stability, enable an agent to PICK it UP. Finally, enabling all the previous actions, along with having an open concavity, allows the agent to perform his goal action of INGESTing the hot liquids from the ceramic mug.

The output of this step, then, is an explanation of how the base example (the ceramic mug) satisfies the purpose of HOT-CUP.

3. MAP Step

This step copies the explanation of the base example over to the target example (the styrofoam cup).

4. JUSTIFY Step

This step takes as input the explanation mapped over, the target example, and domain theory, and attempts to justify that the explanation is satisfied by the target example. If it cannot justify it using the explanation as its stands, it modifies the explanation to show that the target example is a member of the goal concept in a slightly different way.

First, it attempts to show how the attributes of the styrofoam cup satisfy the structural and functional requirements of something that is a HOT-CUP in the same way as the ceramic mug. If it fails to do that, it attempts to modify a portion of the explanation to show that the functional requirements are satisfied by alternative structural features. If it is unable to do that, it attempts to show that alternative actions satisfy the agent's goal action. If that is unsuccessful, the justification step fails. This processing is similar in spirit to derivational analogy [3], and partial provisional planning [24].

In this example, the styrofoam cup satisfies most of the structural and functional requirements in the same way as the ceramic mug. It differs structurally only in that it is the styrofoam material, not ceramic, which insulates the heat; and it is the conical shape, rather than a handle, which makes it graspable. Since the styrofoam cup also fits these relevant functional requirements and therefore the purpose, even if with different structural characteristics, it is considered analogous to the ceramic mug, and may also be classified as a HOT-CUP.

The result of this step is a (possibly modified) explanation of how the styrofoam cup satisfies the purposes of a HOT-CUP.

5. Learning

Given the two explanations as input, learning is achieved first by retaining the two explanations derived by the system. This provides the system with the ability to classify the target example as a member of the concept. Next, the system proceeds to form a generalization based on the explanations generated for two examples. Given these two explanations, the system can summarize the common structural characteristics (and when finding none in common--the functional ones) to form a sufficient definition of the goal concept. Thus the output of this step is a sufficient definition of a HOT-CUP: an object which can have an open concavity, can be made of nonporous, insulating material, can be stable, lightweight, and can be graspable. This sufficient definition can be used from now on to recognize examples of a HOT-CUP more easily, since it is described in more *operational* terms [20], i.e. in terms of structural, observable characteristics (see section V.A for further discussion of 'operationality').

C. Case Study 2: Vehicle in Park

We are also applying Purpose-Directed Analogy to a more complex case study, that of forming legal concepts by legal reasoning from precedents (initiated within the TAXMAN II project [16, 21]). (For other research on AI and legal reasoning see [9]). Given the legal statute "A vehicle is prohibited in a public park" [7], the task is to learn the concept DISTURBING-VEHICLE, an object which enables driving but interferes with park use. We do not present a detailed solution here. Rather, we sketch it briefly.

A case is brought before the court for violating the statute 'A vehicle is prohibited in a public park'. It is the case of Tommy, an 18-year old, who was found speeding through the park on a bicycle by a policeman. The system performing Purpose-Directed Analogy can be viewed as modelling the task of the prosecuting lawyer. The lawyer will argue that riding a bicycle in the park is analogous to a case where a passenger car was driven into the park, a clear example of a vehicle prohibited in the park. (This style of argumentation from precedents is a common form of legal argumentation.) The argument involves presenting the relevant facts that justify why for this law, the bicycle case is analogous to the case involving a passenger car. Knowledge of the purpose of the vehicles that the law intends to prohibit (DISTURBING-VEHICLES, objects which enable driving but interfere with park use) is used to derive the relevant explanation used in this analogy. The problem of learning the legal concept DISTURBING-VEHICLE by argumentation from precedents, guided by knowledge of legislative intent, more specifically the purpose of DISTURBING-VEHICLE, is thus an instance of the general problem of learning concepts by Purpose-Directed Analogy.

First, the system *retrieves* the clear precedent case (involving a passenger car). We assume that clear precedent cases are known, and can easily be retrieved. Next, the system *explains* why the precedent case has violated this law, and thus involves a DISTURBING-

VEHICLE. The statute's intent is to prohibit driving those vehicles which would interfere with people's use of the park, such as enjoying the serene setting, and the natural habitat provided by the park. Driving a passenger car clearly interferes with these aspects of the park: it makes noise, and thus disturbs the serene setting. It pollutes the air, and may trample the lawn and even small animals, and thus destroys the natural habitat. Next, the system *maps* this explanation to argue (*justify*) that the case involving a bicycle is a DISTURBING-VEHICLE in the same relevant respects. Because the bicycle trampled the lawn, flowers, and sped by park users, it similarly interfered with the serene setting and the natural habitat of the park, and is therefore analogous to the case of a passenger car in the aspects relevant for these purposes, and may also be classified as a DISTURBING-VEHICLE. Finally, the system generalizes to a sufficient definition of DISTURBING-VEHICLE, based on the explanations, so that future cases can be identified more easily as having violated the legal statute.

V Conclusion

A. Limitations and Future Research

We plan to complete the implementation of the system in the near future. In addition, we expect to experiment with the system using case studies of increasing complexity. We also plan to test the system on case studies for learning concepts with alternative purposes. Further, several major theoretical issues still need to be addressed before we have a robust Purpose-Directed Analogy technique.

Generalizing the technique to other domains: We provided an initial design of a technique to learn concepts of everyday artifacts. Can this technique be generalized to other domains such as those studied by Gentner, Burstein, and Winston? We use a very specific notion of purpose, the purpose for which an artifact is intended to be used. 'Purpose' in analogy can express many different intentions. It can refer to the purpose of the agent forming the analogy, the purpose of the agent understanding the analogy, the purpose of the analogy process itself, the purpose of the agent using the concept learned as a result of the analogy, and so on. One important open problem to be solved before the technique can be generalized is to represent classes of purposes, and their relationship to one another.

Deriving the Goal Concept: Currently, the system is given a single concept to learn, and a single purpose. Yet in most real-world forms of learning, there are many concepts to learn. In addition, there are many, sometimes conflicting, purposes for learning. Can a system arrive at the desired concept(s) to learn, and infer the purpose(s) automatically from context? In his research on *contextual learning*, Keller has demonstrated a scenario for this in the context of heuristic search [11]. An important issue for future research is to examine the problem of formulating concepts and purposes in these domains.

Adequacy of The Domain Theory: The explanations derived by Purpose-Directed Analogy are only as adequate as the underlying domain theory used. In the case studies presented here, the system had all the correct domain theory rules needed to derive the explanations. Yet theories of the domains of commonsense artifacts and law are both *inexact* and *information-incomplete*. In fact, the representation of most real-world domains will always be lacking. Learning in these domains will have to account for a weak underlying theory (see [15, 16] for one approach).

The domain theory is inexact in that the axioms represent what is only approximately true. For example, a rule such as 'has(vehicle,Motor) ⇒ pollutes(vehicle,Air)' is not infallible: what if the motor is dead? One issue to deal with is how to learn concepts when exceptions to these rules arise (see also [1]). Both analytic and empirical techniques that deal with exceptions will need to be developed ([28] is one such analytic technique). In addition, the theory in these domains is information-incomplete in that we cannot hope to represent all the needed information about these domains. For example, our representation might be missing the rule 'has-part(x, handle) ⇒ graspable(x)' needed to generate the explanation of how something is a cup. Thus the issue of when to approximate when generating an explanation will come into play. In the long term, techniques will need to be developed to learn these axioms from empirical techniques, or from reasoning from first principles.

Operationality of Definitions: Intuitively, we want our concept definitions to enable agents to easily recognize members of the concepts. When is a concept definition 'operational' in these domains? When is it 'non-operational' [20]? Currently we assume that structural definitions allow a human agent to easily classify examples as being members of a concept (hence are operational), while functional and purposive definitions do not (and hence are non-operational). The notion of concept operationalization has been investigated in [10]. Keller advocates defining operationality in terms of the intended use of the concept. The intended use in the case studies examined here seems to be one of classification. An interesting open issue to explore is whether knowledge of the type of domain (artifacts, say) and the intended use of a concept (classification, say) can be used to automatically define operational and non-operational languages for defining concepts in a given domain.

B. Summary

To summarize, we have outlined a framework of existing models of analogical reasoning, within which we discussed three existing models of concept learning by analogy. We argued that a key limitation is that these models cannot automatically generate the causal network of relations relevant for the purpose of a particular analogy. We then introduced an initial design for Purpose-Directed Analogy in concept learning, which addresses this limitation by using a specialized notion of purpose to automatically derive the relevant explanation (causal network). This specialized purpose is the

purpose for which an artifact is intended to be used. Given explicit knowledge of the purpose of the artifact, two examples are considered analogous by the system if they share an explanation which proves that both can be used for the same purpose. We illustrated the technique with two case studies of learning concepts of everyday artifacts.

Building a machine that learns by analogy and reasons in commonsense domains is still beyond our abilities, yet we are slowly progressing toward that goal.

VI Acknowledgments

Thanks go to Tom Mitchell, my thesis advisor, who strongly influenced the work and its presentation. I would also like to thank Thorne McCarty for his guidance. Rich Keller's thesis research influenced the goal and context-directed nature of this investigation. I would like to thank Jack Mostow for his thoughtful comments on the research and drafts of this paper. Thanks also go to Rich Keller, Sridhar Mahadevan, Mike Sims, Chuck Schmidt, Louis Steinberg, N.S. Sridharan, Saul Amarel, Prasad Tadepalli, Keith Williamson, and other Rutgers and GTE colleagues who provided useful suggestions during the course of the research, and helpful comments on drafts of this paper.

References

1. Borgida, A., Mitchell, T. and Williamson, K. E. Learning Improved Constraints and Schemas from Exceptions in Data and Knowledge Bases. In *On Knowledge Base Management Systems*, Brodie, M. L. and Mylopoulos, J., Eds., Springer Verlag, New York, NY, 1985. forthcoming.

2. Burstein, M. H. A Model of Learning by Incremental Analogical Reasoning and Debugging. Proceedings AAAI-83, Washington, D.C., August, 1983, pp. 45-48.

3. Carbonell, J. G. Derivational Analogy and Its role in Problem Solving. Proceedings AAAI-83, Washington, D.C., August, 1983, pp. 64-69.

4. DeJong, G. Acquiring Schemata Through Understanding and Generalizing Plans. Proceedings IJCAI-8, Karlsruhe, West Germany, August, 1983, pp. 462-464.

5. Gentner, D. "Structure Mapping: A Theoretical Framework for Analogy". *Cognitive Science 7*, 2 (April-June 1983), 155-170.

6. Hall, R. P. Analogical Reasoning in Artificial Intelligence and Related Disciplines. Department of Information and Computer Science,University of California, Irvine, February, 1985.

7. Hart, H. L. A. "Positivism and the Separation of Law and Morals". *Harvard Law Review 71* (1958), 593-629.

8. Kedar-Cabelli, S. Analogy - From a Unified Perspective. DCS-TR-146, Laboratory for Computer Science Research, Rutgers University, July, 1985. forthcoming.

9. Kedar-Cabelli, S. Analogy with Purpose in Legal Reasoning from Precedents. LRP-TR-17, Laboratory for Computer Science Research, Rutgers University, July, 1984.

10. Keller, R. M. Learning by Re-expressing Concepts for Efficient Recognition. Proceedings AAAI-83, Washington, D.C., August, 1983, pp. 182-186.

11. Keller, R. M. Sources of Contextual Knowledge for Concept Learning. Unpublished Thesis Proposal, July 1984, Rutgers University Department of Computer Science.

12. Kowalski, R.. *Logic for Problem Solving*. Elsevier North Holland, Inc., New York, NY, 1979.

13. Lebowitz, M. Concept Learning in a Rich Input Domain: Generalization-Based Memory. In *Machine Learning: An Artificial Intelligence Approach, Vol. 2*, Michalski, R. S., Carbonell, J. G., and Mitchell, T. M., Ed., Morgan Kaufmann, Los Altos, CA, 1985. forthcoming.

14. Mahadevan, S. Verification-Based Learning: A Generalization Strategy for Inferring Problem-Decomposition Methods. Proceedings IJCAI-9, Los Angeles, CA, August, 1985. .

15. McCarty, L. T. and Sridharan, N. S. The Representation of an Evolving System of Legal Concepts: II. Prototypes and Deformations. Proceedings IJCAI-7, Vancouver, B.C., Canada, August, 1981, pp. 246-253.

16. McCarty, L. T. and Sridharan, N. S. A Computational Theory of Legal Argument. LPR-TR-13, Laboratory for Computer Science Research, Rutgers University, January, 1982.

17. Minton, S. Constraint-Based Generalization: Learning Game-Playing Plans From Single Examples. Proceedings AAAI-84, Austin, TX, August, 1984, pp. 251-254.

18. Mitchell, T. M. Learning and Problem Solving. Proceedings IJCAI-8, Karlsruhe, West Germany, August, 1983, pp. 1139-1151.

19. Mitchell, T. M., Mahadevan, S. and Steinberg, L. LEAP: A Learning Apprentice for VLSI Design. Proceedings IJCAI-9, Los Angeles, CA, August, 1985.

20. Mostow, D. J. Machine Transformation of Advice into a Heuristic Search Procedure. In *Machine Learning*, Michalski, R. S., Carbonell, J. G. and Mitchell, T. M., Eds., Tioga, Palo Alto, CA, 1983.

21. Nagel, D. Concept Learning by Building and Applying Transformations Between Object Descriptions. LPR-TR-15, Laboratory for Computer Science Research, Rutgers University, June, 1983.

22. Nilsson, N. J.. *Principles of Artificial Intelligence*. Tioga, Palo Alto, CA, 1980.

23. Schmidt, C. F, Sridharan, N. S. and Goodson, J. L. "The Plan Recognition Problem: An Intersection of Psychology and Artificial Intelligence". *Artificial Intelligence 11* (1978), 45-83.

24. Schmidt, C. F. Partial Provisional Planning: Some Aspects of Commonsense Planning. In *Formal Theories of the Commonsense World*, Hobbs, J. and Moore, R., Eds., Ablex Publishing Co., Norwood, NJ, 1985, pp. 227-250.

25. Winston, P. H. Learning New Principles from Precedents and Exercises: The Details. M.I.T. AI Lab, May, 1981.

26. Winston, P. H. "Learning New Principles from Precedents and Exercises". *Artificial Intelligence 19*, 3 (November 1982), 321-350.

27. Winston, P. H., Binford, T. O., Katz, B., and Lowry, M. Learning Physical Descriptions from Functional Definitions, Examples, and Precedents. Proceedings AAAI-83, Washington, D.C., August, 1983, pp. 433-439.

28. Winston, P. H. Learning by Augmenting Rules and Accumulating Censors. In *Machine Learning: An Artificial Intelligence Approach, Vol. 2*, Michalski, R. S., Carbonell, J. G., and Mitchell, T. M., Ed., Morgan Kaufmann, Los Altos, CA, 1985. forthcoming.

LEARNING CONCRETE STRATEGIES THROUGH INTERACTION

R.W. Lawler and O.G. Selfridge
GTE Laboratories, Inc., Waltham, Ma. 02254

Abstract

We discuss learning and the adaptive generation of concrete strategies through interactive experience. The domain is the game Tictactoe. The knowledge structures embodying strategies we represent as having three parts: a Goal, a sequence of Actions, and a set of Constraints on those actions (GAC). We simulate such structures in a program that plays Tictactoe against different kinds of opponents. Applying these strategies leads to moves that often result in winning or losing; which in turn leads to the creation of new structures, by modifying the current GACs. These modifications are controlled by a small set of specific rules, so that the GACs are related by the ways modifications can map from one to another. Subject to certain limitations, we do a complete exploration of certain classes of strategy. This learnability analysis takes guidance from previous cognitive studies of a human subject by Lawler. The simulations were performed on a Symbolics 3600 in LISP.

This work avoids abstraction in order to explore learning based on the modification of fully explicit strategies learned through particular experiences. The results are a catalog of specific experiences through which learning occurs within this system and a description of networks of descent of concrete strategies from one another. We conclude with a description of experience motivated analogy, a mechanism proposed for the development of limited interrelations between structures based on particular experiences.

Introduction

The ultimate aim of this work is to model computationally a sequence of developmental steps leading from naive knowledge to domain specific mastery. We start with a model that embodies three well recognized characteristics of children's thinking and with a representation chosen taking guidance from the rich corpus of a detailed case study.[1] The three primary characteristics are egocentricity, concreteness, and interactivity. EGOCENTRICITY we take to mean an original, exclusive focus on one's-own goals and plans. By CONCRETENESS we refer to the importance of particular experience in determining the descriptions of what things are. We take seriously the concreteness of young children's thought, and we are willing to explore how very specific structures function and develop. The self-centeredness of children's play, even their speech (see Piaget, 1926), is characteristic of the naive learner. This leads us to use information structures that focus almost entirely on the intentions and actions known by the player. INTERACTIVITY is central in the twin senses of learning from interaction with others and from the interaction of disparate structures within the self. The development of structures from such a basis is eventually to proceed to mastery through recognizably significant processes such as the following:

- learning new concrete goals and plans as elementary variations of predecessors
- learning constraints upon those plans
- redescribing goals based on the application of plan constraints to goal specifications
- recognizing the applicability of general ideas (such as symmetry)
- internalizing the opponent's role (thus permitting vicarious play)
- developing mental play (thus permitting search for a victory through the tree of possible games)
- achieving a classification-based mastery of possible play in the domain.

We now explore learning-by-interaction of concrete strategies through analysis of computer-based simulations. The puzzle is how to understand adaptivity through interaction with some "other" thing that is not itself well-comprehended.

[1] Chapter 4 of Lawler, 1985, is a detailed analysis of the complete history of one child's play of Tictactoe throughout a two year period. References to data and analyses from that study will be to "Lawler's subject."

Although this exploration takes guidance from the study of a single child, it is NOT a model of a single child's development. The immediate aim of the work reported here is to explore the interrelatedness of the set of concrete strategies, such as can be generated with minimal transformations of a given strategy through purpose governed interactions with an uncomprehended opponent.[2] We believe that development can follow from reflection upon the disparaity between anticipated outcomes of plans and actual outcomes of particular experiences. The cental task of this model is to modify its repertoire of strategies when surprising outcomes occur in playing particular games. The model proceeds by the recognition of new specific goals and the construction of new plans for achieving those goals. This kind of learning embodies the reflexive construction of concrete strategies.

Central ideas for us are the disparateness of structures and their functional lability.[3] We apply the assumption of disparateness in our simulations by creating a separate object for each strategy; each object proposes its next move regardless of the proposals of every other strategy. Our analysis combines (1) a suitable representation of knowledge structures; (2) decisions made to limit the interactions to some manageable number; (3) proposals of learning mechanisms operating on the knowledge structures; and (4) the analysis of sequences of simulations. Simulation lets us specify and activate the structures and developmental sequences we consider important, creating thereby a kind of experimental epistemology. This in turn permits us to connect analysis of the particular context of an incident of learning to specific changes in an organization of self-modifying information structures.

Representation of Knowledge

We represent knowledge structures as having the parts necessary for adaptive functioning. Learning what to do is essential: GOALS are explicitly represented. Knowing how to achieve a goal is essential; ACTION PLANS are explicitly represented. Knowing when a planned action will work and when it won't is essential; CONSTRAINTS limiting application of actions are represented explicitly. The structure composed of this triad, a GAC (Goal, Action, Constraints), is our representation of a strategy for achieving a fork in Tictactoe. Goals are considered as a three element set of the learner's marks which take part in a fork. This is the first element of a strategy. Plans of three step length, which add the order of achieving goal steps, are represented as lists. Constraints on plans are two element sublists, the first element being the step of the plan to which the constraint attaches and the second being the set of cell numbers of the opponent's moves which defeated the plan in a previous game. In our simulations, SLIM (our Strategy Learner, Interactive Model) plays against REO (a relatively expert opponent). REO can win, block, and apply various rules of cell choice – though ignorant of any strategies of the sort SLIM is learning. Within the execution of our simulation, the structure of GAC 1 below will lead to the three games shown depending on the opponent's moves (letters are for SLIM's moves, numbers for REO's):

	GOAL	ACTION	CONSTRAINT
GAC 1:	{1 3 9}	[1 9 3]	< [3 { 2 5 8 }]>
win by plan	plan defeat	constrained draw	cell numbers
A | 3 | C _____ | 1 | D _____ 2 | | B	A | | C _____ 2 | 1 | 3 _____ | | B	A | C | 3 _____ 4 | 1 | E _____ D | 2 | B	1 | 2 | 3 _____ 4 | 5 | 6 _____ 7 | 8 | 9

The player SLIM has no notion of symmetry (ongoing work explores how it may be learned). Our representations and learning mechanisms are committed to cell-specificity; they are also self-centered,

[2] This line of research arose from the attempt to develop the kinds of concrete learning models proposed in Selfridge and Selfridge '85 in a direction less dependent on the external setting of goals.

[3] Satinoff, 1978, presents a physiological example of disparate systems for thermal regulation. Lawler 1985 assumes the disparateness of cognitive structure in interpreting learning in the human case. The functional lability of structures names the effectiveness of a given structure for something other than the purpose which shaped its development. Jacob 1981 argues the importance of functional lability in the evolution of species. The idea is not forward in this article, but is central to the explanation of learning through experience generally. See Lawler 1985, 1985b.

focussing on the learner's own plans and knowledge (as they must since, by principle, SLIM begins not knowing what the opponent will do; SLIM does not have the ability to model or predict an opponent's moves in any abstract way).[4]

Here are the fourteen corner-opening GACs from each of which SLIM began one of its play-sequences against REO (they are grouped in symmetrical quartets; because GACs five and six are doubly symmetrical, there are only two members in its group):

GAC 1 {139}[193]<>	GAC 5 {137}[137]<>
GAC 2 {179}[197]<>	GAC 6 {137}[173]<>
GAC 3 {139}[139]<>	
GAC 4 {179}[179]<>	
GAC 7 {135}[153]<>	GAC 11 {125}[152]<>
GAC 8 {157}[157]<>	GAC 12 {145}[154]<>
GAC 9 {135}[135]<>	GAC 13 {125}[125]<>
GAC 10 {157}[175]<>	GAC 14 {145}[145]<>

The central question is: when SLIM starts from one specific GAC, which others will it discover through experience – given a set of well defined assumptions about learning mechanisms and the opponent, and learning mechanisms. We will describe the central elements of the domain, the opponent, and learning mechanisms.

The Domain

There are at least four ways of mobilizing knowledge to play Tictactoe. Most often, mature people think of playing forward in the tree of possible moves. A second kind of mature play is categorical: one may group all possible pairs of openings and responses, reducing them by symmetry to twelve unique game openings; of these twelve, seven permit direct wins for the initial player and the other five permit setting traps for the unwary opponent. A third way is more primitive than the first two; we call it tactical: TACTICAL play is purely state dependent; from the tokens already played, choices are made to win when possible, to block at need, and to choose one cell from those available, one choice at a time and with no forward play.

The fourth way we call strategic: it depends upon having a multi-move action-plan for achieving a fork, which distinguishes it from tactical play. Initially it completely ignores the opponent's moves, which distinguishes it from look ahead. It involves NO systematic knowledge of the game at all, and does NOT make ANY explicit use of symmetry in play.

Lawler's subject clearly exhibited use of the third and fourth ways, that is tactical and strategic play. SLIM was so programmed, preferring strategic play. But when SLIM's strategies are frustrated, it plays with structured tactics (to be defined shortly). Winning with strategies depends upon SLIM's making a winning move tactically after forking the opponent. SLIM's opponent, REO, ALWAYS plays tactically. Both SLIM and REO can recognize and achieve a win. REO always blocks any threat by SLIM (or one of two when forked). SLIM, not noticing its opponent's moves, initially does not block while executing a strategy – until a defeat establishes its vulnerability, and analysis leads to identifying and constraining execution of the vulnerable plan step.

SLIM plays in two modes, executing strategies when possible and resorting to tactical play when necessary (all strategies frustrated) or appropriate (a fork has been achieved). When playing tactically, SLIM blocks as effectively as REO.[5] When a single strategy is abandoned because of a threat, play reverts to the tactical mode – and thus blocking is effective.

---

[4] The general commitment to egocentric knowledge representation has psychological justification in this specific case. Lawler's subject suffered the defeat above trying to achieve the victory of GAC 1 (the only strategy she knew), not attending to her opponent's move nor anticipating any threat to her intended fork.

[5] SLIM does not block while pursuing a strategy. This seeming inconsistency reflects the behavior of Lawler's subject. The fragmentation of knowledge implied by such behavior is implemented in the model through non-communicating instances of Zetalisp flavors.

SLIM's opponent REO plays tactically in three variations. They can be defined in terms of application of a set of production rules:[6]

<own-winning-pattern>->	play winning-cell	rule	type of play
<opponent-win-pattern>>	play blocking-cell	—	—
<center-cell free>	-> play center-cell	1:	highly structured play only
<corner-cell free>	-> play corner-cell	2:	highly structured and structured play
<any-cell free>	-> play any-free-cell	3:	un-structured play; last choice for others.

When all the last three rules function, the opponent plays in a HIGHLY STRUCTURED fashion. This means that rule 1 inhibits the choice of any corner or side cell when cell 5 is free. When rule 1 is not functioning, the opponent plays in a STRUCTURED fashion. When both rules 1 and 2 are disabled, the opponent plays in an UNSTRUCTURED fashion – moving in any free cell. Since our simulations generate the complete spaces of potential experiences, disabling the limitations implied by one specific rule constraining its successors generates an increased number of games in the space.

Spaces of Potential Experience

The space of possible Tictactoe games is of magnitude 2 x 9!, or 725,760. We have severely limited the portion of that space to be explored in order to permit our tracing how the space of executable games changes with increasing strategic knowledge. Domain oriented limitations we have applied are focussing on corner opening play with SLIM moving first. These choices are based on the significant opening advantage of corner-oriented play and on the important role we assign to winning games as the primary cue for learning a new strategy. These constraints still leave on the order of 8! (40,320) games. We ignore drawn games and have minimized the generation of redundant games (alternative responses to a fork and so forth). We have also omitted the analysis of eight-move losses and nine-move wins as too complex for SLIM; we believe these circumstances are too complex for analysis by naive human players of Tictactoe also. These limitations of choice restrict the games playable, but they also permit us to explore how very simple strategies can generate others which will in turn expand the space of potential experiences.

Beyond these limitations of choice, we apply a more ESSENTIAL limitation. The most natural constraint on play is its specific purpose, in the twin senses of having winning as a general objective and intending to apply knowledge available as a means to that end. If one knows only a single, very specific strategy – such as "move first in this corner, then that other, and finally in the third one there" – and is determined to apply that knowledge, play is very much limited. At the beginning, our computer learner SLIM had such commitments, just like Lawler's subject. Our computer simulations are based on generating the space of KNOWLEDGE-LIMITED POTENTIAL EXPERIENCES. We want to know how different new strategies can be derived from the various outcomes of a single predecessor strategy's play. These descendents we call the FAN-OUT from a strategy. We want to know where you can go from where you start out. Will there be convergence to some single end or residual differences in final states?

We generate the entire space of potential experiences which a learner could have and then analyze all those games where learning occurs through activation of the allowed mechanisms. The learning mechanisms are limited to changing only one element of a structure at a time. Learning through losing games leads to the addition of a constraint element in the third slot of a GAC (if there isn't any GAC to which the constraint may be attached – a frequent situation – no such learning can occur). Learning through winning creates a new GAC, often two. Each has a goal, an action-plan, and an initially empty list of constraints. Whenever a seven-move win occurs, it is always possible to infer the pattern of a new goal.

Differences between methods of GAC creation depend upon how the new plan to achieve the goal is constructed. Because SLIM moves first in cell 1, any new strategy will have plans different in either step two or step three. The first mechanism, plan terminal modification (PTM), changes the final element of a known plan to match the actual win that occurred in a specific game. PTM generates one new GAC, the method is dependable and is always tried before the second, Goal-guided Center Deletion (GCD); GCD is more complex in inference and less certain in value.

[6] Newell and Simon, 1972, pg. 62 shows such a system in detail as an example.

GCD changes the center element of a three-step plan: once a goal has been inferred, SLIM attempts to locate a known goal with at least two elements matching those of the new goal. If there is one, the associated plan is taken as the template for constructing a new plan. The element of the old plan not in the new goal is deleted from the original. The element of the new goal lacking in the original plan must then be inserted. But where? SLIM is not sophisticated enough to know or find out which is appropriate. Consequently, two new GACs are generated with a common goal but with plans differing in the order of the last two steps. The human counterpart would be recognizing a specific new goal but being uncertain as to the order of the last two moves.

Simulations

Strategic play proceeds until it achieves its expected victory – or is blocked or unexpectedly defeated. The typical loss by a strategy is through its circumvention (REO lays down three in a row while SLIM pursues its plan). The application of constraints to plans turns such six-move defeats into draws. The essential analysis is that loss after fork completion implies that the opponent posed a threat after two moves. Since SLIM pays no attention to what the opponent was doing, it has no sense of the order of moves in which the REO's victory was achieved. Therefore, the decision to apply a constraint to an ongoing game tests for the opponent's having acquired any two of the three moves in the triad by which REO lately defeated SLIM. If this be "look-ahead", it is not a general capability but a specific functional prohibition motivated by a prior defeat. A general capability would require further development.

An unexpected win is the cue to begin reflection. SLIM assumes that a victory was won not received as a gift; therefore the moves preceding the win contain a fork. (This works because REO always wins when possible and blocks at need.) Since the most recent move by which a win was completed could not have been part of the fork, the remaining moves are not merely IN the fork, they COMPRISE the fork pattern exactly. A valid forking goal can be found for any seven-move win against a relatively expert opponent.

Play Against a Highly-Structured Opponent

SLIM's processes of learning can best be shown by examples. Against a highly structured opponent (one who ALWAYS moves in the center when it is empty, who ALWAYS prefers a corner move to a side cell), the space of potential experiences is very small, and thus the learning that occurs is quite limited. We exhibit the action of the learning mechanism in particular games. Consider the case where SLIM knows only the strategy represented by the plan [193]. Because of a highly structured opponent's dominant commitment to an initial cell 5 move, only two games are possible.[7] The first is success of SLIM's strategic plan in game [1597326]. The second is a victory achieved after the blockage of SLIM's plan [1593726]. Because SLIM will prefer a corner cell to a side cell, when REO blocks his third plan step by moving in cell 3, SLIM achieves the win of GAC 2's plan [197].

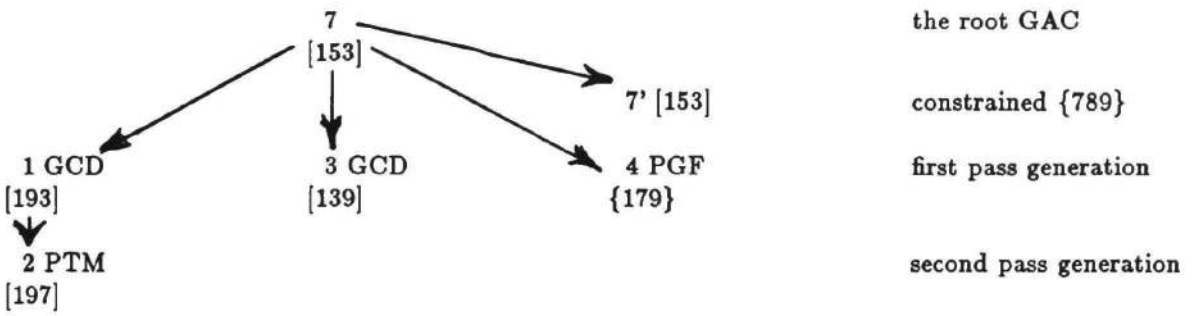
In reflexive analysis after play, SLIM constructs GAC 2. The determination of the goal is 179. The plan derives from this analysis: a win implies a fork. The fork is the three moves preceding the winning move. The order of moves must be similar to the frustrated well-known plan [193], because it was blocked only at move three by a token still available for examination. The move substituted by tactical play for the blocked move remains equally present. The conclusion is that [197] is a good candidate plan for attachment to the goal 179. The GAC is formed by this terminal modification of a previously known plan.

A case of a win for which PTM cannot construct a plan is game [1597326], played with a strategic objective of achieving plan [153]. SLIM abandoned its plan at move two because it was blocked. The form of the fork actually achieved was GAC 1's goal 139. Changing the final step of plan [153] can in no way lead to a plan [193]. The plan was constructed from [153] as a template (only GAC 7 was known at the time) by goal-guided center deletion. In fact, a second plan, [139], was also constructed.

Although one can always infer a new goal from a win, one cannot always construct a new plan for that goal, as with GAC 7's play against a highly structured opponent in game [1593748]. The effective forking pattern is GAC 2, but to create a plan with elements 9 and 7 from a template [153] is not

[7] We will represent games in the text as number strings within brackets. Each digit represents the cell number of a move. The first move is that of SLIM, the second REO's, and so on alternately.

possible by mechanisms that can change only a single element. This is Plan Generation Failure (PGF). In this case of strategies descended from play in the space of GAC 7's potential experience against a highly structured opponent, we can see in miniature the major processes that act in the more extended simulations discussed below.



This tree of structures descended from one strategy is its FAN OUT. Our simulations have documented the fan out for the fourteen GACs specified earlier. Against a highly structured opponent, SLIM is able to learn very little, as shown by the first run summary:

GACs	1	2	3	4	5	6	7	8	9	10	11	12	13	14
games	2	2	1	1	1	1	6	6	2	2	6	6	1	1
wins	2	2	0	0	0	0	2	2	0	0	2	2	0	0
losses	0	0	1	1	1	1	0	0	0	0	0	0	0	0
draws	0	0	0	0	0	0	4	4	2	2	4	4	1	1
new GACs	1	1	0	0	0	0	2	2	0	0	0	0	0	0
constraints	0	0	1	1	1	1	0	0	0	0	0	0	0	0

Playing Against an Opponent of Structured Preferences

Table 1 summarizes the results. Each section shows play begun with a specific GAC as the learning root. For each of the 14 GACs is shown (1) its fan-out of learnable strategies, (2) the specific games through which learning occurred, and (3) the mechanism by which the new structure was generated. SLIM learned everything possible to learn in two passes. Further, the first six GACs all converged to an essentially uniform result wherein they were all known; each of these six GACs initially generated 10 games, while in the final common state 156 games were generated by them. These six thus form a central collection of strategies whose specialness needs be emphasized. Strategy three may be taken as typical of the others. Play in five specific games generates the other five central GACs. A sixth game installs a constraint upon the root GAC of the set. The remaining eight GACs fall into three categories when seen as bases for further learning: feeders (7,8); teasers (9,10,11,12); and mules (13,14). Feeders are strategies that themselves fail but generate games permitting construction of strategies in the central set; consequently, they give access to all other central strategies. When they are frustrated and abandoned, teasers lead to other winning games for which SLIM is unable to construct any plan. Mules are strategies which can win some games against the opponent of structured preferences but which cannot generate wins of any other forking patterns and are thus sterile as generators of new structures.

The specialness of the six central nodes is a consequence of their symmetry in respect of co-generability. Some generate others directly by PTM or GCD. Some generate others only with the intervention of an intermediary GAC (this is the reason that two passes are necessary to complete the central set). Some of those directly generable can generate each other; they are reciprocally generable. Those which are remotely generable nonetheless lead to each other through intermediaries; they are cyclically generable. For these six central strategies, the trees of structure descent fold together into a connected network of descent whose relations of co-generativity are shown in the diagram below; notice that those strategies cyclically generable form the interconnection by GCD between themselves and others reciprocally generable by either PTM or GCD:

Table 1

STRUCTURE GENERATION
OPPONENT WITH STRUCTURED PREFERENCES

ROOT GAC	RUN ID	GENRT GAME	INFRD PLAN	REASON	GOAL/ PLAN or CONSTRNT	LEARNG MECH	GAC ALTERED	COMMENTS
1	1.1	[1974325]	1	root	{137} [137]	GCD	5	Guided center deletion
[193]	1.2	"	"	"	" [173]	GCD	6	creates two GACs.
	1.3	[1593748]	"	"	{179} [197]	PTM	2	Plan tail modification
	2.1	[1732956]	5	01 x L3	{139} [139]	PTM	3	creates one GAC.
	2.2	[1374958]	6	01 x L3	{179} [179]	PTM	4	
2	1.1	[1974325]	2	root	{137} [137]	GCD	5	At the end of the
[197]	1.2	"	"	"	" [173]	GCD	6	runs six central
	1.3	[1795326]	"	"	{179} [197]	PTM	1	plans have been
	2.1	[1732596]	5	01 x L3	{139} [139]	PTM	3	learned. No plans
	2.2	[1374958]	6	01 x Le	{179} [179]	PTM	4	have been constrained. at all.
3	1.1	[1932745]	3	root	{137} [137]	PTM	5	
[139]	1.2	[153298]	"	"	⟨[9{258}]⟩	PCA	3	First constraint made.
	1.3	[1395748]	"	"	{179} [197]	GCD	2	
	1.4	"	"	"	" [179]	GCD	4	Six experiences make
	2.1	[1974325]	4	01 x L3	{137} [173]	PTM	6	one constraint and
	2.2	[1795326]	2	01 x L3	{139} [193]	PTM	1	five new GACS.
4	1.1	[1974325]	4	root	{137} [173]	PTM	6	
[179]	1.2	[1795326]	"	"	{139} [193]	GCD	1	
	1.3	"	"	"	" [139]	GCD	3	
	1.4	[157496]	"	"	⟨[9{456}]⟩	PCA	4	Root plan constrained.
	2.1	[1932745]	3	01 x L3	{137} [137]	PTM	5	
	2.2	[1593748]	1	02 x L3	{179} [197]	PTM	2	Six plans completed.
5	1.1	[1732956]	5	root	{139}[139]	PTM	3	
[137]	1.2	[153278]	"	"	⟨[7{258}]⟩	PCA	5	
	1.3	[1395748]	"	"	{179} [179]	GCD	4	
	1.4	"	"	"	" [197]	GCD	2	
	2.1	[1974325]	4	01 x L3	{137} [173]	PTM	6	Six plans completed;
	2.2	[1795326]	2	01 x L3	{139} [193]	PTM	1	root plan constrained.
6	1.1	[1795326]	6	root	{139} [139]	GCD	3	Symmetry in
[173]	1.2	"	"	"	" [193]	GCD	1	relation to the
	1.3	[157436]	"	"	⟨[3{456}]⟩	PCA	6	the learnability of
	1.4	[1374958]	"	"	{179} [170]	PTM	4	plans is discussed
	2.1	[1932745]	3	01 x L3	{137} [137]	PTM	5	in the text and
	2.2	[1593748]	1	02 x L3	{179} [197]	PTM	2	show in the descent network of Figure 1.

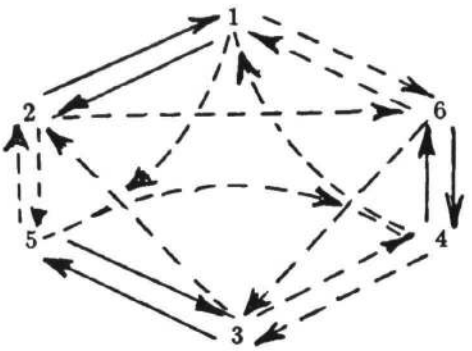
Abbreviations used in the table:

GCD: Goal-guided Center Deletion; 01,02: Opponent moves 1 and 2; PTM: Plan Terminal Modification; L2,L3: Learner moves 2 and 3; PCA: Plan Constraint Application; 01xL2: Move 01 blocks L2; PGF: Plan Generation Failure.

Table 1
STRUCTURE GENERATION:
OPPONENT WITH STRUCTURED PREFERENCES

ROOT GAC	RUN ID	GENRT GAME	INFRD PLAN	REASON	GOAL/ PLAN or CONSTRNT	LEARNG MECH	GAC ALTERED	COMMENTS
7	1.1	[195738]	7	root	⟨[3{789}]⟩	PCA	7	These mechanisms
[153]	1.2	[1597326]	"	"	{139} [139]	GCD	3	don't succeed when
	1.3	"	"	"	" [193]	GCD	1	a goal is not near
	1.4	[1593748]	"	"	{179} [ ]	PGF	.	known games; first example: failure
	2.1	[1974325]	1	01 x L2	[137]	GCD	5	
	2.2	"	"	"	[173]	GCD	6	
	2.3	[1593748]	1	02 x L3	{179} [197]	PTM	2	
	2.4	[1374958]	3	01 x L2	{179} [197]	GCD	2	First example:
	2.5	"	"	"	" [179]	GCD	4	generation.
8	1.1	[195376]	8	root	⟨[7{369}]⟩	PCA	8	3 PLANS IN GENERATION intended; actual; basis
[157]	1.2	[1597326]	"	"	{139} [ ]	PGF	.	[157][193][157]
	1.3	[1593748]	"	"	{179} [179]	GCD	4	[157][193][157]
	1.4	"	"	"	" [197]	GCD	2	[157][193][157]
	2.1	[1974325]	4	01 x L3	{137} [173]	PTM	6	[179][173][193]
	2.2	[1932745]	2	01 x L2	{137} [173]	GCD	6	[197][173][157]
	2.3	"	"	"	" [137]	GCD	5	[197][137][157]
	2.4	[1795326]	2	01 x L3	{139} [193]	PTM	1	[197][193][197]
	2.5	[1732926]	4	01 x L3	{139} [193]	GCD	1	[179][193][179]
	2.6	"	"	"	" [139]	GCD	3	[179][139][179]
9	1.1	[1395748]	9	root	{179} [ ]	PGF	.	No learning follows
[135]	1.2	[1374958]	"	"	{179} [ ]	PGF	.	trying this plan
10	1.1	[1795326]	10	root	{139} [ ]	PGF	.	No learning follows
[175]	1.2	[1732956]	"	"	" [ ]	PGF	.	trying this plan
11	1.1	[195748]	11	root	⟨[2{789}]⟩	PCA	11	First example: two
[152]	1.2	[195326]	"	"	⟨[2{369}]⟩	PCA	11	constraints on a GAC;
	1.3	[1597348]	"	"	{139} [ ]	PGF	.	they need NOT force
	1.4	[1593748]	"	"	{179} [ ]	PGF	.	different games of
	2.1	[1597326]	"	no other	{139} [ ]	PGF	.	or new learning
	2.2	[1593748]	"	"	{179} [ ]	PGF	.	
12	1.1	[195748]	12	root	⟨[4{789}]⟩	PCA	12	Symmetrical case to
[154]	1.2	[195346]	"	"	⟨[4{369}]⟩	PCA	"	failures of GAC 11.
	1.3	[1597326]	"	"	{193} [ ]	PGF	.	
	1.4	[1593748]	"	"	{179} [ ]	PGF	.	2nd run omitted.
13	1.1	[192356]	13	root	⟨[5{369}⟩	PCA	13	The only learning.
14	1.1	[194758]	14	root	⟨[5{789}]⟩	PCA	14	The only learning.

Figure 1
STRATEGY DESCENT NETWORK: STRUCTURED OPPONENT



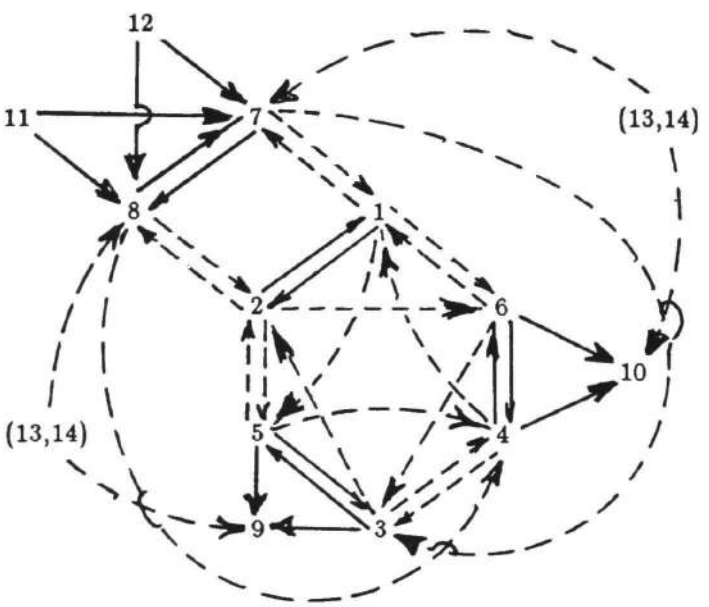
solid arrows show descent by PTM; dashed arrows show descent by GCD

With the two mechanisms assumed and the experience of appropriate specific games, the central strategies will all be learned once any one is known.

Against the opponent of structured choices, the feeder strategies exemplify both plan generation failure and redundant generation. Thus the goal 179 for which SLIM can construct no plan in pass one is regenerated in pass two. In the second pass, the expanded repertoire of known plans permits construction of GAC 2's plan. They also show the value of distinguishing between SLIM's intended plan when a game began, the pattern of the realized fork, and the plan drawn from its repertoire which serves as the template for construction of the new plan. There is no necessary identification of the first with the third. The plans of GACs 9 through 12 are too dissimilar from the actual forks they generate to permit construction of a new plan for the goals SLIM infers.

Figure 2 summarizes the results of comparable simulations of play against an unstructured opponent.

Figure 2
STRATEGY DESCENT NETWORK: UNSTRUCTURED OPPONENT



solid arrows show descent by PTM; dashed arrows show descent by GCD
GACs 13 and 14 appear twice to simplify the network drawing

GACs 1 to 6 still remain central, but the flexibility of the unstructured opponent permits SLIM to learn additional strategies. GACs 13 and 14 remain non-learnable through experience with this opponent. The specific reason is that SLIM's tactical preferences remained structured in these simulations. Thus SLIM, never trying games whose second move is to a side cell, will never win accidentally with a fork containing such patterns; consequently, SLIM can never learn such strategies.

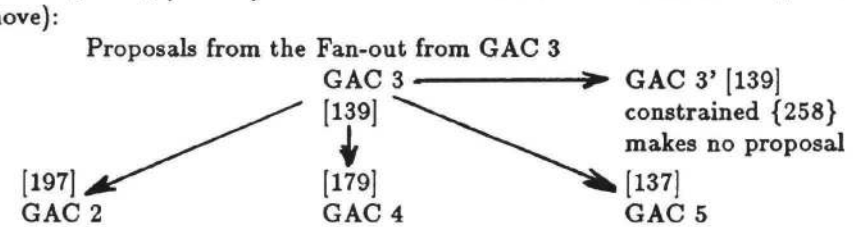
The form of these descent networks is related to symmetry relations among forking patterns, but the networks include more; they reflect also the order in which the fork is achieved, the play of the opponent, and the specific learning mechanisms permitted in the simulations. Whereas fork pattern symmetry can be used as an alternative mechanism for proposing new forks and games to play, these descent networks are summaries of results. The question yet to be answered is whether or not the relations represented in these networks can serve as the basis for the discovery that pattern oriented symmetry functions with power in this domain.

A general observation is that the generation of constraints seems less important than we expected. Against the opponent of structured choices, only the root GAC is constrained. In one sense, this is an artifact of programming, but it points to a significant further issue. Our simulations generate the trees of potential experiences by expanding a list of possible games around the generation of sets of plausible moves according to the appropriate strategic or tactical choices. The set of plausible moves is collected from separate instances of a class of strategy objects which do not communicate with each other. During the first pass, ony the root GAC is accessible to be constrained. During the second pass, the alternative choices offered by the three or four functional strategies are so effective that forking the opponent closes off the possibly vulnerable proposals of its simultaneously active strategies.

Because of this collection process, the proposal of one strategy could at first be blocked by a constraint; yet the "forbidden" game could be generated anyway by an identical cell proposal being advanced by some other strategy for its own plan. This interaction has been suppressed in these simulations – but it does suggest a future direction of development for models of this class: the communication of constraints upon the occurrence of a "forbidden defeat". This can serve as the basis for experience motivated analogy, a new mechanism for the development of organization where none formerly existed.

Experience Motivated Analogy

Beginning with GAC 3 as the root of its strategies (playing against the structured opponent), the first simulation set of games permits generating GACs 2, 4, and 5. A constraint is also put on GAC3. Consider play in the game beginning [1532...] – while these four GACs are simultaneously active – at move 5 (SLIM's third move):



GAC 3, with its established constraint on a third move to cell 9 recognizes the vulnerability of its plan and quits; it marks cell 9 as a forbidden move. This proposal results in a game being played by the tactical routines, which block REO's threat. At this same pass, GAC 5, whose plan is compatible with that of GAC 3 through move 2 and is unconstrained, proposes a move to cell 7. This results in a different game being generated and lost when REO moves next in cell 8. GAC 4 at move 5 proposes a move to cell 7, the second step of its plan. The plan of GAC 4 may be characterized as having been interrupted in execution and as being diluted by an extraneous move. This game loses to REO's threat as well. Gac 2 proposes a move to cell 9, the second move of its plan, which is also interrupted and diluted. This is exactly the move that is constrained in GAC 3. The game is generated and lost.

When these losses occur, the process of blame assignment begins: who proposed these losing moves? Each GAC justifies its behavior. GAC 5 admits to the cell 7 proposal, with the additional information that its plan was uninterrupted and pure (not diluted); SLIM sends a message to GAC 5 that it should constrain itself on the cell 7 move by the threatening pattern 258. GAC 4 also proposed cell 7, but its

plan was interrupted and diluted. SLIM informs GAC 4 to constrain itself at its cell 7 move (the second plan step) with the threatening pattern 258. GAC 3 forbade the move to cell 9. GAC 2 proposed the move anyway. SLIM marks GAC 2 as an "idiot". (Etymologically, the term's Greek meaning refers to someone who only knows or cares about his own affairs.) SLIM informs GAC 2 that it should constrain itself by subordination to GAC 3, which forbid the move. The implication is that at its second move, GAC 2 should request constraints from GAC 3 and apply them in its own proposal process.

The significance of this procedure is in specifying a particular set of experiences which justify the judgment that GAC 2 is sufficiently similar to GAC 3 that it SHOULD use analogy to determine the vulnerability of its own plan. The first consequence isimproved play. A second is that should GAC 3 constrain itself by some other threatening pattern – for example the perpendicular threat 456, GAC 2 would inherit that protection as well. This subordination creates an interrelationship among structures learned through experience without any specific control structure being imposed at the time of the generation of the structure.

Conclusions and Frontier Problems

Our analysis of a part of the space of strategies exemplifies the complexity that arises from experiential and interactive learning. The strategies and the learning mechanisms are simple, but the simulations show the surprising power that they can exhibit.

A modest extension of this model leads to goal redescription through application of plan-constraints to goal specifications, calling upon a Sussmannesque inversion of a failure-description to generate a new type of term (the hole or empty-cell) for use in the goal description. Networks of descent describe sets of richly interrelated structures created through processes of reflexive concrete construction. We are now exploring whether two other types of reflexive processes, acting upon these structures as objects, will be adequate to recognize the applicability of symmetry in this domain and ultimately to create a complete classification of games adequate to exhibit domain mastery. The first process classifies objects by specifiable features; the second classifies objects by analysis of their function and genesis. Piaget (1971) calls these processes Aristotelian classification and reflective abstraction.

A different frontier represents the effort to explore how to structure the model so that the components exhibit the sort of functional lability that characterizes human intelligence and seems essential to begin development of a model of the play of the opponent.[8] Furthest from our current grasp is exploring the extent to which experience motivated analogy and other forms of developing control relationships permit the establishing through experience of coherent intereaction among a set of initially disparate structures. The extension of such a model as this to other domains will be worthwhile attempting when it successfully traverses its path to mastery. It is our expectation that the concrete representations developed will tend to be domain limited but that the processes for the analysis of concrete strategies will be less bound to particulars.

Acknowledgements

Thanks to Bud Frawley for creating the control structure which made these simulations possible. Our colleagues Golden, Cooperman, Fawcett, and Braunegg have provided friendly criticism and help. Marie McPartland-Conn made it possible to have a readible copy of this text.

References

Jacob, F., **The Possible and the Actual.**New York: Pantheon Books, 1982.

Lawler, R., **Computer Experience and Cognitive Development.** Chichester, England and New York: Ellis Horwood Ltd. and John Wiley, The Halstead Press, 1985.

Lawler, R. "The Internalization of External Processes," unpublished, 1985b.

Newell, A. and Simon, H. **Human Problem Solving** Englewood Cliffs, N.J.: Prentice Hall, 1972.

Piaget, J., **The Language and Thought of the Child.** New York: New American Library, 1974.

Piaget, J., **Biology and Knowledge** Chicago: University of Chicago Press, 1971.

Satinoff, E., "Neural Organization and Evolution of Thermal Regulation in Mammals," **Science 201,** pp. 15-22, 7 July 1978.

Selfridge, M. G. R. and Selfridge, O. G. How Children Learn to Count: a computer model. 1985.

[8] A discussion of this work appears in The Internalization of External Processes, Lawler 1985b.

Failure-Driven Acquisition of Figurative Phrases by Second Language Speakers*

Uri Zernik
Michael G. Dyer

Artificial Intelligence Laboratory
Computer Science Department
University of California
Los Angeles, CA 90024

Abstract

The problem of manually modifying the lexicon appears with any natural language processing computer program. Ideally, a program should be able to acquire new lexical entries from context, the way people learn. We address the problem of acquiring entire phrases, specifically figurative phrases, through augmenting a phrasal lexicon. We show that idiosyncratic behavior of certain phrases can be encoded in the lexicon only by modeling the learning process. We also show how metaphor mappings are acquired through parsing. The acquisition of novel figurative phrases, encountered in context, involves three problems: First, since all the constituents of the phrase are known, the existence of a novel phrase needs to be detected. Second, the scope and the generality of the linguistic pattern of the new phrase needs to be determined. Third, the meaning of the phrase needs to be extracted from the context. Our model is based on second language speakers' behavior and observation of their errors. We have designed and implemented a program called RINA which receives new figurative phrases in context and through the application of a sequence of failure-driven rules, creates and refines both the patterns and the concepts which hold syntactic and semantic information about phrases.

1. Introduction

The lexical approach to language processing [Becker75, Searle79, Bresnan82, Pawley83, Fillmore84] emphasizes the role of the lexicon as a knowledge source. Rather than maintaining a single "generic" lexical entry for each word e.g., `take`, the lexicon contains many phrases, e.g., `take on`, `take to the streets`, `take to winning`, `take over`, etc. This approach proves effective in parsing and in generation [Wilensky84]. However a huge lexicon must be acquired, which cannot be done manually, especially when considering subtle phrase meanings and idiosyncratic behavior of phrases. Moreover, we show in this paper that there are phrases whose behavior can be captured and encoded in the lexicon only by modeling the acquisition process. Therefore, for phrase-based programs to communicate effectively in natural language they must be able to augment their own lexicon, by simulating human learning. The program RINA [Zernik85] which is used as an experimental model, simulates a second language speaker learning new phrases.

Learning Figurative Phrases

Within language learning, our task domain is verb-phrase acquisition. The modeled phenomenon is described in the dialogues below. In the first dialogue, RINA is introduced to an unknown phrase: `take on`. The words `take` and `on` are familiar to RINA, who also remembers the biblical story of David and Goliath. RINA, modeling a language learner, interacts with a native speaker as follows:

David vs. Goliath

Native:	`Remember the story of David and Goliath? David took on Goliath.`
Learner:	`David took Goliath somewhere?`
Native:	`No. David took on Goliath.`
Learner:	`He took on him. He won the fight?`

* This work was made possible in part by a grant from the Keck Foundation.

Native:	No. He took him on. David attacked him.
Learner:	He took him on. He accepted the challenge?
Native:	Right.
Native:	Here is another story. John took on the third exam question.
Learner:	He took on a hard problem.

Another dialogue involves put one's foot down. Again, the phrase is unknown while its constituents are known:

Going Punk

Native:	Jenny wanted to go punk, but her father put his foot down.
Learner:	He moved his foot down? It does not make sense.
Native:	No. He put his foot down.
Learner:	He put his foot down. He refused to let her go punk.

A figurative phrase such as: put one's foot down is a linguistic pattern whose associated meaning cannot be produced from the composition of its constituents. Indeed, an interpretation of the phrase based on the meanings of its constituents often exists, but it carries a different meaning. The fact that this literal interpretation of the figurative phrase exists is a misleading clue in learning. Furthermore, the learner may not even notice that a novel phrase has been introduced since she is familiar with down as well as with foot. Following Becker [Becker75], we describe a space of phrases ranging in their generality from fixed proverbs: charity begins at home through idioms: lay down the law and phrasal verbs: put up with one's spouse, look up the name, to literal verb phrases such as: sit on the chair. He suggested employing a *phrasal lexicon* to capture this entire range of linguistic structures.

Issues in Phrase Acquisition

Four issues must be addressed when learning phrases in context.

(1) **Detecting failures**: What are the indications that the initial interpretation of the phrase: take him on as "to take a person to a location" is incorrect? Since all the words in the sentence are known, the problem is detected both as a *conceptual discrepancy* (why would he take his enemy anywhere?) and as a *syntactic failure* (the expected location of the assumed physical transfer is missing).

(2) **Determining scope and generality of patterns**: The linguistic pattern of a phrase may be perceived by the learner at various levels of *generality*. For example, in the second dialogue, incorrect generalizations could yield patterns accepting sentences such as:

> Her boss put his left foot down.
> He moved his foot down.
> They put down their feet.
> He put it.

A decision is also required about the *scope* of the pattern. For instance, the scope of the pattern in John put up with Mary could be (1) *?x:person put:verb up* where with is associated with Mary or (2) *?x:person put:verb up with ?y:person*, where with is associated with put up. This issue is described in greater detail in [Zernik85].

(3) **Finding appropriate meanings**: The conceptual meaning of the phrase must be extracted from the context which contains many concepts, both appropriate and inappropriate for hypothesis formation. Thus there must be strategies for focusing on appropriate elements in the context.

(4) **Example generation**: The learner must generate examples to convey her hypothesis about the phrase. Memory organization is required to allow (1) easy access to episodic examples and (2) limit the number of episodes accessible for each phrase.

2 Our Approach and Its Background

Past work in language learning emphasized either learning of linguistic patterns or learning of conceptual representations. There are three models for learning linguistic patterns:

- PST [Reeker76] operated by GPS principles [Newell57] and similarly used a table of *difference-action* pairs. PST learned grammar by acting upon differences between the input sentence and an internally generated sentence. Six types of differences were classified and the detection of a difference which belonged to a class caused the associated alteration of the grammar.
- LAS [Anderson77] learned ATNs (Augmented Transition Networks) from sample sentence/meaning pairs. LAS presented one element in a larger cognitive model which accounted for general human inference and memory access. This work intended to demonstrate that language learning could be modeled using general learning principles.
- AMBER [Langley82] modeled learning of basic sentence structure and function words. The learning process was directed by mismatches between input sentences and sentences generated by the program. Learning involved recovery from both *errors of omission* (omitting a function word such as `is` and `the` in `daddy bouncing ball`) and *errors of commission* (producing `daddy is liking dinner`). Like LAS, AMBER's main thrust was to apply general learning principles in language learning.

On the other hand, two models emphasized learning conceptualizations:

- FOUL-UP [Granger77] learned meanings of single unknown words from context. The meaning was extracted from the script [Schank77] which provided the context. A typical learning situation was `The car was driving on Hwy 66, when it swiveled off the road.` The unknown verb was guessed from the *$accident* script. FOUL-UP introduced three important elements: (1) Learning was invoked by parsing failures. However, there was only one possible failure—the absence of a word in the lexicon—thus failure analysis was not required. (2) Word meanings were figured out from currently active scripts. (3) Linguistic clues, such as preposition senses, took part in forming meanings.
- CHILD [Selfridge80] modeled a one-year old child learning native language. At that age, concepts rather than language word-order conventions account for comprehension. A sentence such as `Joshua, put the ball in the box` is understood from the conceptual relationships and conceptual clues. Thus CHILD was able to learn basic word meanings starting only with a minimal linguistic knowledge. CHILD introduced heuristics to identify the unknown word in a sentence and to identify the intended concept in a context. Learning was accomplished by associating the new word with that concept.

The integration of these two aspects is at the focus of our approach. For each new phrase, RINA acquires both the linguistic pattern and the conceptual meaning. It turns out that these two aspects are not independent: For example, learning the conceptual meaning of `take on` depends on knowledge of the words and their combination, while learning the linguistic pattern of that phrasal verb depends on the concepts involved.

Literal or Figurative, Dead or Alive

Gibbs [Gibbs84] has addressed the issue of figurative-phrase comprehension, arguing against Searle's notion of *literal interpretation*. Searle [Searle79] had proposed the existence of literal interpretation as a default comprehension mechanism. According to Searle, in comprehension of figurative phrases (speech acts such as that of `Can you please pass the salt?`), people try literal interpretation ("are you able to do that?") first and only when this attempt fails do they resort to the intended figurative interpretation ("pass it, please!"). Gibbs proved using experimental data that people spend no more time processing figurative phrases than they do processing analogous literal phrases. Thus, he claims, there is no extra effort in processing these phrases as implied by Searle. However, Gibbs does not propose any constructive model for the interpretation of figurative phrases. Moreover, he obviously concerns his discussion merely with the interpretation of *dead phrases*, namely *familiar* figurative phrases, while he does not explain how people interpret novel phrases—phrases they have not encountered before. Our contention is twofold. First, a uniform mechanism, the phrasal lexicon, accounts for comprehension of figura-

tive as well as literal phrases, provided that the phrase is already known. Second, an unknown phrase requires a special treatment which actually constitutes the learning of that new phrase.

Regularity and Idiomaticity in Phrases

Idiosyncratic behavior of phrases is difficult to capture in the lexicon. For example, read the next two sentences:

- `Peace was struck between Israel and Egypt. The hatchet was buried.`
- `Finally, death prevailed. The bucket was kicked.`

As opposed to the second phrase `kick the bucket`, the first phrase `bury the hatchet` can take the passive voice and still maintain its figurative meaning. What is the reason for this difference, and how can it be predicted for idiomatic phrases in general? Fillmore, Kay and O'Connor [Fillmore84] address the problematic behavior of idiomatic phrases, classifying them into categories according to the knowledge required for the understanding of each idiom. Our contention is that the only way to predict phrase behavior is by modeling the learning process.

Modeling Second Language Acquisition

Learning in general, and specifically learning linguistic knowledge, is an ongoing process. Two groups in particular experience extensive language learning: children learning native language and adults learning a second language [Hatch83, Gasser85, Ulm75]. Adults, as opposed to children, may augment their linguistic knowledge while, to a large extent, maintaining otherwise unchanging world knowledge. Three aspects of second language acquisition are investigated in our research:

(1) The various types of errors committed: acquiring incorrect concepts, acquiring incorrect linguistic patterns and performing incorrectly while having the correct linguistic knowledge.

(2) The processes underlying these errors (observing errors is the only way to expose these processes).

(3) Strategies for error-recovery based on failure-analysis.

The implications of this study, therefore, are not confined only to second language speakers. Rather, observing second language speakers may reveal general learning processes which are used more frequently by second language speakers.

3. The Program

RINA is a computer program designed to learn English phrases. It takes as input English sentences which may include unknown phrases and conveys as output its hypotheses about novel phrases. The program consists of four components:

(1) **Phrasal lexicon**: This is a list of phrases where each phrase is a declarative *pattern-concept* pair [Wilensky84].

(2) **Case-frame parser**: In the parsing process, case-frame expectations are handled by spawning *demons* [Dyer83]. The parser detects comprehension failures which are used in learning.

(3) **Pattern Constructor**: Learning of phrase patterns is accomplished by analyzing parsing failures. Each failure situation is associated with a pattern-modification action.

(4) **Concept Constructor**: Learning of phrase concepts is accomplished by a set of strategies which are selected according to the context.

Schematically, the program receives a sequence of *sentence/context* pairs from which it refines its current *pattern/concept* pair. The pattern is derived from the sentence and the concept is derived from the context. However, the two processes are not independent since the context influences construction of patterns while linguistic clues in the sentence influence formation of concepts.

Phrasal Representation of the Lexicon

RINA uses a declarative phrasal lexicon suggested by Wilensky, where a lexical phrase is a *pattern-concept* pair. RINA's patterns are similar to those in [Arens82]. The notation is explained by three example patterns:

```
P1:   ?x:animate nibble:verb <on ?y:food>
P2:   ?x:person take:verb on ?y:person
P3:   ?x:person <put:verb foot:body-part down>
```

Figure 1: **The Pattern Notation**

(1) A *token* is a literal unless otherwise specified. For example, **on** is a literal in the patterns above.

(2) **?x:***sort* denotes a variable called **?x** of a *semantic* class *sort*. **?y:food** above is a variable which stands for references to objects of the *semantic* class **food**.

(3) *Act*:**verb** denotes any form of the **verb** *syntactic* class with the root *act*. **nibble:***verb* above stands for expressions such as: `nibbled, did not nibble, will never be nibbled.`

(4) By default, a pattern sequence does not specify the order of its tokens*. However, based on general English knowledge also provided as patterns, it is the *actor* which is expected to precede the verb in the active form of a sentence.

(5) Tokens delimited by **<** and **>** are restricted to their specified order. In P1 above, **on** must directly precede ?y:food.

Each pattern has an associated meaning. Meaning representations are not discussed here; they are specified using Dyer's [Dyer83] *i-link* notation which defines a set of intentional links connecting primitive actions, plans, and goals [Schank77].

Case-Frame Parser

Three tasks in phrasal parsing are identified, ordered by degree of difficulty:

(1) **Phrase disambiguation**: When more than one lexical phrase matches the input sentence, the phrase intended by the speaker must be selected by the parser. For example, `John took to the streets` could mean: "he led a criminal life", "he demonstrated" or "he was fond of the streets".

(2) **Ill-formed input comprehension**: Even when an input sentence is not well phrased according to textbook grammar, it may be comprehensible by people and so must be comprehensible to the parser. For example, `John took Mary school` is somehow telegraphic, but comprehensible, while `John took Mary to` conveys only a partial concept.

(3) **Error Detection**: when the hypothesized phrase does not match the input sentence/context pair, the parser is required to detect the failure and return with an indication of its nature which is geared to the construction of a more accurate hypothesis.

The key element in accomplishing these tasks is the use of case frames for pattern representation, as elaborated in [Zernik85].

* In order to derive phrases with a definite word order, *lexical patterns* must interact with *ordering patterns* [Arens82] which hold general English word-order conventions.

Failure-Driven Pattern Constructor

Learning of phrases is an iterative process. The input is a sequence of sentence-context pairs, through which the program refines its current hypothesis about the new phrase. The hypothesis pertains to both the pattern and the concept of the phrase. The basic cycle in the process is:

(a) A sentence is parsed on the background of a conceptual context.

(b) Using the current hypothesis, either the sentence is comprehended smoothly, or a failure is detected.

(c) The analysis of a failure directs the update of the current hypothesis.

The crucial point in this scheme is to obtain from the parser an intelligible analysis of the failure. As an example, consider this part of the first dialog:

1	Program:	`He took on him. He won the fight?`
2	User:	`No. He took him on. David attacked him.`
3	Program:	`He took him on. He accepted the challenge?`

The first hypothesis is shown in Figure 4.

pattern:	?x:person take:verb <on ?y:person>
concept:	?x win the conflict with ?y

Figure 4: **First Hypothesis**

Notice that the preposition `on` is attached to the object ?y, thus assuming that the phrase is similar to `He looked at Mary` which cannot produce the following sentence: `He looked her at`. This hypothesis underlies Sentence 1 which is erroneous in both its form and its meaning. Two observations should be made by comparing this pattern to Sentence 2:

- The object is not preceded by the preposition `on`.
- The preposition `on` does not precede any object.

These comments direct the construction of the new hypothesis:

pattern:	?x:person take:verb on ?y:person
concept:	?x win the conflict with ?y

Figure 5: **Second Hypothesis**

where the preposition `on` is taken as a modifier of the verb itself, thus correctly generating Sentence 3. In Figure 5 the conceptual hypothesis is still incorrect and must itself be modified.

4. Strategies for Learning Concepts

In the first dialog (Section 1.1), the context has been presented by the biblical story of *David and Goliath*. RINA comes up with the wrong hypothesis, assuming that `take on` means "to win a fight"*. What is the process underlying this error? We explain such errors by the theory of *story points* [Wilensky82] and salient expectations. Story points encapsulate our impression of the story beyond the level of mundane details (e.g., David's hair color, the weapons involved, the various moves in the fight) which get forgotten with time. When the learner is required to select a conceptualization from the context, she attempts to use the dominant story point. Since "David won the fight in spite of his physical inferiority" is the salient point of the story, this point serves as the learner's first guess.

* This is a real error recorded by the authors from second language speakers.

When the context is given as a story, RINA uses story points to construct the meanings of unknown phrases. For other contexts, where a variety of knowledge structures are dominant we have developed a set of strategies for extraction of meanings from context. Interestingly, all these strategies demonstrate the integration of learning and comprehension. Each learning strategy presented here is derived from an existing comprehension strategy.

Story Points

In the context of *David and Goliath*, RINA selects the concepts from a set of given story points. Initially, the first story point ("David won the fight in spite of his physical inferiority") is selected. The factors in the selection of this point are:

- Prefer a story point which determines the outcome of the situation.
- Use linguistic clues. For instance, the preposition `on` carries the sense of winning (in `turn on`, `put on`, `hang on`, etc., `on` depicts a "positive" state). It matches the outcome of the fight.

Later in the dialogue, when the native speaker negates this interpretation by stating `he attacked him`, the learner reverts to the second story point (`he accepted the challenge`) which depicts the correct meaning.

Script-Based Expectations

In a way similar to FOUL-UP [Granger77], RINA is able to associate the new phrase with script-based expectation. For example consider the following text: `After a long illness, his beloved wife finally kicked the bucket.` The first sentence invokes $*disease* (the disease script) which incorporates a chain of events such as feeling sick, staying in bed, seeing a doctor and eventually recovering or dying. From linguistic clues (e.g., `finally`) and from the causal clue (`a long disease`) RINA selects the last event in the script as the meaning to refer to. It is interesting to notice what triggered the learning process: The failure in the literal interpretation of `the bucket` suggests the existence of an unknown idiomatic phrase. There is a similarity between this method and general expectation-based parsing [Riesbeck74]. In parsing, outstanding expectations are used in disambiguation of meanings. In learning, outstanding expectations are taken as conceptual meanings for novel phrases.

Goal/Plan-Based Expectations

An expectation at the plan/goal level dominates the following text (in Section 1.1, *Going Punk*): `Jenny wanted to go punk, but her father put his foot down.` The context describes a goal conflict between Jenny and her father. Jenny is expected to implement a certain goal (satisfy-vanity) when the word `but` is encountered, indicating that the implementation of the goal is blocked. The act denoted by the phrase is expected to cause this goal to become blocked. However, the metaphor has not yet been resolved.

Metaphor Mapping

A *metaphor* [Carbonell83] is defined as a *mapping* between *patterns* such as `put your foot down`, `shoot one's foot`, `put one's foot in one's mouth`, `climb the walls`, etc., and specific *episodes* in memory. In parsing an already familiar metaphor, the lexicon provides the mapping from the elements of the pattern to the elements of the episode. However, while the mapping is still unknown, the episode cannot be accessed from the sentence itself since "putting one's foot on the floor" does not necessarily convey resistance. The episode may be accessed only by indirect memory search through links from affects and goal/plan situations. For example, the process in *Going Punk* proceeds as follows: Since the act (moving one's foot down) does not conform with the goal/plan situation, and the reference (`foot`) cannot be resolved in the context, a search for a metaphor is initiated. A link is found from the goal-conflict situation and the resistance stance to the episode of stamping one's foot. The phrase is interpreted successfully in that episode, and as a result, the mapping between the pattern and the episode is established in the lexicon.

Generalising Word Meanings

In idioms such as: `They buried the hatchet`, `He threw the book at him`, and `He laid down the law`, the meaning is constructed by generalizing single words' meanings. `After a long dispute, the couple buried the hatchet.` The literal interpretation of the action does not make sense (burying some tool in the ground), and no episode is found to constitute a metaphor. Therefore, a sequence of generalizations is initiated:

hatchet –> weapon –> fight-plan
bury –> disenable use –> cease-a-plan

Through these generalizations, the act is realized as the cease-a-plan of the fight-plan which finally relates to the context. This process is triggered by sensing that the word `hatchet` is *too specific* (according to Rosch's [Rosch78] *basic-level* principle of categorization). Such a word is expected to appear in in a specific script, otherwise it suggests a figurative use.

5. Connotations

RINA's lexicon indexes episodic as well as generic elements. Indexed episodes which facilitate example-generation are built up in the learning process. RINA generates sentences which exemplify the correct use (or the current hypothesis) of newly acquired phrases. For instance, RINA could generate: `The Lakers took on The Celtics`, using a familiar episode as an example of the phrase `take on`. Example generation is the preferred way for people to discuss phrases (rather than speak in terms of syntax and semantics). Upon request, a person may easily generate a few examples of correct phrase use:

```
David took on Goliath.
My brother took on a new job.
The Lakers took on The Celtics.
```

relating to specific episodes. However, beyond a limited set of examples, generation of additional examples becomes more difficult. Computationally, example generation seems to be a complex task. In a database system, example generation for a phrase would require scanning the entire database for an appropriate situation which satisfies the constraints imposed by the meaning of the phrase.

Connotations Indexed by Phrases

RINA simulates this phenomenon in terms of memory organization. A phrase in the lexicon is a *pattern-concept* pair. However, in addition, a phrase accumulates links to episodes in which RINA has encountered the phrase. When required, RINA generates an example by selecting an episode which is linked to that phrase. How are links to episodes created? If on each encounter with the phrase `take on`, an episode were linked to the phrase, then the linkage system would grow out of bounds. Therefore, a new link is created only in situations where a learning effort is required. Thus only one episode of a kind is linked, avoiding redundant links. An illustrative learning scenario for `take on` is:

Native: `My brother took on a new job.`
Learner: <link phrase to episode career-challenge1>

(The learner is not familiar with the phrase. She learns the phrase, assuming that it means to start a fight and she links it to the episode.)

Native: `The Lakers took on The Celtics.`
Learner: <link phrase to episode sporting-event1>

(Again a new sense of the phrase is learned while the episode is linked.)

Native: `The Redskins took on The Whiteskins.`
Learner: <no linking>

(This time there is no need to learn a new sense. The sentence was comprehended using existing phrases.) Following this scenario, the learner can generate only two examples for `take on`, which are *career-challenge1* and *sporting-event1*. The third episode is not indexed to the phrase.

6. Conclusions

The issue addressed in this paper is the construction of the phrasal lexicon. As opposed to other systems (PHRAN [Arens82] and PHRED [Jacobs85], for example) where the lexicon is constructed manually, in RINA new phrases are acquired through learning. This is significant not only for robustness and flexibility, but also for the correct encoding of lexical phrases. For example, observe phrases such as `bury the hatchet` and `kick the bucket`. On the outset, it is difficult to determine that the first phrase takes the passive voice while the second one does not. (How can the systems programmer, who sets up the lexicon, predict the behavior of other such idiomatic phrases?) Moreover, how can such information be encoded in the lexicon? One way is to mark the phrase `kick the bucket` by the *syntactic* feature: **does not take the passive voice.** Our contention is that such a feature is *conceptual* rather than syntactic and that word-order restrictions actually reflect phrase concepts. The methodology for acquiring this feature is through learning: The first phrase is acquired through word-sense generalization (see Section 4), where the entire phrase has an associated concept, but each individual word also stands for a certain concept (specifically, `hatchet` stands for "war"). Thus, the passive voice serves a communicative discourse function. However, in the second phrase which is acquired as a whole unit, the single words do not stand for concepts (on the contrary, the phrase was actually acquired by noticing that neither `bucket` nor `kick` could be interpreted in the context). Not only does the passive voice serve no function, but it is also misleading, causing the listener to interpret the words literally, and search for their individual meanings.

We have shown in this paper a computational method for metaphor resolution. A metaphor is defined as a mapping from a pattern to an episode. Familiar metaphors are comprehended through the mappings themselves which are provided by the phrasal lexicon. A new mapping is constructed when the phrase is heard in context. The successful construction depends on: (1) the accessibility of a metaphor episode through salient elements in the context, and (2) the interpretation of the phrase in the episode which establishes the mapping of pattern elements to episode concepts.

7. Future Work

We have shown the steps in learning new phrases: a novel phrase is detected, and its pattern is shaped by the given sentences: detected comprehension failures cause pattern modifications. The concept of the phrase is formed by (a) the given context, (b) linguistic clues, (c) generalized word meanings, and (d) by metaphor mappings. In the future we intend to investigate more concept-forming strategies and focus on generalization of concepts. For example, two different episodes are associated with `take on`:

`David took on Goliath.`

`He took on a new job.`

The associated concepts extracted from the corresponding contexts are:

"deciding to fight"

"undertaking a responsibility"

The dilemma a learner is facing is whether there is a general common concept which encompasses these two concepts: do they share one lexical entry or are they two separate entries? In this task, similar to learning linguistic patterns, people make errors by overgeneralizing and by undergeneralizing. Therefore, analysis of erroneous hypotheses both for linguistic patterns and for conceptual meanings, accounts for learning.

References

[Anderson77] Anderson, J. R., "Induction of Augmented Transition Networks," *Cognitive Science* **1**, pp.125-157 (1977).

[Arens82] Arens, Y., "The Context Model: Language Understanding in a Context," in *Proceedings 4th Annual Conference of the Cognitive Science Society*, Ann-Arbor MI (1982).

[Becker75] Becker, J., "The Phrasal Lexicon," in *Proceedings Interdisciplinary Workshop on Theoretical Issues in Natural Language Processing*, Cambridge MA (1975).

[Bresnan82] Bresnan, J. and R. Kaplan, "Lexical-Functional Grammar," in *The Mental Representation of Grammatical Relations*, ed. J. Bresnan, MIT Press, MA (1982).

[Carbonell83] Carbonell, J. and S. Minton, "Metaphor and Common-Sense Reasoning," cmu-cs-83-110, Carnegie Mellon University, Pittsburgh PA (1983).

[Dyer83] Dyer, Michael G., *In-Depth Understanding: A Computer Model of Integrated Processing for Narrative Comprehension*, MIT Press, Cambridge, MA (1983).

[Fillmore84] Fillmore, C., P. Kay, and M. O'Connor, "Regularity and Idiomaticity in Grammar," Cognitive Science Working Paper, University of California (1984).

[Gasser85] Gasser, M., "Second Language Production: Coping with Gaps in Linguistics Knowledge," UCLA-AI-15, LA CA (July 1985).

[Gibbs84] Gibbs, R., "Literal meaning and psychological theory," *Cognitive Science* **8**(3) (1984).

[Granger77] Granger, R., "FOUL-UP: A Program That Figures Out Meanings of Words from Context," in *Proceedings Fifth IJCAI*, Cambridge, Massachusets (1977).

[Hatch83] Hatch, E. M., *Psycholinguistics: A Second Language Perspective*, Newbury House, Rowley MA (1983).

[Jacobs85] Jacobs, Paul S., "PHRED: A Generator for Natural Language Interfaces," UCB/CSD 85/198, University of California Berkeley, Berkeley CA (Jan 1985).

[Langley82] Langley, P., "Language Acquisition Through Error Recovery," *Cognition and Brain Theory* **5**(3) (1982).

[Newell57] Newell, A., C. Shaw, and A. Simon, "Preliminary Description of GPS-I," CIP Working Paper 7, Carnegie Institute of Technology, Pittsburgh, PA (1957).

[Pawley83] Pawley, A. and H. Syder, "Two Puzzles for Linguistic Theory: Nativelike Selection and Nativelike Fluency," in *Language and Communication*, ed. J. C. Richards R. W. Schmidt, Longman, London (1983).

[Reeker76] Reeker, L. H., "The Computational Study of Language Learning," in *Advances in Computers*, ed. M. Yovits M. Rubinoff, Academic Press, New York (1976).

[Riesbeck74] Riesbeck, C., "Analysis of Sentences and Context," AI-238, Yale (1974).

[Rosch78] Rosch, E., "Principles of Categorization," in *Cognition and Categorization*, ed. B. Lloyd, LEA (1978).

[Schank77] Schank, R. and R. Abelson, *Scripts, Plans, Goals, and Understanding*, LEA, Hillsdale, NJ (1977).

[Searle79] Searle, J., "Speech Acts and Recent Linguistics," in *Expression and Meaning*, ed. J. Searle, Cambridge University Press, Cambridge (1979).

[Selfridge80] Selfridge, M., "A Process Model of Language Acquisition," 172, Yale, New Haven CT (1980). Ph.D. Dissertation.

[Ulm75] Ulm, S., *The Separation Phenomenon in English Phrasal Verbs, Double trouble*, UCLA (1975). M.A. Thesis.

[Wilensky82] Wilensky, R., "Points: A Theory of Structure of Stories in Memory," pp. 345-375 in *Strategies for Natural Language Processing*, ed. W. G. Lehnert M. H. Ringle, Laurence Erlbaum Associates, New Jersey (1982).

[Wilensky84] Wilensky, R., Y. Arens, and D. Chin, "Talking to UNIX in English," *CACM* **27**(6) (1984).

[Zernik85] Zernik, U. and M. G. Dyer, "Towards a Self-Extending Phrasal Lexicon," in *Proceedings 23rd Annual Meeting of the Association for Computational Linguistics*, Chicago, Ill. (July 1985).

TWO KINDS OF FEATURE? A TEST OF TWO THEORIES OF TYPICALITY EFFECTS IN NATURAL LANGUAGE CATEGORIES

Robin A. Barr & Leslie J. Caplan,
Department of Psychological Science, Ball State University, Muncie, IN 47306

The authors thank the teachers and students of Burris Laboratory School, Muncie, Indiana for their cooperation. Experiment 1 was supported by a Ball State University research grant.

In two experiments, we tested predictions of a model which states that natural language categories are represented primarily by intrinsic features (features true of an exemplar in isolation) or by extrinsic features (features true of an exemplar interacting with some other entity. We hypothesized that categories whose intensions consisted primarily of extrinsic features would have "fuzzier" extensions than those whose intensions consisted primarily of intrinsic features. This hypothesis was supported by the results of Experiment 1. In Experiment 2, we demonstrated that the features which represent a category are equally descriptive of it, regardless of whether the representation is primarily intrinsic or extrinsic - a finding inconsistent with a probabilistic theory's account of the results of Experiment 1. We argue that investigation of the properties of different kinds of feature is an appropriate focus for future research on natural-language categorization.

Typicality effects in studies of natural language category representation have frequently been interpreted as evidence that categories are "fuzzy" (e.g., Rosch, 1973, 1975). Recently, however, other investigators have questioned whether the existence of typicality effects does, in fact, imply that categories are necessarily fuzzy (e.g., Armstrong, Gleitman, & Gleitman, 1983; Barsalou, 1982; Osherson & Smith, 1981).

Our purpose here is to show that some typicality effects are caused by a particular kind of category intension, rather than by the fuzziness of the category per se. We previously (Barr and Caplan, note 1) have distinguished between intrinsic and extrinsic features in describing the intensions of categories. Intrinsic features are those which are true of a category member considered in isolation. For example, an intrinsic feature of "dog" is "has fur". Extrinsic features are true only when the category member interacts with some other entity. For example, an extrinsic feature of "clothing" is "covers people".

In that earlier paper we found that most of the features chosen by subjects to define artifactual categories (e.g. "tools" and "furniture") were extrinsic. Conversely, most of the features chosen to define naturally-occurring categories (e.g. "fruit" and "trees") were intrinsic. We

also found that subjects agreed with each other just as much in choosing defining features for artifactual categories as in choosing features for naturally-occurring categories. However, they agreed less when generating exemplars of artifactual categories than when generating exemplars of naturally-occurring categories. In other words, although the intensions of the two types of category were equally clear, the extensions of artifactual categories were "fuzzier" than those of naturally-occurring categories.

We argued that extrinsic features cause fuzzy extensions while intrinsic features result in clearer extensions. Because intrinsic features are always true of a category member, it makes little sense to apply a qualifier when deciding whether an exemplar possesses a particular feature (e.g., "Sparrows are sometimes hatched from eggs"). Either the exemplar possesses the feature, or it does not. On the other hand, it does make sense to apply qualifiers when making decisions about extrinsic features (e.g., "A sled is sometimes used for transportation"). Borderline category members will be those which can be considered a member only with the addition of strong qualifiers (e.g. "Rubber bands are SOMETIMES used to hurt people"). Therefore, categories which are represented primarily by extrinsic features should have fuzzier extensions than those represented primarily by intrinsic features.

Intrinsically represented categories will, however, still yield some effect of typicality on membership judgments. Several different factors will contribute to the size of a typicality effect. For example, as some investigators (e.g., Armstrong, Gleitman & Gleitman, 1983; Osherson & Smith, 1981) have suggested, subjects may well use an "identification function" in typicality experiments. However, because intrinsic features are true of exemplars without qualification, the effect of typicality should be smaller in intrinsically represented than in extrinsically represented categories.

We argued in Barr and Caplan (note 1) that, in practice, many categories will be represented by a mixture of extrinsic and intrinsic features. Accordingly, the relative importance of extrinsic and intrinsic features to the representation of the category will determine how fuzzy the extension becomes. Our data implied that the artifactual categories we used in these experiments were represented more by extrinsic features than were the naturally-occurring categories that we used. Therefore, in Experiment 1, we tested the prediction that the rated typicality of an item will yield larger effects on membership judgments for artifactual categories than on membership judgments for naturally-occurring categories.

It also follows that the extensions of intrinsically represented categories should be easier to learn than those of extrinsically represented categories.To learn intrinsically represented categories, children need only attend to the exemplar itself when learning the category label. However, to learn extrinsically represented categories, children need to attend both to the exemplar and to the context in which the exemplar occurs. Atypical exemplars of extrinsically represented categories will rarely be encountered in contexts consistent with the category's features In the first experiment, we therefore tested the hypothesis that school-age children's extensions of artifactual categories would be less adult-like than their extensions of naturally-occurring categories. In particular, we predicted that children's category-membership judgments will be least adult-like for atypical, artifactual exemplars.

Experiment 1

The artifactual categories we chose were "furniture" and "clothing". In our previous paper, we asked three independent judges to rate the five most popular features defining a category (as chosen by our subjects) as either intrinsic or extrinsic. According to these judgments, three of the top five features for "furniture" were extrinsic, and four of the top five features for "clothing" were extrinsic. The remaining features for both categories were judged to be intrinsic. The naturally-occurring categories used in this experiment were "birds" and "fruit". According to our independent judges, four of the top five features for "birds" were judged intrinsic, and three of the top five features for "fruit" were judged intrinsic. The remaining features for both "fruit" and "birds" were judged to be extrinsic.

We asked subjects in this experiment to decide whether exemplars which differed in category goodness-of-example ratings according to Rosch (1975) were members of categories. We predicted that the differences in RT and "no" judgments for "true" sentences between naturally-occurring and artifactual categories would be greater for atypical than for typical instances of the categories.

Method

Subjects. Subjects were 28 college students, some of whom received course credit for participation, and 33 school-age children. Seventeen of the children were from combined second- and third-grade classes (mean age = 8 years, 7 months), thirteeen of them were from the fourth grade (mean age = 10 years, 2 months), and nine were sixth graders (mean age = 11 years, 9 months). All of the children were students at Burris School, a public laboratory school run by Ball State University which serves grades kindergarten through twelve. The adult volunteers were all undergraduates at the same university.

Stimuli. The stimuli for the practice trials were constructed from two sets of three exemplars from the categories "animals" and "tools". They were presented in sentences of the form "A hammer is a tool". Each exemplar was presented twice, once in a true sentence, and once in a false sentence (i.e., a sentence using the exemplar from one of the two categories, paired with the other superordinate), yielding a total of twelve sentences.

Stimuli for the main part of the experiment were constructed from sets of ten exemplars drawn from each of the naturally-occurring categories "fruit" and "bird" and from each of the artifactual categories "furniture" and "clothing". As in the practice trials, each stimulus was a sentence of the form "Apples are fruit". For each category, two exemplars were selected at each of five different levels of goodness-of-example (Rosch, 1975). As far as possible, mean typicality ratings for each level were equated across the different categories. However, as only one "bird" ("bat") has a mean goodness-of-example rating above 5.00 in the Rosch data, it was not possible to completely control for differences in these numerical ratings. In addition, the mean number of letters in exemplar words from all categories was equated as much as possible across typicality levels.

As in the practice trials, each of the exemplars used in the experimental trials was presented once in a true sentence, and once in a false sentence. To construct the false sentences, bird exemplars were paired with the superordinate "fruit", furniture exemplars were paired with the superordinate "birds", fruit exemplars were paired with the superordinate "clothes", and

clothing exemplars were paired with the superordinate "furniture". The resulting 80 sentences (40 true and 40 false sentences) were divided into two trial blocks of 40 sentences each. Each trial block included 20 true sentences, one at each level of typicality for each of the four categories. The remaining 20 false sentences were divided equally among the four categories used.

Apparatus. Stimuli were presented using a three-channel tachistoscope. Reaction times were recorded by a voice-activated relay and timer system.

Procedure. Each subject was tested individually. At the beginning of each session, children were tested for their ability to read the exemplars and superordinates to be used in practice and experimental trials, by reading these words aloud when they were presented on flashcards. If a child was unable to read a word, the child was corrected and that word was later re-presented. Words read correctly were not re-presented. This procedure was repeated until the child had read each word aloud correctly once. This procedure was not used with adult subjects.

Subjects were instructed that they were going to see a series of sentences, and that their task was to decide whether each sentence was true or false. If they decided the sentence was false, they were to say "false" aloud. If they decided it was true, they were to say "true" aloud. All subjects were instructed to respond as quickly and accurately as possible. They were also instructed to fixate a fixation point at the beginning of each trial.

Practice trials were then presented, followed by the experimental trials. Each sentence was preceded by a fixation point for 1.5 seconds, and each sentence was presented for 4 seconds. The order in which the two trial blocks were presented was counterbalanced. Reaction times were recorded from the onset of the stimulus sentence, for experimental trials only.

Results

Despite pretraining in the reading of stimulus words, some children were still unable to read tachistoscopically presented stimulus sentences. Therefore, data from four of the second- and third graders and from one fourth grader were not included in the following analyses.

We tested the following predictions: 1) that the number of "no" responses would be greater for artifactual than for naturally-occurring categories, especially for atypical exemplars, 2) that subjects would be slower to verify atypical artifactual exemplars than to verify atypical naturally-occurring exemplars, 3) that children's judgments would differ from adults' most for atypical artifactual category exemplars, and 4) that the difference in RT between children and adults would be greatest for atypical artifactual exemplars. A summary of the data used in the analyses reported below is presented in Table 1.

"No" judgment analyses. A three-way mixed design analysis of variance was conducted on the number of "no" judgments made for "true" sentences, with category type (artifactual vs. naturally-occurring) and typicality level (levels one through five) as within-subject variables, and age (children vs. adults) as the between-subject variable. For both this analysis, and the one presented below, children were not divided into age groups because previous analyses had failed to reveal any interaction between age and the other variables of interest when the data from children were considered alone.

More "no" judgments were made for artifactual categories than for

naturally-occurring categories, $F(1, 57) = 49.70$, $p < .0001$. These judgments also increased as typicality decreased, $F(4, 228) = 71.84$, $p < .0001$. However, an interaction between category type and typicality level was also obtained, $F(4, 228) = 29.74$, $p < .0001$. Subjects made more "no" judgments for atypical artifactual items than for atypical naturally-occurring items, although the number of "no" judgments was similar for typical artifactual and naturally-occurring categories. All remaining main effects and interactions failed to reach significance at the $p < .05$ level.

The predicted three-way interaction between age, category type, and typicality level was clearly not significant in the above analysis. However, our prediction originally involved atypical items. Therefore, in a second analysis, we investigated the effects of age, category type, and typicality level only for the three least typical conditions. In this analysis, the three-way interaction was significant, $F(2, 118) = 3.11$, $p < .0485$. Children's and adults' judgments were similar for naturally-occurring categories. For artifactual categories, children made more "no" judgments than adults for medium typicality items, and fewer "no" judgments than adults for the most atypical items. As in the previous analysis, there were significant effects of category type ($F(1, 59) = 63.14$, $p < .0001$), and of typicality level ($F(2, 118) = 64.58$, $p < .0001$), and an interaction between category type and typicality ($F(2, 118) = 42.40$, $p < .0001$).

Reaction time analyses. A three-way mixed design analysis of variance was conducted on individual subjects' mean reaction times, using the same design as that of the first analysis reported above. Subjects took longer to verify statements about artifactual categories than naturally-occuring categories, $F(1, 43) = 25.48$, $p < .0001$. Reaction times also increased as typicality decreased, $F(4, 172) = 17.40$, $p < .0001$. Children showed a greater increase in RT for atypical items than did adults, as reflected in an interaction between age and typicality, $F(4, 172) = 2.92$, $p = .0229$. The difference in RT between naturally-occurring and artifactual categories was marginally greater for children than for adults, $F(1, 43) = 3.63$, $p = .0634$. The predicted interaction between category type and typicality level was not obtained. Children's RT's were longer than those of adults, $F(1, 43) = 67.59$, $p < .0001$. All remaining effects failed to reach significance at the $p < .05$ level.

Finally, as in analyses of "no" judgments, we investigated the effects of age, category type, and typicality level on RT, using only the three least typical conditions. Once again, the effects of category type ($F(1, 43) = 8.45$, $p = .0058$, and age ($F(1, 43 = 61.26$, $p < .0001$) were significant. The main effect of typicality was no longer significant at the $p < .05$ level. All remaining effects failed to reach significance at the $p < .05$ level.

Discussion

The results of this study are consistent with those of our earlier work (Barr & Caplan, note 1). Once again, there was a clear difference between the the nature of the extensions of artifactual and naturally-occurring categories. Reaction times and the number of "no" judgments were higher for artifactual than for naturally-occurring categories. In addition, artifactual categories demonstrated a more pronounced effect of typicality in "no" judgments than did naturally-occurring categories. Finally, children's RT's

and judgments suggest that they have more difficulty learning the extensions of artifactual categories than of naturally-occurring categories.

Our original predictions were similar to the results obtained. However, we had expected to find an interaction between category type and typicality level in the RT data, which we did not find. Instead, judgments for artifactual categories were slower than those for naturally-occurring categories at all levels of typicality.

How might this result be explained? We had expected this interaction because of the nature of extrinsic features. Presumably, atypical artifactual items would have forced subjects to engage in a lengthy search for contexts appropriate to the category's features. This search would increase the reaction times for atypical artifactual exemplars. However, the difference between artifactual and naturally-occurring categories was the same, regardless of the exemplars being considered. This suggests that the difference between the two types of category is directly related to the features associated with the category label, not to those of the exemplar.

When a category is represented by intrinsic features, it usually also shares many extrinsic features with other members of the category. For example, if a category member has wings and feathers, it is very likely to fly. On the other hand, when a category is extrinsically represented, it is less likely to share intrinsic features with other members of the category. For example, if a category member is used to enhance house decor, it can take many shapes, sizes, etc. Therefore, when a subject tries to generate the features of an intrinsically represented category, either the intrinsic or extrinsic features will be useful in judging whether an item belongs to a category. However, when he tries to generate the features of an extrinsically represented category, any intrinsic features he may generate will not be useful. He must instead search for extrinsic features. Because this process will take time, and is independent of the typicality level of the exemplar involved, category membership judgments will be longer for artifactual than for naturally-occurring categories.

Although this explanation can easily account for our results, there is an alternative explanation. Some theories of category representation imply that features are only probabilistically associated with a category (e.g. Rosch, 1975). Membership is determined by some kind of weighted sum of these probabilistic features. Presumably, borderline instances have a lower overall sum than more typical instances. One might hypothesize that the features of artifactual categories are associated to the category with a lower probability than are those of naturally-occurring categories. This, in turn, would yield results similar to those we obtained. In the next experiment, therefore, we tested between these alternative explanations of our results.

Experiment 2

In this experiment we presented to subjects a list of features which were previously selected as defining of a category (Barr & Caplan, note 1). The subjects' task was to name the category described by the features.

If the probabilistic account of the results of Experiment 1 is correct, subjects should be less likely to name the artifactual categories described by the features than they are to name the natural categories. After all, according to the probabilistic account, the features of the artifactual categories we used should have a lower overall association to the category

label than the features of the natural categories that we used.

On the other hand, the intrinsic/extrinsic account explains the results of Experiment 1 by pointing to the kinds of features associated with the category, rather than to their relative probabilities. This account, therefore, predicts no difference in subjects' ability to name the artifactual and natural categories described by the features.

Method

Subjects. The subjects were 44 undergraduate volunteers at Ball State University. Some of the volunteers received course credit for participation.

Stimuli. The stimuli were sets of ten features written on pages of a small booklet, one feature to a page. The features were chosen from sets of features selected as defining by subjects in a previous study (Barr and Caplan, note 1; Experiment 1). The features used were the ten most popular features for each category in the earlier experiment. Seven artifactual categories (furniture, clothing, tools, weapons, vehicles, toys and sports) and seven naturally-occurring categories (fruits, birds, vegetables, trees, mammals, flowers and metals) were used. The order in which features were presented to subjects was determined by a Latin square.

Procedure. Subjects first completed the consent form. They were then told that they would receive 14 booklets, each containing 10 features. The features in each booklet described one particular category. They were instructed to go through the booklets, page by page. On each page they were to write down their idea of what was being described by the features. Subjects were encouraged to guess if they had no particular idea as to the category and to write down a response on each page before turning to the next page. They were permitted to look back over features on earlier pages, but they were not permitted to look forward beyond the current page. Finally they were encouraged to think of the broadest category (i.e., the highest level) which would fit the features. Each subject received the 14 booklets in a different random order.

Results

Because of space limitations, the results from only the four categories used in Experiment 1 will be reported. These were furniture, clothing, fruits and birds. In the results below we counted as a correct response either (1) the category label, or (2) the label of an exemplar of the category. For example, if a subject was working with the features of "furniture", his response would be scored as correct if he had responded with an exemplar such as "bed"

All 44 subjects correctly identified "birds" and 41 out of 44 subjects correctly identified "fruit". Among the artifactual categories, all 44 subjects correctly identified "clothing". Thirty-seven out of 44 correctly identified "furniture".

Subjects required a mean of 1.40 features before first identifying "bird", a mean of 3.24 features before first identifying the category "fruit", a mean of 4.43 features before identifying the category "furniture" and a mean of 1.67 features before identifying "clothing".

It was also possible to calculate which features proved to be most helpful to subjects in identifying the categories. In the summary below, we counted a feature as helpful if it allowed at least one-third of the subjects to identify the category correctly for the first time. The category "birds" had six such features (four were intrinsic and two were extrinsic). "Fruit"

had three such features (all three were intrinsic). "Clothing" had six such features (two were intrinsic and three were extrinsic). "Furniture" had just two such features (both were extrinsic).

Discussion

Clearly, nearly all subjects identified all four categories. There was little, if any, difference between the number of subjects who identified the naturally-occurring categories and the number who identified the artifactual categories. There were differences in the number of features needed before the subjects successfully identified the categories. But the differences did not appear related to the naturally-occurring versus artifactual distinction. Thus "birds" and "clothing" required relatively few features before they were identified, whereas "fruit" and "furniture" required more features.

On the other hand, there were dramatic differences in the kinds of feature which proved particularly helpful to subjects identifying the categories. The two artifactual categories were identified more with the aid of extrinsic features than of intrinsic features. The naturally-occurring categories, on the other hand, were identified more with the aid of intrinsic features.

The results, then support the "intrinsic/extrinsic" account of the results of Experiment 1 rather than the "probabilistic" account.

General Discussion

We have argued that "fuzzy" extensions are caused in part by the kind of features which form the intension of a category. Extrinsic features permit qualifiers to be applied when considering whether an exemplar is a member of a category (e.g., a pool table is OCCASIONALLY used inside houses). The extent to which qualifiers are necessary determines degree of membership in the category. The existence of degrees of membership, in turn, creates a fuzzy extension. Intrinsic features, on the other hand, do not permit qualifiers to be applied (e.g., an apple OFTEN contains vitamins?). Accordingly, when an intension is described entirely by intrinsic features, the extension is clear.

The results of the two experiments reported in this paper support our position. The two artifactual categories (furniture and clothing) are apparently represented largely by extrinsic features. The two naturally-occurring categories (fruit and birds) are represented largely by intrinsic features (see Experiment 2). The effect of typicality on membership judgments was very much more marked on the artifactual categories than on the naturally-occurring categories. (Experiment 1). Children also differed from adults more in their judgments of atypical artifactual items than in their judgments of atypical naturally-occurring items (Experiment 1). Presumably, the fuzzy extension created by extrinsic features is more difficult to learn than the clearer extension created by intrinsic features.

More traditional accounts of typicality effects (and other demonstrations of "fuzzy" natural categories) rely on the hypothesis that features are only probabilistically associated with a category to explain the demonstrated fuzziness. (e.g. Bourne, 1982; Rosch, 1975; also see Smith & Medin, 1981). Our account does not depend on this kind of uncertainty for its explanatory power. We believe that the features "covers people" and "is worn" are just as closely associated to the category "clothing" as the features "has feathers" and "has a beak" are to the category "bird". The results of Experiment 2 support this view, since subjects appeared almost as able to retrieve the artifactual

category names from a list of their features as they were to retrieve the naturally-occurring category names.

We suggest that a research focus on the properties of the features which describe a category's intension is more likely to elucidate categorization than earlier emphases on the uncertain nature of category boundaries.

Reference Note:

Barr, R. A., & Caplan, L. J. Some comparisons of the representations of two different classes of category Manuscript under review, Cognition.

References

Armstrong, S. L., Gleitman, L. R., & Gleitman, H. (1983). What some concepts might not be. Cognition, 13, 263-308.

Barsalou, L. W. (1982). Context-independent and context-dependent information in concepts. Memory and Cognition, 10, 82-93.

Bourne, L. E. (1982). Typicality effects in logically defined categories. Memory & Cognition, 10, 3-9.

Osherson, D. N., & Smith, E. E. (1981). On the adequacy of prototype theory as a theory of concepts. Cognition, 9, 35-58.

Rosch, E. (1973). On the internal structure of perceptual and semantic categories. In T. E. Moore (Ed.), Cognitive development and the acquisition of language. New York: Academic Press.

Rosch, E. (1975). Cognitive representations of semantic categories. Journal of Experimental Psychology: General, 104, 192-233.

Smith, E. E., & Medin, D. L. (1981). Categories and concepts. Cambridge, Mass.: Harvard University Press.

Table 1. Mean Response Times (sec) and Percentage of "No" Responses for "True" Sentences as a Function of Age, Category Type, and Typicality Level (all five typicality levels included; lower numbers indicate increased typicality).

ARTIFACTUAL CATEGORIES

		Typicality level				
		1	2	3	4	5
Adults	RT	3.118	3.127	3.219	3.267	3.318
	% "No"	0.0	8.9	9.5	34.5	66.4
Children	RT	4.264	4.438	4.792	4.579	4.773
	% "No"	3.5	7.8	21.5	32.1	57.9
Mean	RT	3.729	3.826	4.058	3.967	4.094
	% "No"	1.9	8.3	16.0	33.2	61.8

NATURALLY-OCCURRING CATEGORIES

		Typicality level				
		1	2	3	4	5
Adults	RT	2.958	3.011	3.148	3.256	3.207
	% "No"	1.8	0.9	5.9	24.4	17.5
Children	RT	4.133	4.116	4.544	4.564	4.451
	% "No"	4.5	4.5	8.3	21.0	19.4
Mean	RT	3.584	3.600	3.892	3.953	3.871
	% "No"	3.3	2.9	7.2	22.5	18.5

Empirical evidence for a global workspace theory of voluntary control.

Bernard J. Baars

Langley Porter Psychiatric Institute

University of California, San Francisco

1.0 Introduction.

Over the past ten years our research group has gathered a great deal of evidence on speech production through the medium of experimentally elicited slips of the tongue (e.g., Baars, in press; Motley, Camden & Baars 1983). For current purposes we will discuss two kinds of slips, spoonerisms and word-exchange slips. We can elicit spoonerisms with a variety of properties, e.g.

(*) (1) bad goof - gad boof (nonlexical)
(2) barn door - barn door (lexical)
(*) (3) vice nery - nice very (nonsyntactic)
(4) nery vice - very nice (syntactically OK)
(5) lice negs - nice legs (sexual comment)
(6) reel fejekted - feel rejected (depressed comment)

Word-exchange slips can be elicited, to create slips like:

(*) (7) "She touched her nose and picked a flower."
--- "She picked her nose ..." (socially embarassing)
(*) (8) "She hit the ball and saw her husband."
--- "She hit her husband ..." (aggressive affect)
(*) (9) "The teacher told the myths and dismissed the stories."
--- "The teacher dismissed the myths..." (hard to pronounce).
(*) (10) "She looked at the boy and talked softly."
--- "She talked at the boy and looked softly." (semantically anomalous).
(*) (11) "Is the gray sea below the blue sky?"
--- No, the blue sky is below the gray sea. (false)

As one can see, experimentally induced slips give us great control over relatively normal speech.

Two general phenomena have been observed. First, the likelihood of a generically acceptable slip is increased by priming with materials related to it. Second, the rate of slips that are designed to violate generic regularities is lowered compared to control slips. The priming phenomena may simulate the formulation stage of a normal speech plan, while the drop in rule-violating slips may reflect an mismatch editing capability.

1.1 Empirically simulating the formulation of a speech plan.

Motley and Baars (1979) showed that semantic priming increases the rate of phonologically primed slips. The influence of priming is not limited to linguistic factors. It certainly includes the speaker's physical and social environment. Thus Motley, Baars & Camden (1979) showed that subjects who were led to expect the possibility of an electric shock, made more slips like shad bok - bad shock, while male subjects who were run by an especially attractive female experimenter were more likely to make slips like lake muv - make love. Recently, Jabbour and Baars (1984) demonstrated that mood can affect the likelihood of slips. The Velten Mood Induction procedure (Velten, 1966) was used to establish a depressed mood, and in consequence, the rate of slips like deel fown - feel down, and juzza werk - was a jerk increased significantly.

One attractive theoretical proposal to model these effects is a "spreading activation network", with different subnetworks representing the different levels of control. Dell and Reich (1980, 1981) have developed a computer simulation of just such a network with two levels of control, phonological and lexical. Their model generates spoonerisms as well as correct responses.

But spreading activation may not be able to capture all the important properties of speech production. For example, no one has yet run a simulation of a spreading-activation network with many levels of control. If any single speech act reflects many simultaneous constraints, this creates a problem for these models. In spreading activation models, an increase in simultaneously interacting subnetworks will increase the background noise level, resulting in a system with very poor signal-to-noise ratio, and hence an unacceptably high error rate. Considerations like this suggest that spreading activation may not be the whole answer.

1.2 Empirically simulating the editing of a speech plan.

Not all manipulations of the slip task increase the rate of slips. By designing slips that violate some level of control, and comparing it with matching rule-governed slips, we have found a number of cases where the rate of rule-violating slips drops precipitously, sometimes even to zero (Baars, 1977; Baars, Motley and MacKay, 1975). (All starred slips (*) listed above violate such generic rules.)

1.21 Bottom-up flow of control.

In general, we have discussed these effects as a kind of editing (Baars, Motley, & MacKay, 1975; Motley, Camden & Baars, 1979). The notion of editing has two special implications. Consider a slip like (1) above. This is a spoonerism, i.e. an exchange of consonants, an error at the level of phoneme

sequencing. However, the rate of these errors is influenced by whether the resulting phoneme string consists of real words or nonsense syllables. If it is lexical, the slip occurs much more frequently. Thus, the likelihood of a change in phoneme-sequencing is affected by its consonance with lexical regularities. But in the normal linguistic hierarchy, the lexical level is above the phonemic level. This implies that the flow of control in speech production can go "bottom-up" from a lower structural level to a higher one.

The same argument can be made in the editing of word-exchange errors, which take place at the level of word-sequencing (syntax). The probability of such slips is very much affected by the semantic and pragmatic outcome of the word-exchange --- whether it creates a semantic anomaly, a false statement, or a statement that is socially embarassing (Baars, 1977; Baars & Mattson, 1981). Again, a word-sequencing slip is affected by structurally higher levels of control.

All these results imply that the flow of control in speech planning is not strictly top-down. It must be possible to have information flow in the opposite direction. This contradicts some suggestions that speech production is exclusively top-down (e.g. Garrett, 1976; Fay, 1980), and is in accord with findings from spontaneous slips, which also support the presence of bottom-up flow of control (e.g. Harley, 1984).

Spreading activation networks do support a flow of control in both directions (Dell & Reich, 1981; Rumelhart & McClelland, 1982). Indeed, Dell & Reich (1981) have shown that a computer simulation of a spreading activation model with phonological and lexical levels can simulate the "editing" results of Baars, Motley, and MacKay (1975), namely the finding that lexical slips are made far more often than matched nonsense slips. Dell & Reich argue, in effect, that spreading activation is all that is needed to account for the "editing" results described so far.

1.22 Editing involves mismatch detection.

This brings us to the second criterion for editing: in everyday usage when we speak of "editing" we mean a review process in which one person (such as a newspaper editor) checks the output of another (a reporter, perhaps) with respect to a set of criteria --- of linguistic adequacy, of conforming to the newspaper's editorial policy, and the like. That is to say, the editor is always monitoring for matches and mismatches with respect to his or her criteria. But, so far at least, there seems to be no mechanism whereby spreading-activation models can detect mismatches (McClelland & Rumelhart, 1982; Dell & Reich, 1981).

To show that editing in the sense defined here occurs in speech production, we needed to demonstrate not only that bottom-up flow of control occurs, but also that the speech production

system can detect mismatches between a speech plan and prior criteria. Motley, Camden & Baars (1982) report that for a task eliciting sexually expressive slips (lake muv - make luv, bice noddy - nice body), there is a large and immediate rise in the electrical skin resistance on sexual slip trials even if the slip is not actually made. On neutral control trials, there is no such effect. Since the Electro-Dermal Response (EDR) is one of the standard measures of the Orienting Response --- a predictable physiological concomitant of surprise, novelty, and mismatch with expectations --- these results suggest that a mismatch was detected even when the slip was successfully avoided.

1.3 Modular dissociation.

The above results suggest that unadorned spreading activation is not enough to explain what we know about speech planning and production. There is an additional fact that is not handled by current models: that is the ability of entire editing criteria to "drop out of" or "lock into" the speech planning process, depending upon contextual factors. This is obvious in everyday life: speech in a locker room is likely to violate criteria that obtain at formal dinner parties (and vice versa). We speak differently to children than to adults, to professional colleagues than to family members.

A result from Baars, Motley and MacKay (1975) supports this observation even with respect to our willingness to speak nonsense: when subjects in the spoonerism task saw only nonsense word pairs (some of which could slip into genuine word pairs, while others only changed into other nonsense syllables) they stopped editing for lexical status: no longer was there any difference in the rate of lexical vs. nonsense slips. It is as if the lexical editing criterion dropped out as a whole. However, introducing lexical items into the task context (as filler items) reinstated lexical editing to its previous level. This kind of discrete "modular dissociaton" of an entire editing system seems alien to the spirit of spreading activation, in which every knowledge source can spread polymorphously to every other.

There are several other kinds of such modular dissociation. The most obvious is the existence of slips themselves --- both slips of the tongue and slips of action. Slips are actions that would not have happened if the proper source of knowledge had been brought to bear on the speech planning process before execution. They represent a failure of internal communication between the knowledge source and the planning process. But only a momentary failure: the evidence is good for our commonsensical assumption that we would avoid most slips if we just took more time to think about our speech plan (Dell, in press). Slips, then, involve a momentary dissociation between two components of the planning process. But this disconnection is much more transitory than the dissociation of lexical editing mentioned above.

But perhaps the most significant example of modular dissociation comes during the process of acquiring a skill. As the skill becomes more and more practiced, it tends to become more and more autonomous and dissociated from detailed voluntary control (LaBerge, 1980; Shiffrin & Schneider, 1977). In addition, and especially important for our theoretical approach (2.0), serial processing tends to change to parallel processing.

The notion that much processing is done by special-purpose modules is becoming increasingly popular (e.g. Fodor, 1983; Baars, 1983). It has a number of empirical sources of support, both psychological (e.g. Swinney, et al, 1982) and neurophysiological (Mountcastle, 1980; Geschwind, 1979). Further, computer scientists have been investigating the properties of parallel distributed systems, which consist of collections of special-purpose modules that can either cooperate or compete to perform some task (Reddy & Newell, 1974).

A number of action errors collected by Reason (1984a) suggest that overpracticed components of actions can become quite autonomous. Thus:

"I intended to place my hairbrush in its usual place by the bookcase. I put my boyfriend's lighter there instead."

"I had an appointment at the dentist's, but went to the doctor's instead."

"I went into my room intending to fetch a book. I took off my rings, looked in the mirror and came out again --- without the book."

Reason (1984b) calls the low-level components that can be exchanged with others, or dropped as a whole, the action "schemata", and suggests that they are mostly unconscious components which "can be independently activated and behave in an energetic and highly competitive fashion to try to grab a piece of the action."

Since it is a relatively unitary "chunk", it is likely that a module can be called upon as a whole to perform its function. If we are skilled bicycle riders, we do not want to leap on a bicycle only to find that we have to bring together our spatial orientation abilities, our motor control functions for the feet, legs, and arms (separately, of course), our balance mechanism, visual abilities, etc. Rather, we would like to call upon a single "bicycle riding processor", which will unite and organize all the necessary components of bicycle riding.

However, it would be awkward in getting off the bicycle to find that the "bicycle-riding module" could not decompose and reorganize, in order to make parts of the same module available for use in standing, walking, and running. For these actions we also need spatial orientation, motor control, balance, and vision, just as we do in riding a bicycle. And if something goes

wrong with the bicycle while we are riding it, it is certainly important to find out which component of bicycle riding must be checked and modified to deal with the problem.

So it seems that we need two abilities that may seem at odds with each other: the ability to call upon complex functions in a unitary way, and the ability decompose and reorganize those functions when the task or context changes. Modules are like Chinese puzzle boxes: In computer science terms, they are defined recursively, so that a module may consist of a coalition of modules, and in turn, the original module may be a member of an even larger coalition of modules which can also be viewed as a module. We should not expect to define any functional module independent of task and context, although there will be some tasks and contexts that occur so commonly that they may require the services of a relatively permanent control system.

1.4 The need for inter-modular access.

Modules cannot be leak-proof. As we have argued above, it must be possible to decompose modules, and to recombine them into larger modules. Further, even when a module remains stable, it must be able to accept input commands and parameters, and to output commands to effectors; it must be able to accept feedback regarding its success or failure in achieving its governing goal. And when it malfunctions, it must be possible to find which particular submodule must be changed to deal with the problem. Thus there is a need for inter-modular access, and we argue below that consciousness is heavily involved in cases that require novel means of access.

A special case of inter-modular access is our ability to talk about, or act upon, any conscious content. If we assume that a conscious content is the result of a specialized module, and that speaking is also controlled by specialized systems, the question arises how the speech module gains access to the conscious content. The system architecture proposed below (2.0) provides one answer. In general, we claim that specialized modules can have access to many others. However, novel access across modules loads central limited capacity.

2.0 The Global Workspace System (GWS): A system architecture that supports both Formulation and Editing.

What sense can we make of these facts? Fortunately, there is a very simple interpretation. Cognitive scientists have for some time worked with system configurations with properties very much like those described above. These systems consist of multiple specialized processors, which are by themselves quite active and independent, sometimes called "parallel distributed processors" (PDPs). Together, they create a "society" of specialized systems, each able to handle some predetermined type of problem.

The trouble with a PDP society is that, although it can handle routine tasks just by assigning such tasks to the appropriate specialist, it has difficulty in solving new problems, which are best approaced by joining different specialties. To permit interaction between the distributed specialists, various researchers have added a global workspace to the system, a memory whose contents are broadcast to all processors (e.g. Reddy and Newell, 1974; Erman and Lesser, 1975; Hayes-Roth, 1984). Recent work in computer science also suggests that the GWS system may provide a "general framework for problem-solving" (Hayes-Roth, 1983). The global workspace (sometimes called a blackboard) is really a publicity organ for the distributed system. In principle, it allows any processor that can gain access to the global workspace to pose a problem to be solved by all the other processors. (While global messages are available to all processors, they can only be interpreted by those specialists which are relevant to the content of the message.) A detailed discussion of its functional pros and cons, and its mode of operation, may be found in Baars (1983).

2.1 The GWS theory and the conscious limited-capacity component of the nervous system.

In a series of recent papers I have pointed to cognitive, functional, and neurophysiological evidence in favor of something like a GWS to model the limited-capacity component of our nervous system, which is so closely associated with conscious experience (Baars, 1983; Baars, in press a, b; Baars, in preparation). Others have also described competition of action systems for access to some limited capacity system as a model for conscious or attentional phenomena (Shallice, 1976; Norman & Shallice, 1980; Reason, 1984; Anderson, 1984). The GWS theory offers a functional reason why it would be advantageous for goal systems to control the limited-capacity component: Since global goals are broadcast to all modules in the system, they can recruit effectors able to carry out their goals. Indeed, without access to global broadcasting, no truly novel goal is likely to match up with the appropriate effector systems.

2.2 The GWS model can incorporate spreading-activation, plus dissociation, cooperation and competition between modules.

A central aspect of the GWS architecture is the ability of many different processors to cooperate in supporting some global message, or to compete against it. Cooperation between different modules may occur by spreading activation via the global workspace. But modules can act as a whole, so that the system supports modular association and dissociation with respect to any globally displayed process. And of course, any module, or set of modules, can act so as to compete for global access with any global message, thereby serving to "edit" the global message. Norman and Shallice (1980) refer to this kind of competition as "contention scheduling". Indeed, we could view cooperation and competition as two different operating modes of the system, although it is quite possible for these modes to alternate rapidly. As we suggested above, cooperative interaction through the global workspace is equivalent to the Formulation of the speech plan which is needed to explain our findings about multi-level priming of slips and error-free speech, while competition between modules corresponds to the Editing process, for which evidence has been cited above.

References

Baars, B.J. (1977) Is there semantic editing before speech articulation? Unpublished doctoral dissertation, Department of Psychology, UCLA.

Baars, B.J. (1980) On eliciting predictable speech errors in the laboratory: Methods and results. In V.A. Fromkin (ed.) Errors of speech and hearing. NY: Academic.

Baars, B.J. (1983) Conscious contents provide the nervous system with coherent, global information. In R. Davidson, G. Schwartz, and D. Shapiro (eds.) Consciousness and Self-regulation, Vol. 3. NY: Plenum.

Baars, B.J. (in press, a) The cognitive revolution in psychology. NY: The Guilford Press.

Baars, B.J. (in press, b) Can involuntary slips reveal one's state of mind? --- with an addendum on the conscious control of speech. In M. Toglia and T.M. Shlechter (eds) (in press) New directions in cognitive science. Norwood,N.J.: Ablex.

Baars, B.J. (in press, c) Biological implications of a global workspace theory of conscious experience. In G. Greenberg (ed.) Language, cognition, consciousness: Integrative levels., Englewood, N.J.: Erlbaum.

Baars, B.J. (in preparation) A Cognitive Theory of

Consciousness. London: Cambridge University Press.

Baars, B.J. and Kramer, D. (1982) Conscious and unconscious components of intentional control. Proceedings of the 4th Cognitive Science Conference, Ann Arbor, MI.

Baars, B.J. and Mattson, M.E. (1981) Consciousness and intention: A framework and some evidence. Cognition & Brain Theory, 4 (3), 247-263.

Baars, B.J., Motley, M.T., & MacKay, D.G. (1975) Evidence for lexical editing in artificially elicited slips of the tongue. JVLVB.

Brown, R., & McNeill, D. (1966) The "tip of the tongue" phenomenon. Journal of Verbal Learning and Verbal Behavior 325-337.

Clark, H. & Clark, E. Psychology and language. NY: Harcourt, Brace, Jovanovich.

Dell, G.S. & Reich, P.A. 1(1981) Stages in sentence production: Analysis of speech error data. JVLVB 20, 611-629.

Dell,G. & Reich, P. (1980) Toward a unified model of slips of the tongue. In Fromkin, V.A. (ed.) Errors in linguistic performance, NY: Academic.

Ericsson, K.A. & Simon, H.A. (1980) Verbal reports as data. Psychological Review, 87, 215-51.

Erman, L.D. & Lesser, V.R. A multi-level organization for problem solving using many, diverse, cooperating sources of knowledge. Proceedings of the 4th Annual Joint Computer Conference, Georgia, USSR, 1975, pp. 483-490.

Fodor, J.A. (1983) The modularity of mind: An essay on faculty psychology. Cambridge, MA: Bradford (MIT Press).

Geschwind, N. (1979) Specializations of the human brain. Scientific American 24 (3), 180-201.

Jabbour, O. & Baars, B.J., (1984) Experimentally elicited slips of the tongue as a measure of depression. Eastern Psychological Association, Baltimore, MD.

Langer, E.J. and Imber, L.G. (1979) When practice makes imperfect: Debilitating effects of overlearning. Journal of Personality and Social Psychology, 37 (11), 2014-2024.

Mandler, G.A. (1975) Mind and emotion. NY: Wiley.

McClelland, J. L., & Rumelhart, D. E., (1981) An interactive model of context effects in letter perception: Part I: An account of basic findings. Psych. Review, 88, 375-407.

Motley, M. T., Camden, C. T., & Baars, B.J., (1979) Personality and situational influences upon verbal slips: A laboratory test of Freudian and pre-articulatory editing hypotheses. Human Communication Research, 5 (3), 195-202.

Motley, M.T., & Baars, B.J. (1979) Effects of cognitive set upon laboratory-induced verbal (Freudian) slips. Journal of Speech & Hearing Research, 22 421-432.

Motley, M.T., Camden, C.T., & Baars, B.J., Polysemantic lexical access: Evidence from laboratory-induced doubles-entendres. Communication Monographs, 50, 193-205.

Natsoulas, T. (1982) Dimensions of perceptual awareness. Behaviorism, 10, 85 - 112.

Norman, D.A. (1981) Categorization of action slips. Psych. Rev., 88, 1-15.

Norman, D.A. & Shallice, T. (1980) Attention to action: Willed and automatic control of behavior. Center for Human Information Processing, UCSD, La Jolla, CA: CHIP Document 99.

Pani, J. R. (1982) A functionalist approach to mental imagery. Paper presented at the 23rd Annual Psychonomic Society Meeting.

Reason, James, (1984a) Lapses of attention in everyday life. In R. Parasuraman & D.R. Davies (eds.) Varieties of attention, NY: Academic Press.

Reason, James, (1984b) Little slips and big disasters. Interdisciplinary Science Reviews, 9(2), 179-189.

Reddy, R. & Newell, A. (1974) Knowledge and its representation in a speech understanding system. In L.W. Gregg (ed.) Knowledge and cognition, Potomac, MD: L. Erlbaum Associates.

Schneider, W. & Shiffrin, R.M. (1977) Controlled and automatic human information processing. I. Detection, search, and attention. Psychological Review, 84, 1-66.

Shallice, T. (1976) The Dominant Action System: An information-processing approach to consciousness. In K.S. Pope & J.L. Singer (eds.) (1978) The stream of consciousness: Scientific investigations into the flow of experience. NY: Plenum.

Shiffrin, R.M. and Schneider, W. (1977) Controlled and automatic human information processing: II. Perceptual learning, automatic attending, and a general theory. Psychological Review, 84, 127-190.

Connectionist Parsing

Garrison W. Cottrell[1]
Department of Computer Science
University of Rochester

Abstract

We have proposed a neural network style model of language processing in an effort to build a cognitive model which would simultaneously satisfy constraints from psychology and neurophysiology. This model was successful in disambiguating word senses in semantically determined sentences, but was unable to distinguish Agent from Object in semantically reversible sentences such as "John loves Mary." In this paper we rectify the matter by specifying the syntactic portion of the model, which is a massively parallel, completely distributed connectionist parser. We also describe the results of a simulation of the model.

1. Introduction

We have proposed a massively parallel, highly distributed neural network model of language comprehension (Small, Cottrell & Shastri, 1982; Cottrell & Small, 1983; Cottrell 1984, 1985) based on the connectionist paradigm of Feldman & Ballard (1982). Similar systems have been proposed by Gigley (1982), Pollack & Waltz (1982; 1985) and Selman & Hirst (1985). Our model specifies independent pathways from the lexicon to syntactic and semantic modules that operate in parallel, mutually constraining one another's activities. Previous work has concentrated on the semantic module and the lexical access mechanism. In this paper we present the design of the syntactic processor, and the results of a simulation of the model. While this is a preliminary version, lacking many of the features one would want in a complete parsing system, it demonstrates the feasibility of the approach, and there are aspects of this model which are rather general and interesting as partial specifications of a parsing mechanism in their own right. These include the mechanism for assigning constituents to their roles which has a natural interface with our model of semantic role assignment, and the ability to represent syntactic attachment preferences. Also we might include here the fact that this parser uses the massive parallelism inherent in the connectionist approach, with the concommitant distributed decision making. And, unlike the parser of Pollack & Waltz (1982; 1985), there is no interpreter that builds a network based on the input sentence and then runs it in parallel; this parser uses a network that is fixed, yet responds flexibly to the input.

In order to implement this parser, we developed a grammar formalism suitable to our needs, and developed a LISP program that takes as input a dictionary and a set of grammar rules, and outputs commands to the ISCON simulator (Small, et al. 1982) to build the network. The network implements a top-down parser. Expectations compatible witha an S are set up. As input comes in, (words are activated at fixed intervals) the structures compatible with the input continue to be active, while productions that are incompatible with the input are turned off. After a settling period, a stable coalition in the network represents the parse tree. Ambiguous lexical items at the leaves are disambiguated through feedback from the parse tree as it develops. One important difference from previous parsers of this type (Pollack & Waltz, 1985) is that the assignment of constituents to their syntactic roles is explicitly represented by nodes in the tree called *binding nodes*. The combination of top-down expectations and bottom-up evidence comes about at the binding nodes, which combine the evidence and compete with one another through mutual inhibition. These nodes can be used to interface with case role assignment in the semantic analyzer.

---

[1]Author's present address: Institute for Cognitive Science C-015, U.C.S.D., La Jolla, California 92093.

As a first cut, this parser is less elegant than one would hope, and less powerful than what is needed for a "real" parser. Although it can handle recursive constructs, and thus has a mechanism for "recruiting" constituent recognizers as needed, no attempt was made to do the same for buffer positions, so it accepts only a sentences of a fixed maximum length. Also, the fact that the network is of fixed size means that there are only a fixed number of constituent recognizers, so the number of constituents of each type that can occur in a sentence is limited. We believe that McClelland's programmable buffers (McClelland, 1985) might be adaptable to resolving these problems, but exactly how is not obvious. Also, this parser has not been tested on an extensive grammar, therefore, take any claims with a grain of salt. Finally, we haven't done anything about: (1) optional repetition of constituents, (2) feature marking of constituents (hence no feature agreement checking), (3) anaphora, or (4) gapping. We would hasten to point out that this doesn't mean that these features can't be implemented in this framework; simply that we haven't done it yet.

The rest of this paper is organized as follows: We first provide a bit of background material on our model to motivate the parser's design, and briefly review some of the psycholinguistic and neurolinguistic data on syntax. Then we describe our grammar formalism, leading into a description of our parser, followed by a sample run of the implementation on a sentence. We will not give an introduction to connectionism due to space limitations. Therefore we will be assuming some familiarity with the paradigm[2].

2. Background

Overview of the model

The overall model consists of a four component, three level network of units that operate in parallel by spreading activation and mutual inhibition, shown in Figure 1. The Lexical level consists of units representing the words in the language, presumably activated by phoneme or letter recognition networks at a lower level, such as those developed by McClelland & Rumelhart (1981). These units in turn, activate units at the word sense level, which buffers the syntactic and semantic features of their definitions. If a word is ambiguous, features corresponding to all of its definitions are activated and compete with one another. From here, the semantic features activate relational nodes in the case network, which makes the assignments of conceptual constituents to their case roles (Fillmore, 1968; Cook, 1979) based on the "best fit" among the (possibly several) predicates, fillers, and case roles. The representation in this network is order-less, and thus it can operate by itself if the roles can be determined from semantic constraints alone, but not when syntactic information is necessary. For example, there is no way for a purely semantic analyzer to make the assignment of Agent and Object in *John loves Mary*, since both John and Mary are equally likely candidates for both roles. In the current design (Cottrell, 1985), fillers are assigned to their roles through the competition

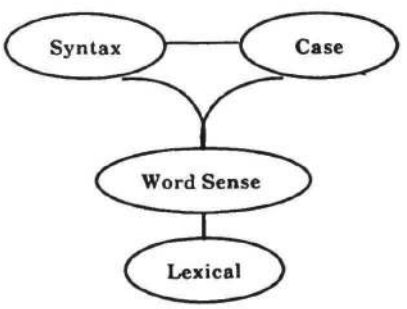


Figure 1. Overview of the system.

[2]The uninitiated reader may consult Feldman & Ballard (1982).

of *binder* units, which explicitly encode the assignments (see Figure 2). Through mutual inhibition, one of these binder units will "win", representing the assignment of, say, Concept1 to the Agent role. It is these units which motivated the design of the syntactic processor, since the kind of information now needed from syntax is of the form "NP1 is the Subject, and NP1 corresponds to Concept1, and the verb is Passive". We will not worry about computing that NP1 corresponds to Concept1 here. Our parser will compute that NP1 is the Subject, however, in the same form as such constituent-role assignments are done in the case network, i.e., through the competition of binding nodes. The result gives a simple interface between the two systems: the binding nodes in one constrain the binding nodes in the other. For example, we can implement the Passive transformation as in Figure 3. Thus, our design of the parser started "from the inside out", with the necessity of binding nodes.

The other thing we would like from syntax is the syntactic disambiguation of lexical items, to prevent spurious predicates and fillers from confusing the semantic system. This can be done through feedback to the syntactic features in the word sense buffer which form a grammatically correct sentence. The decision machinery in this buffer will then kill off the meaning features for the meanings corresponding to the wrong syntactic class, which then will stop sending input to the semantic analyzer.

Psycholinguistic and Neurolinguistic Data

We will very briefly (due to space limitations) review some relevant psycholinguistic and neurolinguistic data. We will try to point out the relevance to the rest of the paper here, rather than return to it later. For a deeper treatment, see (Cottrell, 1985). The current best estimate of what

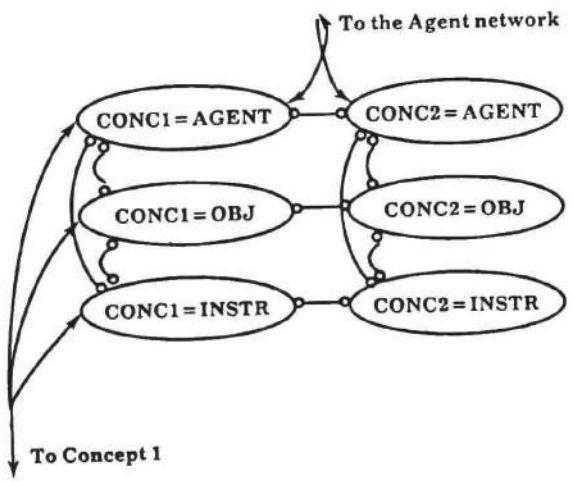


Figure 2. Binding nodes in the case network.

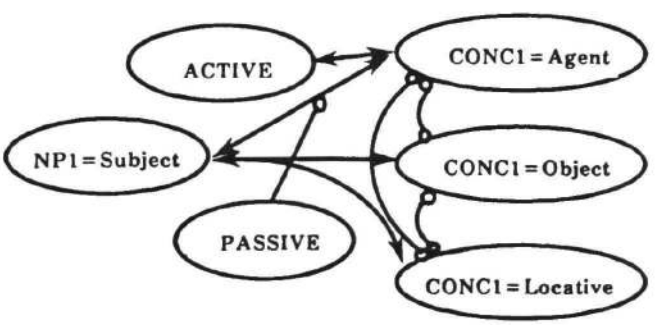


Figure 3. The Passive transformation (assuming the features "Active" and "Passive").

happens when a person hears a word is that all meanings are initially activated, and then by 200 milliseconds after the end of the word, only the appropriate one remains active (Seidenberg et al., 1982). This result is for sentence-final noun-verb and noun-noun ambiguous words, regardless of the strength of context. The exception is the case of noun-noun ambiguous words that are preceded by a word that is strongly related to one of its meanings, in which case it appears that only one meaning is activated. On the other hand, for noun-noun ambiguous words in the first clause of a two clause sentence, where the disambiguating information is contained in the second clause (as in *the teacher looked at her* **pupils** *and noticed that they were dilated*) people appear to be able to maintain both meanings active for as long as 500 milliseconds (Hudson & Tanenhaus, 1984). Thus a model of the human parsing mechanism should activate all meanings of an ambiguous word, and allow them to continue as long as they are compatible with the rest of the sentence, or until a decision is necessary, for example when the sentence has ended.

Turning to higher-level considerations, Frazier (1979) has proposed several principles of the human parsing mechanism including the *Minimal Attachment Principle*: Attach incoming material into the phrase marker being constructed using the fewest nodes consistent with the well-formedness rules of the language. The Minimal Attachment Principle is simply a statement of what appears to be a sound strategy: use the fewest nodes possible to parse a sentence. Without having described the model yet, we can explain Minimal Attachment in terms of it as follows: Imagine a grammar representation which is *active*, in the sense that as parts of productions are recognized, activation spreads to the next part of the production. Different productions in the grammar *compete* for the attachments of constituents that are found in the input. The more nodes involved in a particular interpretation, the farther the activation has to spread, and the longer it takes to activate those nodes that the input actually attaches to. Meanwhile, if there is a representation that matches the input that involves fewer nodes, this will become activated faster and get a head start over the representations involving more nodes. Thus our model explains Minimal Attachment as a *timing* phenomenon, involving the latency of activating simple versus complex representations through a spreading activation network.

Rayner, Carlson & Frazier (1983) have shown that there are purely syntactic preferences for attachment of PP's to NP's vs. VP's in such sentences as *the cop saw the burglar with the binoculars*. When the semantically preferred attachment is different from the syntactically preferred one, people appear to "backtrack" briefly to recover. This argues for an independent contribution of syntax and semantics to the attachment process. It does not necessarily mean that one precedes the other; one system may just operate more slowly than the other. Although we will not delve into it here, there is a natural way to represent these preferences in our system as inhibitory bias (presumably learned through frequency of attachments) between nodes representing competing attachments for a constituent.

Recent data on agrammatic aphasics also supports an independent syntactic processor; Linebarger et al. (1983) have shown that these patients who appear unable to use syntactic clues to comprehend sentences can nevertheless make relatively complex grammaticality judgements. This result implies that agrammatics can *compute* syntactic representations, but can't *use* them to form semantic interpretations. We interpret this data as support for our overall model, given that the observed behavior can be roughly accounted for by a lesion to the constraints between the syntactic and semantic modules.

The Grammar Formalism

Recall that what we want from syntax is information such as "NP1=Subject", represented explicitly as units with this as their value. The grammar we have developed to enable us to automatically generate the parsing network explicitly represents roles and constituents that can fill those roles. Hence we call it a *role-constituent grammar*. We are not aware of any formalism in linguistics that is similar to this although it appears to be weakly equivalent to a context free grammar. However, as has often been remarked, the notation one uses can affect the way one thinks about a problem. In this case, the notation is handy for generating a network which contains binders for constituents to their roles in parent constituents. An example grammar is shown in Figure 4. The left hand side of a production in this grammar is a constituent. The right hand side is a list of roles

S-> Subject.{NP} Predicate.{VP}
Predicate.{VP}

VP-> Main.{VERB} DirObj.{NP}
Main.{VERB} IndObj.{NP DirObj.{NP}
Main.{VERB}

NP-> DetPhrase.{DET} Head.{NOUN}
Head.{NOUN PRO}

Figure 4. A sample role-constituent grammar.

followed by a set of constituents that can fill those roles. In this example, the Head of a noun phrase can be filled by either a PRO a NOUN.

Since there are sometimes a number of alternative constituents that may fill a role, the dot between the role and the set of possible constituents corresponds to a set of binders that must compete for that role. On the other hand, every set that a constituent is in corresponds to a possible role for that constituent, so that binders for that constituent to these roles must compete as well. For example, any particular NP could be the Direct Object, Indirect Object, etc., so binders to these roles must be a) built in to the network and b) mutually inhibitory. The network is set up so that unless a role is expected, then that binder doesn't become activated, so in practice competition does not involve all of the binders.

Another thing to notice about this grammar formalism is that the role a constituent can fill is context sensitive within a production. For example, in the first production for an NP, a NOUN can fill the role of the Head, but not a PRO. In the second production, the Head can either be a NOUN or a PRO. Thus there must be two "Head recognizers", one for each kind of Head.

Now given this grammar, a dictionary containing the possible syntactic classes of the lexical items, and some "magic numbers" (how many copies of each kind of constituent recognizer there will be, and how long the lexical buffer will be), we can generate a network that will recognize sentences that match these rules. After a tour of the lexical buffer, we shall describe this network in greater detail.

The Lexical Buffer

We used a slightly different version of lexical access from the model reported in (Cottrell, 1984) basically because it was easier to implement with respect to the interface with the syntactic analyzer. However, it has some interesting properties in its own right, which we will describe here. For comparison purposes, we show the network for "deck" in Figure 5. The lexical node "deck" activates "grandmother cells" for each of its definitions. These are self-stimulating, to keep the definition around after the lexical node decays. (Thus the lexical node can be re-used later. Activation of successive buffer positions is enabled by control nodes that sequence through the buffer. This figure represents one buffer position.) The definition nodes, in turn, are connected to feature nodes representing syntactic class and meaning. The definition nodes are also mutually inhibitory, but the inhibition weights are such that they can't defeat one another until decisive feedback to the meaning nodes supports one "DEF" node over another. The "meaning" nodes are connected to the case network, and have no further connections to the syntax network. The syntactic class nodes are shared between meanings of the same class. Thus feedback to one of these from a role node above will support both definitions within a class, and kill off the out-of-class syntactic node, followed by the out-of-class meaning and definition nodes. (Of course, if there is only one meaning for a syntactic class, then feedback to that class is enough to cause that "DEF" and meaning node to win.)

Feedback to one of the meaning nodes, e.g. SHIP'S, on the other hand, will kill off the other meaning nodes, and the supported definition node (DEF1) will increase the activation of NOUN, which causes it to win over VERB. However, "DEF2" will get both extra support and extra inhibition.

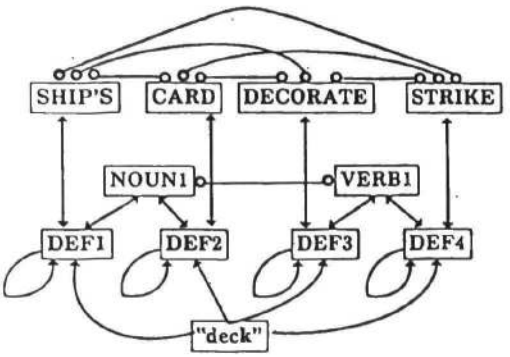


Figure 5. A word sense buffer position. Inhibitory links between DEF nodes omitted.

It gets extra inhibition from "DEF1", but since there is a positive pathway between the two, through the NOUN node, it also gets extra support. The result (with certain parameters) is that "DEF2" gets inhibited, but not below threshold. However, since "CARD" has been killed by "SHIP'S", "DEF2" is invisible to the rest of the network. Only the features compatible with the "DEF1" are visible. A prediction this network makes is that it would be easier to recover from a *within class* mistaken meaning choice, since the "DEF2" node is still firing.

This is the decision machinery for lexical disambiguation. Now, all we need is feedback.

3. The Parser

The parsing network is generated by a LISP program that reads the grammar and a dictionary and outputs commands to the ISCON simulator to build the network. We will start by explaining at a high level what is generated by a grammar rule for a constituent with several productions. Then we will go into slightly more detail about how productions for a single constituent compete with each other, and how the binding nodes compete with each other. Following this, we will skim over some of the ugly details of the whole process, and mention some suggestions as to how they might be circumvented in future designs. This will conclude the "Parser" section; the following section will describe a sample run of the parser.

Overview of the Network Generated by a Production

Figure 6 shows a production and a high level description of the network fragment generated by it. The nodes in Figure 6 are actually representative of networks in the implementation. For the purposes of exposition, however, this level of detail is more appropriate. The S recognizer is connected to two "production recognizers" which correspond to the two productions for an S in the grammar. These implement a simple voting scheme for comparing evidence between the two productions. If the difference in the amount of evidence for the two productions is greater than a certain amount (called the *competition window*) the one with less evidence gives up.

The production recognizers are connected to "role recognizers", that sequence through the roles of the production. As bottom-up evidence comes in for the Subject role, for example, expectations are set up for the Predicate role, activating that recognizer. As mentioned above, since roles can be filled by different constituents depending on what production they are in, there have to be two Predicate recognizers. In any case the control is different for the two; the Predicate recognizer in the first production has to wait for the Subject to be recognized.

The role recognizers operate by enabling the binder nodes for their role, and setting up expectations for the possible role-filling constituents. These constituents are recognized by another set of constituent recognizers that are activated by the role recognizer from a pool of inactive constituent recognizers of that class. As input comes in, it activates the constituents appropriate to it

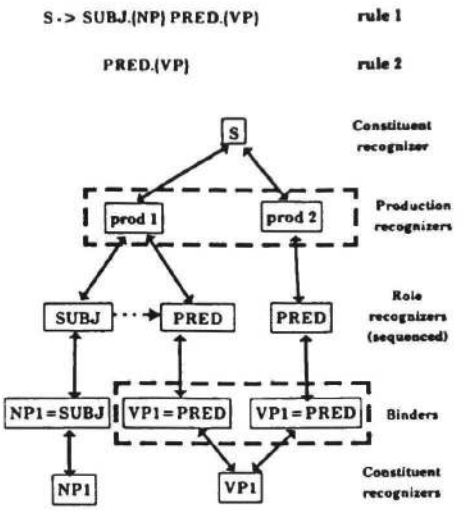


Figure 6. The network generated by a grammar rule.
Dashed rectangles indicate possible competitors.

that are expected, and the binders for that constituent compete. If it was possible, for example, for both Predicate recognizers to be active (it isn't, given that the presence of a Subject would kill off production 2), the two VP1=Predicate nodes would compete with one another. In practice, the competition would be between binders from a constituent to roles in different higher level constituents, such as ones with common prefixes. Binders get feedback from the production recognizers, so that a binding to a well satisfied production will tend to win over others.

This network is repeated for every production in the grammar. It "bottoms out" on syntactic class nodes, for example NOUN or VERB, which are directly connected to the definition node for a lexical item.

Implicit in this description has been the fact that this parser starts out with top-down expectations, and merges them with bottom-up input. So, in the beginning, expectations are generated for everything that could start an S. This means that the constituent recognizers are enabled and ready to go. Then the input that matches with those compete through the binder nodes. The binder nodes represent a merge of bottom-up input with top-down expectations. Whether or not we have a good algorithm for combining these types of evidence, we at least have a paradigm for testing different evidence combination functions.

The end result of a successful parse is that there is a unique binder that has won over its competition for every constituent, and one production that has won for every constituent, from the top level S on down. In the next sections we cover some of the aspects of this model in more detail.

Production Competition

The production competition is a simple voting scheme which favors longer productions over shorter ones, and thus favors "late closure" (Frazier, 1978) of constituents. It works as follows: Every production gets two votes for every "filled" role. "Filled" here means that a binder for a constituent filling that role has won over all competing binders. This is communicated to the production competition network by a higher firing rate for that binder than is possible while it is in competition with other binders. Each production gets one vote for an unresolved role. "Unresolved" here means that there is evidence that a constituent for this role exists, that is, a binder for a role-filling constituent is firing, but it is still competing with other binders. The evidence rule for each production then is:

```
if Max(votes for all productions) - MyVotes < Competition Window
then continue competing;
else lose;
```

In the current implementation, the competition window is 2 votes. Since the votes for this production are included in the Max of votes for all productions, if this is the winning production, the left hand side is zero, and the production continues. If another production gets two votes ahead, it kills this production. This evidence combination rule favors writing grammar productions with alternatives that are not greatly different in length; very long productions, even if only partially satisfied, would always win over shorter ones. However, this seems to be a natural restriction on grammars.

While particular rules like this are certainly arguable (compare to the "normalized" rule used by Selman & Hirst (this volume)), the point is that we have a good framework for evaluating such rules; they are intrinsically interesting because they are completely *local* to the production competition network; there is no global interpreter making decisions about the "best fit" to the grammar. Thus this is a testbed for exploring decision algorithms for a parser that works in a completely distributed manner.

The careful reader will have noticed a problem with the rules as given. See Figure 7 (please forgive the "mix and match" linguistics). Given that these two productions have a common prefix, and given input that matches the prefix, i.e. a "lone" NP, how does production 2 ever win? One answer is to add a "Closure" role to the end of every production that requires no filler (see Figure 8). Then, if no PP comes along, production 2 gets an extra 2 votes and wins. The problem is deciding when the Closure role should fire. Simply having it start firing after an arbitrary interval won't work, since different PP's will have different recognition latencies; they may arrive quickly, unfairly beating production 2 before the Closure role fires (the proper attachment isn't necessarily to this NPbar) or it may arrive too late to come to the rescue of production 1. The answer is to make the Closure role fire when either the PP is recognized, or input inconsistent with a PP is recognized[3]. Then, the shorter production will not win too soon. We can then depend on the semantic feedback to resolve the attachment of the PP (the constraints between syntax and semantics are a two-way street; bindings in semantics will support one attachment over another) and the reader can check that our evidence rule will make the appropriate decision once this binding has been resolved (and not until then). While this appears plausible, we won't feel confident in it until we have tested it on a more extensive grammar[4].

```
Nbar-> Head.{NP} Mod.{PP}    (1)

       Head.{NP}             (2)
```

Figure 7. Production competition: the closure problem.

```
Nbar-> Head.{NP} Mod.{PP} Clos.{}    (1)

       Head.{NP} Clos.{}             (2)
```

Figure 8. The closure problem: A solution.

---

[3]This can be computed directly from the grammar, using the "Follow set" (Aho & Ullman, 1977) (used in predictive parsers of computer languages) of the NP, in this example. Members of the Follow set inconsistent with the PP would cause the Closure role to fire. This is not currently implemented.

[4]We would appreciate counterexamples.

Binders

Binding nodes use a similar rule to the production competition one:

```
if (inhibition - support < window)
{ /* keep going */
  if (inhibition == 0 && support > criterion) then win;
  else continue;
}
else lose;
```

One minor difference from the production competition code is that the inhibition[5] doesn't include what this node sends out. The competitors of a binder are determined from the rule that the same constituent can't be assigned to more than one role, and one role can't be filled by more than one constituent. The inhibition is the maximum of the input from these competitors. "Support" is the sum of bottom-up and top-down input. Top-down input comes from the production evidence network, and reflects how well the **production** is doing. As more roles in a grammar production get filled, the binders to those roles get more feedback. Thus if a binder for another role in the production wins, this is communicated indirectly to the other binders through increased feedback.

Missing Details and Weaknesses

There are several details that have been suppressed in this exposition; most of these stem from the use of a fixed network: Mechanisms had to be implemented to allow role recognizers to select constituent recognizers from a pool of them; the word sense buffer is of fixed length, although a similar recruiting mechanism might work here too; and just the existence of copies of constituent recognizers with identical control structures is not particularly palatable. However, recent advances in connectionist tools make the future look brighter. McClelland (1985) has developed a system called Connection Information Distribution (CID) which allows the storage of connection information in one central network (a *knowledge source*) to be "loaded" into a programmable buffer (like a Hearsay blackboard, Lesser & Erman, 1977) as needed. While the mapping of our system into the CID framework is not immediately obvious, the idea holds promise for avoiding many of the "fixed network" uglinesses.

Also missing here is a detailed description of all of the unit functions and the parameters used (e.g., connection weights and unit thresholds). To get this model to work, some "weight twiddling" was necessary. This unsavory practice is necessitated in part by the unconstrained formalism used, which at present is not amenable to formal analysis. Use of a more constrained formalism can lead to better methods for determining such parameters; see (Selman & Hirst, 1985). Their model also uses pure context-free grammars, avoiding the problems of within-production context sensitivity that forced us to use more copies of recognizers than their model.

Finally, we don't have a convincing demonstration that the networks generated by our program will always converge to a single coherent global interpretation. This points to the need to either (1) convert to a analyzable formalism such as Boltmann machines (Hinton & Sejnowski, 1983) or (2) develop more powerful analytical tools. For the present, we will have to be content with empirical demonstrations.

4. An Example Run

This system was implemented on a VAX/750 running Franz Lisp and C; the network builder was coded in Lisp and fed commands to the ISCON simulator which actually built the network. This in turn was "compiled" into a representation suitable for a much faster network simulator written in C. The actual network for the simple example we will present contained several hundred nodes and over a thousand connections. Hence, we will only give a high level description of the network's behavior.

Given the sentence *he cut the roll*, there are two cases of syntactic ambiguity, *cut* and *roll*.

---

[5]Here we are using the absolute value of the inhibition, which is usually negative.

There are no interesting closure problems, as any difference between this and *he cut*[6] can be handled by using "period" as a lexical item. We simulate reading this sentence by activating each lexical item every 30 steps of the simulation. These are then "read in" to the word sense buffer described earlier, and after 7 more steps, the syntactic class and meaning nodes are activated (these and the other nodes in the buffer accumulate activation slowly). About the time that PRO is activated in the buffer, an NP is expected by the Subject role of the S production. Since "he" is unambiguous, PRO has no competitors and gets highly active quickly. It then rapidly becomes bound to the Head role in the first NP (by iteration 16) since there is no competition for it.

The network then expects a VP, and the two productions for a VP (with and without a Direct Object) set up expectations for a VERB. When "cut"'s features (NOUN and VERB) become activated at clock step 38, they inhibit one another, driving each other below threshold. Since this cuts off the inhibition, they rise up again, like flickering bits, but now feedback from the binder for the VERB to the Main Verb role in the VP gives extra support to the VERB node, allowing it to remain above threshold while NOUN doesn't, resulting in a win for VERB in the second buffer position. Thus, "cut" has been disambiguated as a verb. By a few steps later, the node corresponding to the "meaning" of "cut" as a noun also loses.

After this is propagated up the network, expectations are set up for either a "period" or a Direct Object which can be filled by an NP. The Direct Object role recognizer selects the NP2 recognizer (the next available NP recognizer) which sets up expectations for either a DET, NOUN or PRO. After "the" has been processed, the NP recognizer is only expecting a NOUN, and so when "roll" comes in, it is quickly disambiguated. Eventually the production corresponding to a VP with a Direct Object wins, and the resulting stable coalition represents this parse with the appropriate binding nodes in a highly active state.

Thus our parser has disambiguated the two ambiguous lexical items on the basis of their "fit" into the developing parse tree. Anything incompatible with the expectations developed at the "frontier" of the developing tree was quickly extinguished. If there had been more structural ambiguity, lexical items compatible with either structure would have remained ambiguous until some production won over another. While this may seem implausible, it is compatible with the results of Hudson & Tanenhaus (1984).

5. Conclusions

We have presented a parsing model which has the following features:

1) completely distributed
2) massively parallel
3) automatically generated from a grammar
4) does not use symbol passing
5) matches neurolinguistic data.
6) has a "natural" explanation of Minimal Attachment.

Acknowledgements

We would like to acknowledge helpful discussions with James Allen, Gary Dell, Mike Tanenhaus, and Mark Fanty. The C simulator was written by Sumit Bandopadyay and Mark Fanty. This work was supported under DARPA grant NOO 14-82-K-0913 and NSF grant IST-8208571.

References

Aho, A. & Ullman, J. *Principles of Compiler Design*, Addison-Wesley, Reading, Mass., 1978.

Cook, W.A. Case grammar: The development of the matrix model (1970-1978). Georgetown University Press, Washington, D.C., 1979.

---

[6]Recall that we haven't implemented features other than syntactic class. In particular, there are no verb subcategorization features, so if we wanted to represent that *cut* is different from *left*, we would need to encode them as different syntactic classes in the current parser. Since we haven't, *he cut* is acceptable to our parser.

Cottrell, G. W. A model of lexical access of ambiguous words. In *Proceedings of the National Conference on Artificial Intelligence*, Austin, Texas, August, 1984.

Cottrell, G. W. A connectionist approach to word sense disambiguation. Unpublished Ph.D. Thesis, Dept. of Computer Science, University of Rochester, May 1985.

Cottrell, G. W. and Small S. A connectionist scheme for modelling word sense disambiguation. *Cognition and Brain Theory*, 1983, *6*, 89-120.

Feldman, Jerome A. and Dana Ballard. Connectionist Models and their Properties. *Cognitive Science*, 1982, *6* 205-254.

Fillmore, C.J. The case for case. In Bach and Harms (Eds.), *Universals in Linguistic Theory*. Holt, Rinehart and Winston, 1968.

Frazier, Lyn. On comprehending sentences: Syntactic parsing strategies. Ph.D. Thesis, University of Connecticut, 1978.

Gigley, H. M. Neurolinguistically Constrained Simulation of Sentence Comprehension: Integrating Artificial Intelligence and Brain Theory. Ph.D. Thesis, Department of Computer and Information Science, University of Massachusetts, September 1982.

Hinton, G.E. and T. Sejnowski. Analyzing cooperative computation. In *Proceedings of the Fifth Annual Cognitive Science Society Conference*, Rochester, N.Y., May 1983.

Hudson, S. & Tanenhaus, M. Ambiguity resolution in the absence of contextual bias. In *Proceedings of the Sixth Annual Conference of the Cognitive Science Society*, Boulder, Colorado, June, 1984.

Lesser, V.R., and Erman, L. D. A retrospective view of the Hearsay-II architecture. *Proceedings of the Fifth International Joint Conference on Artificial Intelligence*, 1977.

Linebarger, M.C., Schwartz, M.C., and Saffran, E.M. Sensitivity to grammatical structure in so-called agrammatic aphasics. *Cognition*, 1983, *13*, 361-392.

McClelland, J. L. Putting knowledge in its place: A scheme for programming parallel processing structures on the fly. *Cognitive Science, 9*, 113-146, 1985.

McClelland, James L. and David E. Rumelhart. An interactive activation model of the effect of context in perception: Part I, An account of basic findings. *Psych. Review, 88*,

Pollack, Jordan and Waltz, David. Natural language processing using spreading activation and lateral inhibition. In *Proceedings of the Fourth Annual Conference of the Cognitive Science Society*, Ann Arbor, Michigan, August, 1982.

Pollack, Jordan and Waltz, David. Massively parallel parsing: A strongly interactive model of natural language interpretation. *Cognitive Science, 9*, 51-74, 1985.

Rayner, K. Carlson, M. and Frazier, L. The interaction of syntax and semantics during sentence processing: Eye movements in the analysis of semantically biased sentences. *Journal of Verbal Learning and Verbal Behavior, 22*, 358-374, 1983.

Seidenberg, M. S., Tanenhaus M., Leiman, J. and Bienkowski, M. Automatic access of the meanings of ambiguous words in context: Some limitations of knowledge-based processing. *Cognitive Psychology*, 1982, *14*, 489-537.

Selman, Bart and Hirst, Graeme. A rule-based connectionist parsing system. In *Proceedings of the Seventh Annual Conference of the Cognitive Science Society*, Irvine, California, 1985.

Seidenberg, M. S., Tanenhaus M., Leiman, J. and Bienkowski, M. Automatic access of the meanings of ambiguous words in context: Some limitations of knowledge-based processing. *Cognitive Psychology*, 1982, *14*, 489-537.

Small, S. L., Cottrell, G.W., and Shastri L. Toward Connectionist Parsing. *Proceedings of the National Conference on Artificial Intelligence*, Piitsburgh, Pa., August 1982.

Small, S. L., Shastri L., Brucks M., Kaufman S., Cottrell, G., and Addanki, S. ISCON: An Interactive Simulator For Connectionist Networks. Technical Report 109, Department of Computer Science, University of Rochester, Dec. 1982.

A Rule-Based Connectionist Parsing System

Bart Selman and Graeme Hirst
Department of Computer Science
University of Toronto
Toronto, Canada M5S 1A4

Abstract

We describe a connectionist parsing scheme based on context-free grammar rules. In this scheme we use an updating rule similar to the one used in the Boltzmann machine (Fahlman, Hinton and Sejnowski 1983) and apply simulated annealing. We show that at low temperatures the time average of the visited states at thermal equilibrium represents the correct parse of the input sentence.

In contrast with previously proposed connectionist schemes for natural language processing, this scheme handles the traditionally sequential rule-based parsing in a general manner in the network. Another difference is the use of the computational scheme of the Boltzmann machine. This allows us to formulate general rules for the setting of weights and thresholds in our system.

The parsing scheme is built from a small set of *connectionist primitives* that represent the grammar rules. These primitives are linked together using pairs of computing units that behave like discrete switches. These units are used as binders between concepts represented in the network. They can be linked in such a way that individual rules can be selected from a collection of rules, and are very useful in the construction of connectionist schemes for any form of rule-based processing.

1. Introduction

Recently, several connectionist models for natural language understanding (NLU) have been proposed; for example, Waltz and Pollack (1984; Pollack and Waltz 1982) and Cottrell and Small (1983; Small, Cottrell, and Shastri 1982) give models for word-sense and syntactic disambiguation, and Reilly (1984) gives a scheme for anaphora resolution. The models are based on the deterministic connectionist scheme (McClelland and Rumelhart 1981; Feldman and Ballard 1982). A central aspect of these schemes is that they process the different sources of knowledge used in NLU, such as lexical and world knowledge in a highly integrated way; for example, the syntactic and semantic processing are combined.

A major limitation of these schemes is their very limited capability to handle tasks such as parsing and case filling which seem to require processing to be based on a set of rules. For example, Pollack and Waltz use the output of a conventional chart parser to generate a network for the syntactic parse of the sentence. This network only represents the parse tree (or trees, in case of syntactic ambiguity) of the particular input sentence. We propose a more general approach, namely a network that directly represents the grammar rules and is to be used for parsing of a large number of sentences. A similar approach could be used for other types of rule based-processing, like case filling.

This work was supported by a Government of Canada Award to the first author, and grants from the University of Toronto and the Natural Sciences and Engineering Research Council of Canada to the second author.

Our motivation behind this research is twofold. On one hand we believe that, at least part of the natural language understanding process can be handled by a connectionist architecture and that this form of integrated, parallel processing facilitates the parsing process. On the other hand, we believe that these schemes can only be of practical interest for NLU if they handle rule-based processing, like syntactic parsing, in a general and efficient way and are understood well enough to set the weights and thresholds correctly in large networks.

2. The model

2.1 Topology

We base our scheme on a context-free grammar, but this is not essential in our approach. The syntactic categories of the grammar are represented in a localist manner, that is, each syntactic category is represented by a unit in the network. As we will see this localist approach allows us to represent the grammar rules in a very straightforward manner, and consequently determines the set of non-zero weights (giving the topology of the network). The grammar rules will determine how these units are interconnected.

In the scheme we distinguish two layers. The *input layer* consists of a number of computing units representing the terminal symbols of the grammar. An input sentence will activate some subset of these units. Connected to this input layer is a network that represents the parse trees of all non-terminal strings in the language whose length is not greater than the number of units in the input layer. This network, called the *parsing layer*, is constructed from *connectionist primitives*, which represent the context-free grammar rules. Figure 1 gives two examples of such primitives. The activation of all units of a primitive corresponds to the use of the associated grammar rule in the parse. The number of units in the parsing layer depends on the particular context-free grammar rules and the number of input units. The different parse trees are not represented by completely disjoint sets of units, but share common substructures. This will keep the size of the network manageable.

We use intermediate computing units to link the primitives together. These units play the role of binders in the network and, therefore called *binder units*[1]. The computing units representing the terminals and variables of the grammar are called *main units*.

Figure 2 gives an example of the use of binder units. The four binder units are used to represent the fact that the main unit #0 is part of three grammar rules:

$$VP \rightarrow VP \;\; PP \qquad \text{(1a)}$$

$$VP \rightarrow verb \qquad \text{(1b)}$$

$$VP \rightarrow verb \;\; NP \;\; . \qquad \text{(1c)}$$

The binders are linked in such a way, using inhibitory and excitatory connections, that when the network reaches a stable state the active binders (those whose output equals +1) tell us which one of the three possible grammar rules is used in the parse of the input sentence to decompose the verb phrase represented by unit #0. So, if binder #1 stays active rule 1a is used in the parse, if binder #2 and #3 stay active rule 1b is used and if binder #2 and #4 stay active rule 1c is used.

2.2 Computational scheme

In this section we will consider the way in which the network finds the parse of a sentence. In the input layer of the network, the units are placed in *input groups*. Each group contains a unit for each terminal symbol of the grammar. The input groups are numbered; the n^{th} group is associated with the n^{th} word in the input sentence. Initially the computing units of both the input and the parsing layer of the network are inactive (their output is –1).

[1] Binder units with a similar function are used by Cottrell (1985) in his parsing system based on a deterministic connectionist model.

As a sentence comes in, each word of the sentence activates the computing unit(s) representing its associated syntactic category or categories. So the first word of the sentence activates one or more units (depending on the number of syntactic categories associated with the word) in input group #1, the second word one or more units in input unit #2, and so on. After receiving input data, the network starts the relaxation process. During this process, the outputs of the activated computation units in the input groups are fixed at +1, while the outputs of other units in the input layer are fixed at -1, so that the network can find the optimal match between the input data and the internal constraints representing the grammar rules; this match will represent the correct parse of the input.

In our model we use a variation on the computational scheme of the Boltzmann machine (Fahlman, Hinton and Sejnowski 1983; Hinton and Sejnowski 1983a, 1983b), and apply the simulated annealing scheme of Kirkpatrick et al. (1983) to find the optimal match between input data and internal constraints. Our scheme differs from the original in that we use -1 and +1 as output values of our computing units instead of 0 and +1. This facilitates the representation of symmetrical interdependency relations between hypotheses in the scheme; there exists a one-to-one mapping between this scheme and the original (Selman 1985).

The fact that this scheme searches for a global energy minimum and that at equilibrium the relative probability of a particular state of the system is given by its energy enables us to formulate general rules for the setting of the weights on the connections and the thresholds of the computing units.

We compute the average value of the output of each unit at the different temperatures used in the annealing scheme. In an example given below, we will see how these average values will change during cooling of the system; finally, at a temperature just above the freezing point of the system, the units with outputs close to +1 will represent the parse of the sentence. To find the temperature just above the freezing point of the network, we consider statistical data on the behavior of the network during simulated annealing.

2.3 The setting of weights and thresholds

The setting of weights and thresholds is probably the most difficult problem in the design of a connectionist scheme. The set of weights and thresholds represents the internal constraints and therefore the knowledge in the system. So far we have described how units are interconnected in our parsing scheme; that is the set of links with non-zero weights. Now we will discuss what values should be chosen for the weights on these links.

In the Boltzmann formalism, the behavior of the system during relaxation can be described as a search for a global minimum in the energy of the network

$$E = \sum_k E_{loc,k} \qquad (2)$$

In which

$$E_{loc,k} = (-1/2 \sum_j w_{kj} s_j + \theta_k) s_k \qquad (3)$$

is the contribution of the k^{th} unit with output value s_k and threshold θ_k to the energy, w_{kj} is the weight on the connection between the k^{th} and the j^{th} unit (we assume symmetrical connections), and the summation in equations (2) and (3) is over all units in the network.

Given the fact that the network searches for a global energy minimum, we can, to a first approximation, analyze the behavior of the network by assuming that each unit and its direct neighbors will chose output values such that $E_{loc,k}$ becomes minimal. However this method gives only a rough approximation of the actual behavior, because minimizing E_{loc} for one particular unit often conflicts with minimizing E_{loc} of other units. To get a better insight in the behavior of the system we therefore consider the contribution to the global energy of a

small groups of units.[1] Because of the homogeneous structure of our network we only have to consider a limited number of cases. As an example we will consider the setting of the weights on the excitatory links.

Figure 3 shows some excitatory links in a typical configuration. The network represents two grammar rules:

$$VP \rightarrow verb \qquad (4a)$$
$$VP \rightarrow verb\ \ NP \qquad (4b)$$

Rule 4a is represented by the units 0, 1, and 3; rule 4.4b by the units 0, 2, 4, and 5. During the relaxation process our network has to decide between rule 4a and rule 4b or neither of them. There is no a priori preference for one rule over the other. Therefore, because unit #2 is connected to two other units representing the left-hand side of grammar rule 4b and unit #1 is connected to only one unit representing the left-hand side of grammar rule 4a we have to make the weight on the link between units #1 and #3 twice as strong as the links between units #2 and #4 and between units #2 and #5. (This can be easily generalized for grammar rules with more symbols; one choses the weights such that the sum of the inputs at the binder units is equal for all grammar rules.) So we choose $w_{2,4}$ and $w_{2,5}$ equal to some positive constant and we set $w_{1,3}$ to twice this constant. We set this constant to 1.0. One should note that the absolute value of the constant is irrelevant. This value is only going to to determine at what temperature in our simulated annealing scheme the system is going to freeze, but the temperature is only a formal parameter introduced to do simulated annealing and has no meaning in our final result.

For the use of a grammar rule in the parse, the presence of each symbol in the rule is equally important, and therefore we connect the units in a connectionist primitive representing a grammar rule with links of equal strength, so $w_{4,5}= 1.0$. And finally, because bottom-up and top-down parsing is completely integrated and of equal importance in our networks we choose $w_{0,1}=w_{0,2}= 2.0$.

Selman (1985) gives similar analyses that lead to rules for for setting of weights on inhibitory links and the thresholds of the units. Here is a summary of these rules:[2]

$weight_{excitatory\ link}$	+1.0	*in primitive with three units*	(5a)
	+2.0	*in primitive with two units*	(5b)
$weight_{inhibitory\ link}$	–3.0		(5c)
threshold	0.0	*main unit*	(5d)
	–2.0	*main unit in symmetrical environment*	(5e)
	+2.0	*binder unit.*	(5f)

A *main unit in a symmetrical environment* is a main unit only linked to pairs of binder units (that is connected to both binders) and at most one other binder unit.

Although the local analyses and symmetry considerations on which these rules are based won't guarantee the right global behavior, good simulation results of a network with weights set according to these rules show that apparently such a local analysis gives a reasonable estimate of the global behavior of the parsing network. This is most presumably a consequence of the highly homogeneous structure of our parsing scheme (the networks are built from a small number of primitives).

[1] Of course, for an exact analysis one would have to consider all possible states of the (total) network; this becomes clearly infeasible for networks with more than about 25 nodes.

[2] Often only the ratio of between the different weights and thresholds are relevant; Selman (1985) gives these rules in a more general form.

3. The design and testing of a network

To illustrate our model we will now consider an example. This network is based on the following context free grammar rules (taken from an example in Winograd 1983):

$$
\begin{array}{lll}
S \rightarrow NP\ VP & NP \rightarrow determiner\ NP2 & \\
S \rightarrow VP & NP \rightarrow NP2 & \\
VP \rightarrow verb & NP \rightarrow NP\ PP & (6) \\
VP \rightarrow verb\ NP & NP2 \rightarrow noun & \\
VP \rightarrow VP\ PP & NP2 \rightarrow adjective\ NP2 & \\
PP \rightarrow preposition\ NP & &
\end{array}
$$

We will represent five input groups; in a complete network each input group has a unit for all terminals of the grammar, however to make our example network less complex, we will not represent each terminal in each input group.

For the grammar rules in (6) we can construct connectionist primitives similar to those given in figure 1. To build the parsing layer upon the input layer, using these primitives, we consider the different possible ways in which the syntactic categories can be grouped according to the grammar rules, and design a network that represents those possibilities. One way we could have proceeded is by designing a set of networks, each representing the parse of one unique input sentence, linking all these networks to the input layer, and placing inhibitory connections between them. These inhibitory connections should guarantee that after the parsing network is given an input sentence, only the sub-network representing the parse of the input would remain active.

Apart from the question of whether the design of such a network is even feasible, there are two fundamental reasons why we did not take this approach. Firstly, many parse trees have common sub-structures. So one can save computing units by representing a common sub-structure by one set of units and linking that structure, using binder units, to the different parse trees represented in the network. Secondly, main units represent general concepts, such as 'noun phrase'. It is unlikely that in the human brain (connectionist models are to a certain extent modeled after the brain) these concepts are represented at many different places. Therefore, instead of representing the same general concept at many places in the network, we try to limit the number of main units representing a concept.

So, instead of constructing a network from a set of separate networks, each representing the parse of a sentence, we take an approach in which we try to share common syntactic structures between parse trees and minimize the number of main units. Following these guidelines we can construct from the input layer, using the connectionist primitives, a network like the one given in figure 4.[1] The weights and thresholds in this scheme are set according to (5).

To demonstrate the parsing capability of our network we consider the input sentence[2]

$$noun\ \ verb\ \ preposition\ \ determiner\ \ noun\,, \qquad (7)$$

exemplified by "*John ran down the hill*". For this sentence we ran a simulation of the parallel network on a serial machine using a simulated annealing scheme. To apply this scheme, one has to choose a descending sequence of temperatures such that the system has a reasonable chance of finding the state with the global energy minimum. Therefore one starts at a high temperature and first cools rapidly; once the system approaches the freezing point (the point

[1] Some simple input sentences show that the multiple units for NP's, NP2's, and PP's are necessary; however, one could further minimize the number of VP's. However, this results in a network where the connectionist primitives are less visible, and which is therefore harder to understand.

[2] The network has been successfully tested for a number of input sentences, including some cases of syntactic ambiguity (in such cases more than one unit is activated within an input group), in which no semantic knowledge is necessary to resolve the ambiguity (Selman 1985).

at which it settles down in a state with a local or a global energy minimum; in this state the temperature is too low to escape from this minimum) one should cool very slowly. As we will see, we don't have to freeze the system completely; the right parse of the input sentence is found at a temperature just above freezing.

At each temperature above the freezing point one has to take sufficient computation steps to allow the system to reach equilibrium at that temperature.

To be able to choose the sequence of temperatures and the number of computation steps we did some test runs with the network. An appropriate sequence of temperatures was determined by considering the number of changes in the output value of each computing unit and the energy distribution of the system at each temperature. Based on these indicators we choose a sequence of temperatures starting at $T = 10000$ (to randomize the system), followed by $T = 4.0$, $T = 2.0$, and then in steps of 0.2 down to $T = 0.6$.

To estimate the required number of computation steps at each temperature, we considered the results of a sequence of simulations in which this number was slowly increased. When the average output values of the units become independent of the number of computation steps we assume that enough computation steps have been taken to scan the energy distribution of the system at equilibrium. Two thousand computation steps (i.e. 2000 updates of each unit) per temperature appeared to be sufficient. It should be noted that we did not try to minimize the number of temperatures and the number of steps per temperature.

Figure 5 shows the annealing process for sentence (7); we give six temperatures. Each panel shows the average output value of each computing unit at the temperature given below the panel. The numbers 0 to 44 are the numbers of the computing units, the vertical position indicates their average output value on the interval [-1,+1]. At $T \approx 1.0$ the system freezes; below this temperature the system stays in one state. Comparison of these results with the parse tree of this sentence given in figure 6 shows that the time average of the outputs of the units, at a low non-zero temperature, corresponds to the correct parse of the input sentence if one chooses the units with outputs close to +1 as being part of the parse tree. At low temperatures there is a clear distinction between units with output close to +1 and the other units, as can be seen in figure 5.

Although the set of average output values of the units in this section does not reveal any significant differences between the system in a frozen state or just above the freezing point; more information about the parse can be obtained at a temperature just above the freezing point, as we will see in the next section.

4. Changing weights and thresholds

The weights in the example network given above were set in accordance with the rules given in (5). Because the setting of the weights and thresholds is an important issue in connectionist models, we will now consider what happens if we change some of them. We use again the example network given in figure 4 and, input sentence (7).

First we set the threshold of main unit #32 equal to zero; originally this threshold was set to -2.0, following rule (5e). The simulation results show that in the frozen state the system gives the correct parse, except for the main nodes #18 and #32 and the binder #21; that is the average output values of those units are -1.0. However the average of the output values of these units at a temperature just above freezing are almost equal to zero. This means that at that temperature these units are part of the parse of the sentence for approximately 50% of the time (all other average output values are close to -1.0 and +1.0, consistent with the correct parse). This result can be explained as follows. Just above the freezing point the system jumps between two states, namely:

- state **a**, all units of the correct parse are active; and
- state **b**, same as state **a**, except for the units 18, 21, and 32.

States **a** and **b** have approximately the same low energy; however to jump between these states the system has to visit a state with a higher energy. At a temperature above the freezing temperature there is enough thermal energy to visit the intermediate state with a higher energy; in other words the system has a reasonable chance to visit in the intermediate state compared to the chance to be in the lower energy states **a**, and **b**. Therefore, the system will jump between state **a** and **b** and the average output value of the units 18, 21, and 32 will be around zero. However, when the temperature is lowered the system freezes, that is it will settle in one of the states **a** or **b**, and there is not enough thermal energy to jump to the other low-energy state.

This example clearly demonstrates that the average output values of the units give more relevant information when determined just above the freezing point of the system than at or below that point. It also demonstrates how the Boltzmann mechanism not only finds a global minimum in the energy, but just above the freezing point the system jumps between a number of states with energies close or equal to the global energy minimum. This is an important advantage over a deterministic scheme. Even in case such a scheme manages to find one of the states **a** or **b**, it is very unlikely that a deterministic scheme, just before finding **a** or **b**, would pass through the other state with minimal energy. This example also shows that if we don't follow all the rules given in (5), the system does not behave as well as when we do; however the model still comes up with a result close to the correct parse.

We will now consider what happens if we increase both the strength of the inhibitory links between binder units and the thresholds of these units. We choose a weight of -20.0 on the inhibitory links and thresholds of $+19.0$. These values are in accordance with the general rule for setting the thresholds on binder units and the weights on inhibitory links between them. In this case we don't find any significant differences between the simulation results with this new choice of weights and thresholds and those using the original values.

This is an interesting result, because with this choice of weights it becomes extremely unlikely, at low temperatures, to find a pair of binder units with both outputs equal to $+1.0$. Such a pair would give a large positive contribution to the energy; see equation (3). Therefore, the pairs are functioning as three-state switches with at most one unit with output equal to $+1.0$. This is useful during the search for a correct parse (or global energy minimum), because a pair with both outputs equal to $+1$ corresponds to the obviously incorrect situation in which two grammar rules are applied at the same time to decompose a syntactic category.

In figure 2 we saw two pairs of binder units linked in such a way that they can choose the application of one specific grammar rule out of three. Using a similar approach one can design a network from pairs of binder units that can select one rule out of a large collection; such networks will be useful in general connectionist schemes for rule-based processing.

5. Conclusions

We have discussed how traditional typically sequential, rule-based processing like parsing can be done in a completely parallel manner. In the design of such connectionist schemes the use of pairs of intermediate units that function as binders between units that represent the different concepts appears to be very useful. One can choose the thresholds of the binder units and the weights on the inhibitory links in between them such that they function as three-state switches (both units on is a 'forbidden' state). These pairs linked together in a binary tree structure can be used to select one rule (for example, a grammar rule) out of a collection of alternatives. During the search for an optimal match between input data and the internal constraints in the network, the binder pairs select different rules to test whether they should be used. Interestingly, this bears a close resemblance to how a sequential processing scheme tries rule after rule; the advantage of the connectionist scheme is that many rules, each part of a different collection and represented in different parts of the network, can be applied in parallel, and also there is a complete integration of bottom-up and top-down processing.

We saw that the special properties of the computational scheme of the Boltzmann machine made it possible to set the weights and thresholds by analyzing the energy of small groups of units and some general symmetry considerations. Another useful aspect of the Boltzmann scheme is that the network at temperatures just above the freezing points visits a number of states with energies equal or close to the global energy minimum of the network. Such states will, in general, show only minor differences from the state of the network that represents the correct parse; this makes the network less dependent on the particular choice of weights and thresholds.

The next logical step in this research is the addition of a semantic component in our scheme, to extend the disambiguation capability. Such a model would incorporate rules for case filling.

References

COTTRELL, G.W. (1985). *A connectionist approach to word sense disambiguation.* Doctoral dissertation, Computer Science Department, University of Rochester, Rochester, NY 14627, April 1985.

COTTRELL, G.W. and SMALL, S.L. (1983). A connectionist scheme for modelling word sense disambiguation. *Cognition and Brain Theory*, **6**(1), 1983, 89-120.

FAHLMAN, S.E.; HINTON, G.E. and SEJNOWSKI, T.J. (1983). Massively parallel architectures for AI: NETL, Thistle, and Boltzmann machines. *Proceedings of the National Conference on Artificial Intelligence*, Washington, August 1983, 109-113.

FELDMAN, J.A. and BALLARD, D.H. (1982). Connectionist models and their properties. *Cognitive Science* **6**, 1982, 205-254.

HINTON, G.E. and SEJNOWSKI, T.J. (1983a). Analyzing cooperative computation. *Proceedings of the Fifth Annual Conference of the Cognitive Science Society*, Rochester, NY, May 1983.

HINTON, G.E. and SEJNOWSKI, T.J. (1983b). Optimal perceptual inference. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, Washington DC, June 1983.

KIRKPATRICK, S.; GELATT, C.D.Jr. and VECCHI, M.P. (1983). Optimization by simulated annealing. *Science*, **220**#4598, 1983, 671-680.

McCLELLAND, J.L. and RUMELHART, D.E. (1981). An interactive activation model of context effects in letter perception. Part1, An account of basic findings. *Psychological Review*, **88**, 1981, 375-407.

POLLACK, J.B. and WALTZ, D.L. (1982). Natural language processing using spreading activation and lateral inhibition. *Proceedings of the fourth Annual Conference of the Cognitive Science Society*, Ann Arbor, August 1982, 50-53.

REILLY, R.G. (1984). A connectionist model of some aspects of anaphor resolution. *Proceedings of the* 10^{th} International Conference on Computational Linguistics, Stanford, July 1984, 144-149.

SMALL, S.T.; COTTRELL, G. and SHASTRI, L. (1982). Towards Connectionist Parsing. *Proceedings of the National Conference on Artificial Intelligence*, Pittsburgh, August 1982, 247-250.

SELMAN, B. (1985). Rule-Based Processing in a Connectionist System for Natural Language Understanding. Technical Report CSRI-168, Computer Systems Research Group, University of Toronto, April 1985.

WALTZ, D.L. and POLLACK, J.B. (1984). Phenomenologically plausible parsing. *Proceedings of the National Conference on Artificial Intelligence*, Austin, Texas, U.S.A., August 1984, 335-339.

WINOGRAD, T. (1983). *Language as a cognitive process.*, Reading, MA: Addison-Wesley Publishing Company, 1983.

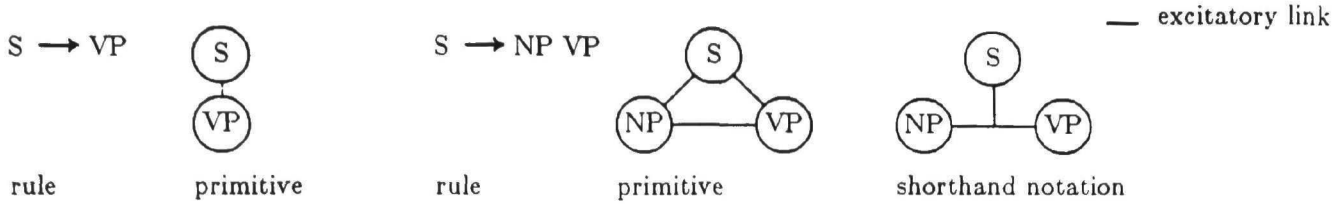


Figure 1 Two examples of connectionist primitives and their associated grammar rules.

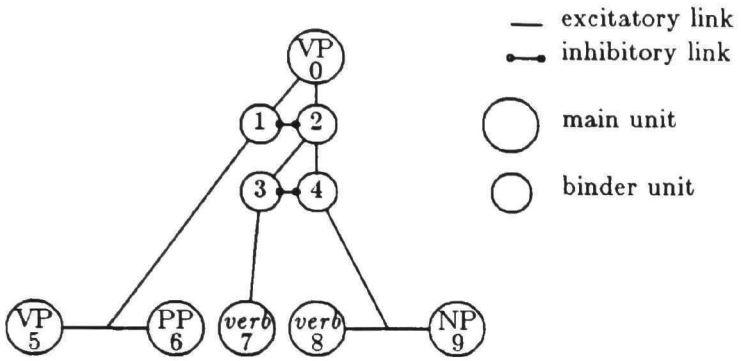


Figure 2 An example of the use of binder units (units 1, 2, 3 and 4).

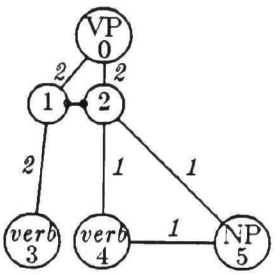


Figure 3 Some excitatory links in a typical configuration in the parsing network; their weights are given alongside the links.

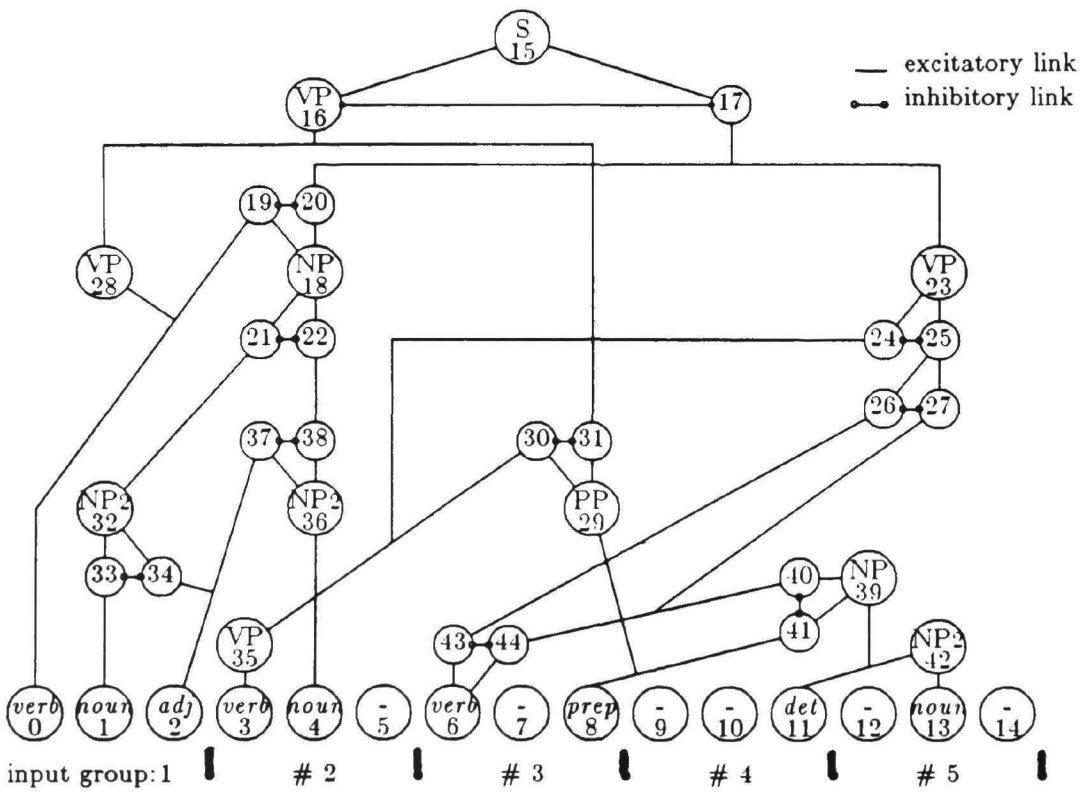


Figure 4 An example parsing network based on the grammar rules given in (6). Input group #1 consists of units 0, 1, and 2; group #2 consists of units 3, 4, and 5, and so forth.

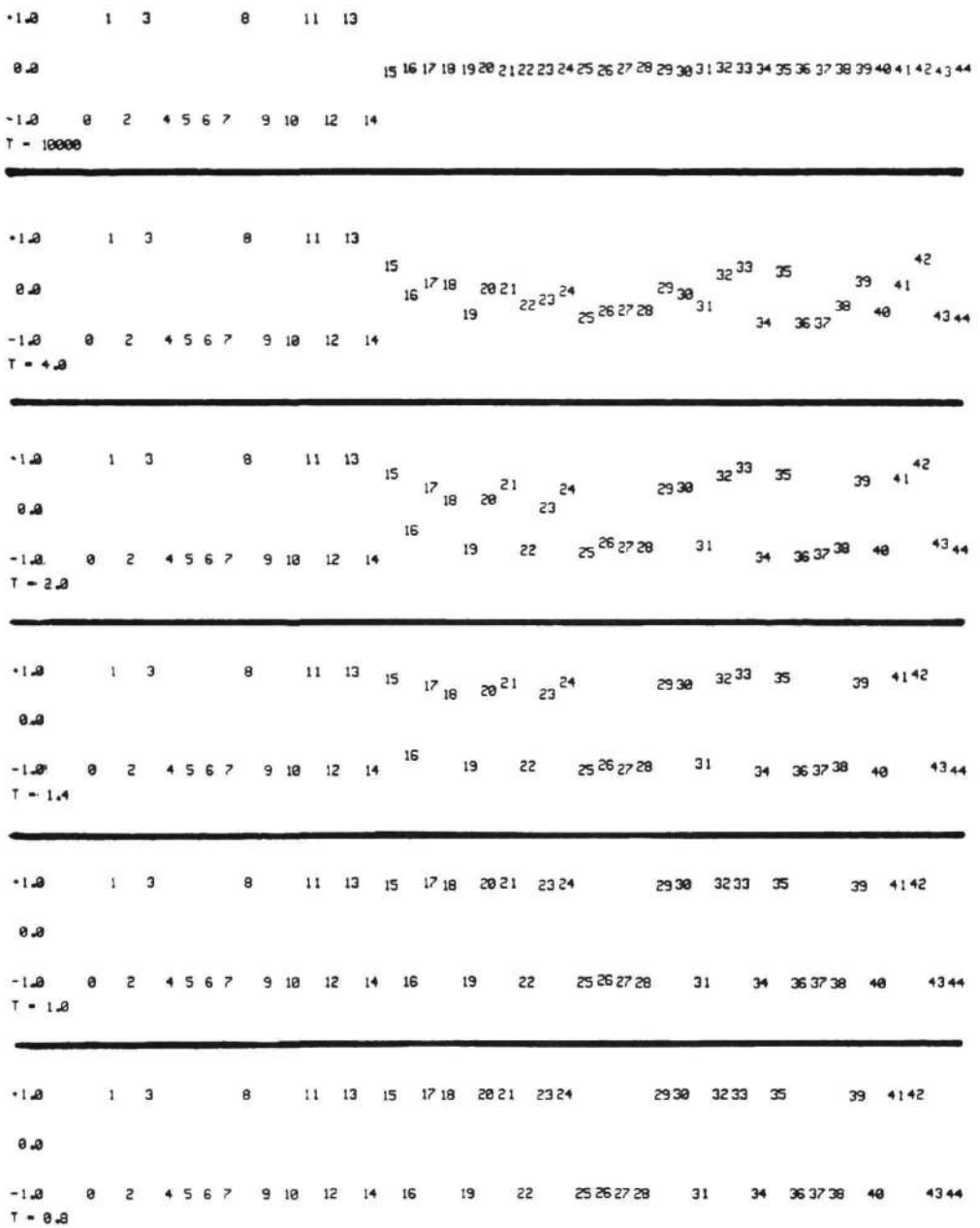


Figure 5 Average output values of computing units during simulated annealing. During the simulations the outputs of units 0 to 14 were fixed to represent the input sentence.

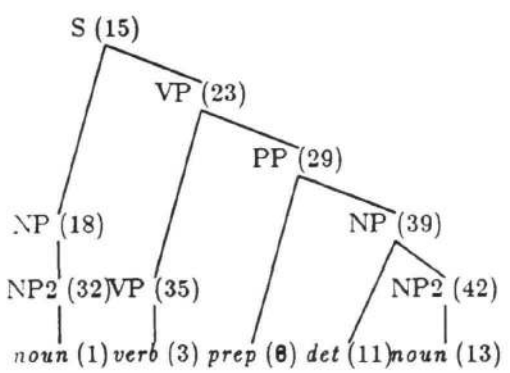


Figure 6 The parse tree of sentence (7). The numbers between parentheses are the numbers of the corresponding computing units in the parsing network.

Adaptive Planning: Refitting Old Plans to New Situations*

Richard Alterman

Division of Computer Science
University of California, Berkeley
Berkeley, CA. 94720

1. Introduction

This paper is about *adaptive planning*. The basic problem that adaptive planning addresses is the development of planning techniques for re-using old plans. The capability to re-use old plans suggests a way to avoid the problem of combinatorial explosion that is inherent in brute force planning methods. Moreover, it accounts for some of the flexibility of human planners: a planner than can re-use plans can plan about a wide range of phenomena, not so much because its depth of knowledge is consistent throughout that range, but because it can refit old plans to novel contexts.

A typical case of re-using plans is the situation when a planner is about to ride the NYC subway for the first time, and uses its experiences on BART (Bay Area Rapid Transit) to guide its planning activity. Consider the steps involved in riding BART (see figure 1). At the BART station I buy a ticket from a machine. Next, I feed the ticket into a second machine which opens a gate to let me into the terminal and then returns my ticket. Next I take the train. At the exit station I feed my ticket to another machine that keeps the ticket and then opens a gate to allow me to leave the station. Compare that to the steps involved in riding the NYC subway: buy a token from a teller, put the token into a turnstile and then enter, ride the train, and exit by pushing thru the exit turnstile. Given the BART Plan as depicted in Figure 1, there appears to be little in common between the two procedures. The problem with the BART Plan, as shown in figure 1, is that it represents this plan in isolation. In isolation the old plan does not provide enough information to refit it to the NYC subway situation. There is a great deal of background knowledge associated with the BART Plan that is not explicitly represented in the figure 1, but is needed in order to re-use the old plan. Without the background knowledge the BART Plan is practically useless in the construction of a plan for riding the NYC subway.

The key idea to understanding the adaptive planning approach to re-using old plans is:

> **Make explicit the content and organization of the background knowledge associated with the old plan.**

What is known about the BART Plan is not only the plan itself, but also that plan in relation to all the other planning knowledge that is available to the planner (see figure 2). Making explicit the background planning knowledge allows for a different kind of planner. Rather than planning by problem solving, it becomes possible to plan by *situation matching*. Rather than treating the old plan as a partial solution which is modified using weak methods, such as GPS (c.f. [1]), the old plan is used as a starting point from which the old and new situations are matched, and in the course of the matching a new plan is produced.

* This research was sponsored in part by the Defense Advance Research Projects Agency (DOD), Arpa Order No. 4031, Monitored by Naval Electronic System Command under Contract No. N00039-C-0235.

** I would like to thank Robert Wilensky for getting me interested in the problems of commonsense planning and the role of memory in planning and some good discussion as the ideas developed. I would also like to thank the other members of the BAIR (Berkeley Artificial Intelligence Research) group for constructive input.

An adaptive planner called PLEXUS (see dictionary for explanation of name) has been constructed. PLEXUS' knowledge-base takes the form of a network. Its basic strategy is to match the old plan situation against the new context. Where differences occur, changes are made to the old plan. PLEXUS achieves both the detection of differences and subsequent changes to the old plan by exploiting the net that surrounds the old plan.

2. Related Work

Early artificial intelligence research on planning and problem solving emphasized weak methods that apply to situations where extensive knowledge is not available. The basic idea was to produce plans by manipulating and combining many low level operations. Typical areas of application were chess playing programs, theorem proving, and robot problem solving. Although these kinds of methods are appropriate in knowledge poor contexts, they suffer from the problem of combinatorial explosion and are therefore less than ideal in the knowledge rich domains that frequently occur in common-sense planning situations.

Macrops [2] was the first attempt to deal with the problem of re-using old plans. Their application domain was robot problem solving. Solutions to old problems were generalized by substituting variables for the arguments of some of the operators. During planning, if some portion of an old plan, in its generalized form, achieved some goal or subgoal of the new plan it was re-used.

The major limitation on macrops was that the goals and situation of the old plan, except for where variables were substituted for constants, had to identically match the new situation. For most real world problems it is rarely the case that the old and new plan situations, or some subportion of them, are identical. More recent developments in the area of re-using plans have attempted to increase the flexibility of this original approach, in terms of both plan retrieval and usage. Kolodner & Simpson [3-5] and Hammond [6] have proposed techniques, based on Schank's theory of Dynamic Memory [7], for indexing old plans. Georgeff [8] has suggested guidelines for variable substitution under analogical conditions.

Carbonell has developed a theory of what he refers to analogical problem solving. He has suggested two approaches to employing analogies. His first approach [1, 9] applies a means-ends analysis to an old problem, gradually transforming it into a solution for a new problem. His second approach [10] dubbed derivational analogy, attempts to recreate the decision making process of relevant past problem solving situations, and apply that decision making process to the new problem situation.

Both of these approaches suffer from the problem of dealing with a plan in isolation. As alluded to in the introduction, without the content and organizational structure of the background knowledge the old plan will be practically useless in the construction of a new plan. Moreover there are more specific problems associated with each of these approaches. A problem with the first approach is the dependence of means-ends analysis on the creation of effective difference tables. A problem for the second approach is that in many cases a derivational history is not available either because it has been forgotten or because the plan was learned by rote and therefore a derivational history never existed.

3. The Plan Network

In adaptive planning, planning knowledge is represented in the form of a network. Associated with every plan in the network are a set of conditions which are used to determine if a plan, or a step in a plan, is appropriate for the current planning situation. Conditions include intentions, (sub)goals, pre-requisites, and expected outcomes. Adaptive planning is largely based on situation matching; the conditions provide a checklist for the planner to use in determining the applicability of an old plan, or step, to a new situation.

Each plan is represented as a sequence of steps, and, recursively, each step is a plan which, in principle, can be decomposed into a sequence of steps. For example, steps in the BART Plan include: buying a ticket, entering a BART station through a turnstile, riding the train, and exiting a BART station through a turnstile. Furthermore, there are substeps involved in buying a ticket: putting money into the machine and receiving a ticket in return.

There is an 'isa hierarchy' in the network, and as is the norm properties are inherited through the 'isa hierarchy'. For example, there is a general plan for buying tickets, and there are at least two specialized versions of that plan:

buying a ticket from a machine and buying a ticket from a teller. Moreover both of these descendants inherit from their common ancestors the normal goal for buying a ticket, which is to gain access to some service. Finally, as mentioned above, associated with each plan (or plan step) are a list of conditions. These can all be inherited as well.

4. The Adaptive Process

4.1. An overview

Having recalled a plan, PLEXUS must use the embedding context of the old plan in order to refit it to meet the demands of the new situation. PLEXUS refits the plan by using the conditions to match the old planning situation to the new one. In the event that differences occur, it is necessary for PLEXUS to modify the old plan to meet the new situation.

Some of the differences between old and new are *anticipated* at the outset, others grow out of the planner's *interaction* with the environment. For example, in the case where PLEXUS is adapting the BART Plan to the situation at the NYC subway, an anticipated difference is that the planner no longer expects to be purchasing a BART ticket, but instead a ticket for the NYC subway. A difference that occurs as a result of the planner's interaction with the environment is the realization that, for the NYC subway, one does not buy a ticket from a machine, but instead from a teller.

The basic procedure works as follows: PLEXUS adapts, in order, one step at a time, the steps of the old plan. For each step, it either anticipates or interacts with the environment to determine if the conditions associated with that step are met by the current environment. If the conditions of that step are met it adapts in a depth-first fashion the subsequences of that step. When PLEXUS bottoms out it moves on to the next step. If it can't apply a step of the old plan it tries to find an *abstraction* of that step which will work in the current context. If it succeeds in finding an abstraction, it next attempts to find a *specialization* of that step that will work in the current context. Whether in its abstraction phase or its specialization phase, PLEXUS is exploiting both the content and organization of the background knowledge associated with the old plan.

More specifically, the old plan is adapted to the new situation as follows:

1) Check the conditions of each step in the old plan. The conditions of a step include, via property inheritance, all of the conditions of the 'isa' ancestors of that step.
2) If the conditions are met then apply that step to the current situation.
3) If at least one of the conditions fails, try a more abstract version of that step by moving up the 'isa' hierarchy.
4) If a more abstract version of an old step works, try to specialize that step by moving back down the 'isa' hierarchy.
5) For each step, if the conditions of that step are met apply the same procedure in a depth-first fashion to the substeps of that step. When the procedure bottoms out move on to the next step.

In a sense PLEXUS' working hypothesis is that the new plan under construction and the old plan share a *significant* ancestor in the 'isa' hierarchy. By significant I mean that the steps of both the old plan and the new plan can be seen as specializations of the shared ancestor's steps. In fact, as the new plan is constructed, PLEXUS discovers the steps of the more abstract plan. These steps are composed of the successfully borrowed steps of the old plan (2) and the successful abstractions of steps from the old plan that failed to apply in the new situation (3,4).

Consider what happens when PLEXUS uses the BART Plan as a basis for constructing a plan for riding a NYC subway. Figure 2 shows a portion of net that is relevant to the problem of adapting the first step of the BART Plan to the problem of riding a NYC subway. The first step of the BART Plan is to buy a ticket from a machine, but in the new situation there is no machine from which to buy a ticket. By moving up the 'isa' hierarchy it finds a more abstract version of this step that will work in the current situation: through abstraction PLEXUS determines that because there is a place for buying a ticket the plan for buying a ticket will work. Next it moves back down the 'isa' hierarchy, specializing the plan to a plan for buying a ticket from a ticket office. In a similar fashion PLEXUS adapts the other three steps of the BART Plan. The following two sections will describe in greater detail what is involved in the processes of abstraction and specialization.

4.2. Abstraction - Some issues

This section will discuss several critical issues involved in abstraction. Associated with each of these issues is an answer that will fall out from the explicit introduction of the content and organization of the background knowledge.

First there is the issue of how to choose the correct abstraction. A given plan step can have any number of abstractions associated with it. Choosing the wrong abstraction can lead to the wrong action. For example, one abstraction of 'buying a ticket from a machine' is to 'use a machine' and a specialization of this is to 'use a candy machine' (see figure 2). In the NYC subway situation the planner needs to find the right abstraction of 'buying a BART ticket' else it may substitute 'buying a candy bar from a machine' for 'buying a ticket from a machine'. The planner can avoid this problem by ascending the 'isa' hierarchy that maintains the purpose of the step in the plan that is being refitted. In this case that means moving up the 'isa' hierarchy towards 'gain access'.

A second issue concerns knowing when to look for an alternate version of a step. For example, suppose the planner is trying to adapt its BART Plan to the Washington D.C. subway system. Like BART on the D.C. metro tickets are bought from a machine. Suppose the planner tries to buy a ticket from a machine but its dollar bill is rejected because it is too crumpled, the planner should not abandon the step to buy a ticket from a machine. By making the background knowledge explicit it becomes apparent why the step is not abandoned (see figure 2). The condition that is failing is not associated with 'buying a ticket from a machine', but instead with one of its substeps 'insert dollar bill'. The point is that the problem is not with the step but with the substep and it is the substep that needs to be refitted.

A third issue concerns the problem of when a planner should stop abstracting. In the NYC subway situation, when the planner discovers there is no ticket machine, the right abstraction to move to is 'buy a ticket'. In the case where the planner is trying to ride BART and the problem is s/he does not have any money, the planner needs to move above 'pay for access' to the abstraction 'gain access', thus allowing for the specialization 'break in' (see figure 2). Again the explicit introduction of the content and organization of the background knowledge allows for a simple solution: the planner needs to move up the 'isa' hierarchy to a position above the condition that is failing. In the case of the NYC subway situation the failing condition is that there exists no ticket machine, which is associated with 'buying a ticket from a machine', but not 'buying a ticket'. In the case of the BART situation the failing condition is associated with 'paying for access' but not 'gain access'. In either case the solution is to move to an abstraction that has no failing condition, but has a son with one.

A fourth way in which the structure of the background knowledge aids adaptation is that it partially orders alternate versions of a failed step. For example, 'gain access' is a more abstract version of 'buying a ticket from a machine' then is 'buying a ticket'. Consequently, in the NYC subway situation 'buying a ticket from a teller' will be available as an alternate plan before 'breaking in'.

5. Specialization - Some Issues

Where abstraction is initiated by the expectations formed from the old plan and driven by the planner's observations, specialization is not necessarily initiated by expectations. Again consider the case of adapting the BART Plan to the NYC subway situation. The expectation formed from the old plan is that there will be a machine from which to buy a BART ticket. This expectation is immediately amended, due to anticipated differences, to an expectation that there will be a machine from which to buy a NYC subway ticket. So the planner looks for a ticket machine, none exists, and therefore it continues abstraction. For present concerns, the key point is that when the planner observes its environment it is looking for a particular object, in this case a ticket machine. Now contrast that to the situation that confronts the planner when it attempts to form a specialization of the abstraction it has just determined. In the case of buying a subway ticket, through the process of abstraction, the planner determines that it still wants to buy a ticket, but there is no machine from which to buy it. The point is that specialization is not necessarily initiated by an expectation generated from the planner's memory. It is entirely possible that the planner looks around and notices a ticket booth, which thereby initiates the specialization of the plan 'to buy a ticket' to a plan 'to buy a ticket from a ticket teller'. On the other hand, it is equally possible that the planner first forms an expectation that a ticket booth might

exist and then it looks for it.

Such considerations should suggest that the movement towards specialization is a more complicated blend of expectation and observation than in the case of the movement towards abstraction. Rather than applying an absolute rule for initializing specialization by one or the other means, it appears that the planner must deal with specialization on a case by case basis. However, it is possible to provide a criteria for choosing one method of initiation over the other. For example, it could be the case that an alternative is strongly suggested by the type of failure that occurs. So if a planner is trying to get change from a change-machine and the dollar is returned because it is to crumpled, an alternate plan is strongly suggested by the nature of the failure (i.e. flatten out the dollar bill and try again). There are also cases where the expectation that a particular alternate plan might work is dictated not so much by the nature of the failure, but rather by a bias towards the normal specialization of a particular abstraction. Such I believe is the case of buying a ticket for the NYC subway. When the plan to buy a ticket from a machine fails and the planner determines that it still wants to buy a ticket, the planner is biased towards the normal plan for buying a ticket which is to buy ticket from a ticket teller. On the other hand, when no alternative stands out, the value of observing the surrounding environment for clues greatly increases. Overall the planner's strategy is to try a likely specialization if it is somehow suggested by circumstances, and if that fails observe the planning environment for clues.

6. Summary and Conclusions

Adaptive planning is a novel approach to planning. As opposed to planning by weak methods, its basic procedure is to recall similar planning situations and refit them to meet the demands of the current context. The basic idea is that the network of relationships in which a planning procedure is embedded can be exploited to refit an old plan to a new situation. The differences in the current planning situation and the context associated with the old plan guide the adaptive process. By exploiting the content and organization of the background knowledge it becomes possible to plan by situation matching.

7. References

1. Carbonell, J. G., A computation model of analogical problem solving, *IJCAI 7*, 1981.
2. Fikes, R., Hart, P. and Nilsson, N., Learning and Executing Generalized Robot Plans, *Artificial Intelligence Journal 3* (1972), 251-288.
3. Kolodner, J. L., Maintaining organization in a dynamic long-term memory, *Cognitive Science 7* (1983), 243-280.
4. Kolodner, J. L., Reconstructive memory a computer model, *Cognitive Science 7* (1983), 281-328.
5. Kolodner, J. L. and Simpson, R. L., Experience and problem solving: a framework, *Proceedings of the sixth annual conference of the cognitive science society*, 1984.
6. Hammond, K., Indexing and Causality: The organization of plans and strategies in memory., Yale Department of Computer Science Technical Report 351, 1985.
7. Schank, R. C., *Dynamic Memory*, Cambridge University Press, Cambridge, 1982.
8. Georgeff, M. P., Strategies in Heuristic Search, *Artificial Intelligence 20* (1983), 393-425.
9. Carbonell, J. G., Derivational analogy and its role in problem solving, *AAAI-83*, 1983, 64-69.
10. Carbonell, J. G., Learning by analogy: formulating and generalizing plans from past experience, in *Machine learning, and artificial intelligence approach*, Mitchell, M. C. (editor), Tioga Press, Palo Alto, 1983.

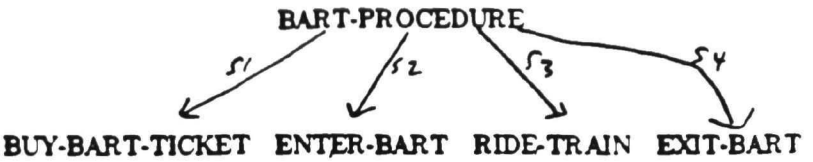


Figure 1. BART Plan in Isolation.

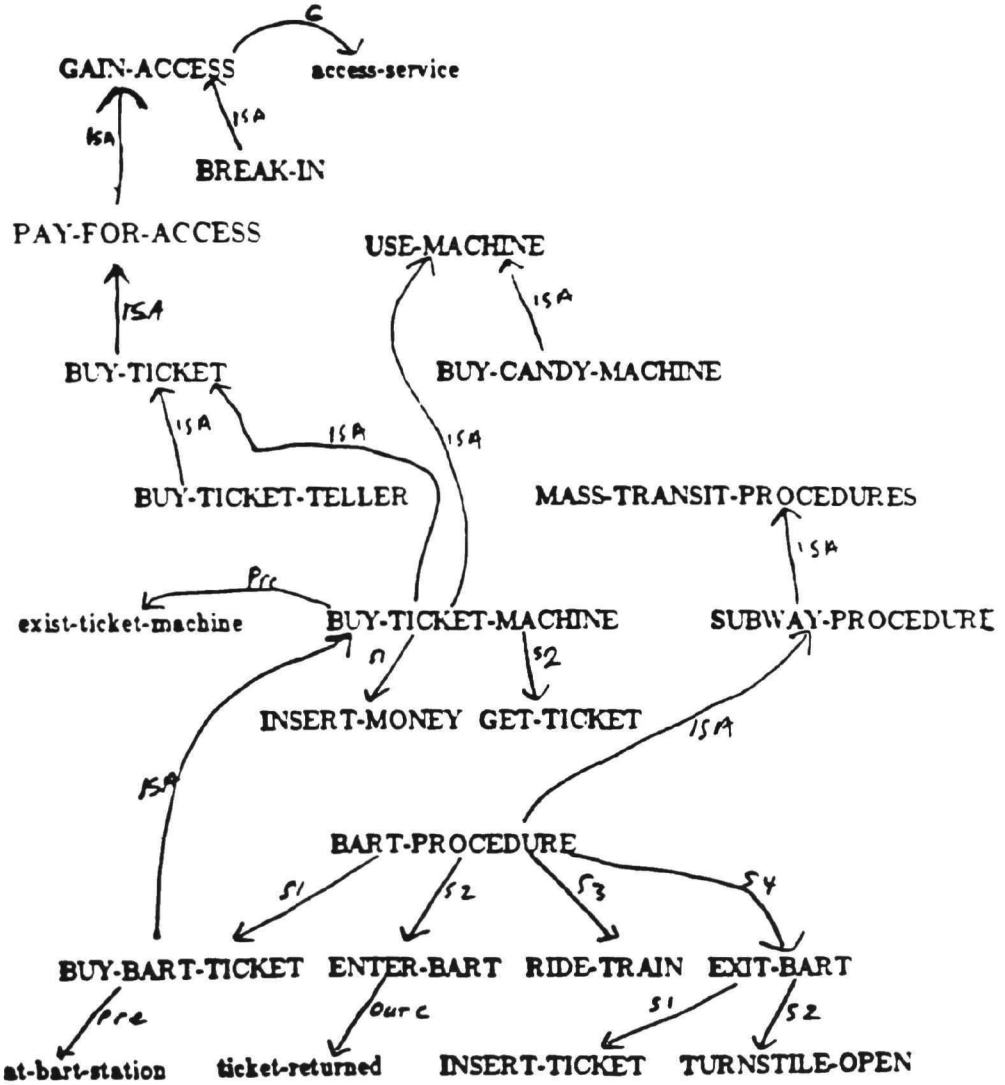


Figure 2. BART Plan with Background Knowledge Explicit.

Memory Representation and Retrieval for Editorial Comprehension[1]

Sergio J. Alvarado
Michael G. Dyer
Margot Flowers

Artificial Intelligence Laboratory
Computer Science Department, 3531 BH
University of California
Los Angeles, CA 90024

Abstract

In this paper, we present a process model of editorial comprehension, representation, and retrieval. Editorial comprehension involves building an argument graph organized by abstract argument strategies which are represented declaratively as *argument units*. Issues include: (a) organizing and indexing goals, plans, events, states, beliefs, and belief justifications instantiated during comprehension of editorial arguments; and (b) retrieving information from conceptual representations of editorial arguments. This process model of reasoning and argument comprehension is currently being implemented in OpEd (Alvarado et al., 1985a), a computer system that reads short politico-economic editorials and answers questions about them.

1. Introduction

Understanding a newspaper or magazine editorial requires building internal conceptual representations of editorial arguments that include goals, plans, events, states, beliefs, and belief justifications. This involves applying linguistic knowledge (e.g., knowledge of lexical items that refer to domain-specific objects, beliefs, causality, etc.) as well as recognizing and instantiating a disparate number of knowledge structures, such as: goals and plans (Schank and Abelson, 1977; Wilensky, 1983); ideologies (Carbonell, 1981); affective reactions (Dyer, 1983); belief and belief justifications (Alvarado et al., 1985a; Flowers, 1985; Flowers et al., 1982; Toulmin, 1979); and argument units (Alvarado et al., 1985a, b). For example, consider the following editorial excerpt from (Thurow, 1983):

ED-TARIFFS

> The Reagan administration argues that America does not need an industrial policy since all government has to do to guarantee economic success under capitalism is keep out of the way. Yet the Reagan administration has just ... increase[d] tariffs on large motorcycles from 4.4 percent to 49.4 percent.

Understanding ED-TARIFFS requires realizing that: (a) the Reagan administration's belief that no industrial policy is needed is supported by its laissez-faire belief; (b) tariffs are PROTECTION-PLANS used by the government to control imports; and (c) the Reagan administration's behavior is hypocritical since increasing tariffs go against a laissez-faire policy.

The memory representation of an editorial must also include indexing structures and access links which are created during editorial comprehension and later used by search and retrieval processes during question answering. For instance, answering the following question about ED-TARIFFS:

Q: Does America need an industrial policy?

A: The Reagan Administration believes that America does not need an industrial policy.

requires: (a) indexing structures from general and specific plans to their instantiations and access links between instantiated PLANS and their associated BELIEFS; and (b) retrieval functions that take PLANS and GOALS as input and retrieve appropriate BELIEFS.

In this paper, we discuss memory representation and memory retrieval techniques which are currently being implemented in OpEd (Opinions to/from the Editor) (Alvarado et al., 1985a), a computer program that reads and answers questions about short-politico economic editorials. OpEd is an integrated natural understanding system, i.e., it uses the same conceptual parser during editorial comprehension and question answering. OpEd's design is based on the demon-based conceptual parser implemented in BORIS (Dyer, 1983) and the question answering theory developed by Dyer and Lehnert (1982) as an extension of previous work by Lehnert (1978). The examples presented in this paper will be taken from the following editorial segment by Milton Friedman (1982):

[1]The work of these authors was partially supported by a grant from the W. M. Keck Foundation, with matching funds from the UCLA School of Engineering and Applied Science. Support was also provided through an IBM Faculty Development Award to the second author.

Recent protectionist measures by the Reagan administration have disappointed us ... [voluntary] limits on Japanese exports of automobiles ... are ... bad for the nation ... Far from saving jobs, the limitations on imports will cost jobs. If we import less, foreign countries will earn fewer dollars. They will have less to expend on U.S. exports. The result will be fewer jobs in export industries.

2. Conceptual Representation of an Editorial

The memory representation of an editorial forms a graph of conceptual constructs instantiated during editorial comprehension. Within this graph, instantiated constructs are connected by memory links that indicate how they relate to one another, i.e., the links indicate knowledge dependencies such as: causal dependencies, support/attack relationships, containment relationships, and indexing relationships.

Five major elements compose the conceptual graph of an editorial:

1) Instantiations of: domain-specific objects (e.g., nations), goals, plans, events, states, affective reactions, and participants in editorial arguments (e.g., the editorial writer and his implicit opponents).
2) Instantiations of beliefs of participants in editorial arguments.
3) An *argument graph* (Birnbaum, 1982; Flowers et al., 1982) which includes all instantiated beliefs, belief relationships, and belief justifications involving the different participants in editorial arguments.
4) Instantiations of *argument units* which organize abstract knowledge about reasoning and argumentation.
5) Indexing structures which allow search and retrieval processes access to the argument graph.

In this section, we focus on beliefs, argument graphs, and argument units.

2.1. Beliefs

Computer comprehension of editorials is based on the capability of recognizing beliefs, belief supports, and belief attacks. Current text understanding programs are capable of reading stories involving stereotypic situations, goal and planning situations, and complex human interactions (Cullingford, 1978; DeJong, 1979; Dyer, 1983; Lebowitz, 1980; Wilensky, 1983). However, those programs cannot read editorials since they lack basic mechanisms for: (a) understanding and keeping track of beliefs and belief justifications, and (b) using world knowledge during reasoning comprehension.

Abelson (1979) has pointed out that even though belief systems share common ground with knowledge systems, beliefs can be set apart from structured knowledge. Beliefs are not goals, plans, events, or states, but rather predications about these structures and their relationships (Abelson, 1973). For example, the following predications can be distinguished:

Evaluative: Free-trade economists believe that protectionist measures are bad.

Judgemental: Free-trade economists believe that the U.S. should not renew voluntary quotas on Japanese exports of automobiles.

Causal: Free-trade economists believe that voluntary limits on Japanese exports of automobiles will cost jobs in the U.S..

Expectational: American auto makers believe that the Reagan administration will renew the voluntary quotas on Japanese exports of automobiles.

Factual: The Reagan administration believes that American auto makers have already returned to profitability.

Beliefs about Beliefs (Wilks and Bien, 1983): Western countries believe that Eastern countries believe that Western countries will launch a nuclear attack.

How are beliefs represented in OpEd? We consider that each belief consists of: (a) the holder of the belief; (b) belief contents; and (c) links that indicate whether the belief attacks, supports, or is supported by other beliefs. For example, Friedman's belief that voluntary limits on imports will cost jobs is represented as:

```
(BELIEF BELIEVER (FRIEDMAN0)
        CONTENT  (PROTECTION-PLAN0 --thwart--> P-JOB0))
```

where FRIEDMAN0 refers to an instantiation of OpEd's knowledge about Friedman, PROTECTION-PLAN0 to an instantiation of protectionist plans, and P-JOB0 to an instantiation of the goal PRESERVE-JOB.

In general, contents of beliefs involve either: (a) a causal dependency; (b) a chain of causal dependencies; or (c) an evaluative component. Causal dependencies usually include intentional relationships between goals, plans, events, and states, such as: goal motivation, goal failure, goal achievement, goal suspension, event realization, plan intention, and plan enablement. These dependencies are represented by means of *intentional links* (I-links) (Dyer and Lehnert, 1982). Other non-intentional causal dependencies such as relationships between economic quantities (Riesbeck, 1983), are represented using a general causal link. For example, ED-JOBS contain the following causal dependencies:

Goal Failure: Executing the Reagan Administration's PROTECTION-PLAN (i.e., the limitations on imports) thwarts the goal PRESERVE-JOBS.

Goal Achievement: Executing the PROTECTION-PLAN will achieve the goal PRESERVE-JOBS.

Consequent State: Earning fewer dollars motivates having less resources to buy imports.

Beliefs which contain evaluative components (Abelson, 1979) refer to high level abstractions such as "X believes that Y is GOOD" and "X believes that Y is BAD" that indicate whether X leads to positive or negative outcomes (e.g., goal and expectation failures or achievements). We believe that "GOOD" and "BAD" are primitives which act as place holders, much like DO in Conceptual Dependency (CD) theory (Schank, 1973, 1975). For example, in the CD representation of "John killed Mary":

```
(LEAD-TO ANTECEDENT (DO ACTOR (JOHN))
         CONSEQUENT (STATE-CHANGE STATE (PHYS-STATE CHARACTER (MARY))
                                  FROM  ( > -10 )
                                  TO    ( < -10 ) ))
```

DO is a place holder for John's action which can later be filled if we hear the actual actions (e.g., "John strangled Mary to death"). Similarly, GOOD and BAD are place holders for unknown positive and negative outcomes. Thus, GOOD and BAD also trigger expectations for positive and negative outcomes.

2.2. Argument Graph

Understanding an editorial requires integrating explicit and implicit beliefs and belief justifications into an *argument graph* (Birnbaum, 1982; Flowers et al., 1982). Within this graph, beliefs are connected by links that indicate whether they support or attack one another. During text comprehension, every new belief or belief justification is integrated into the graph by using those links. The attack and support relationships are established from the application of the inference rules for: (a) recognizing beliefs and belief relationships, and (b) following belief justifications. For example, the following attack and support relationships are present in ED-JOBS:

Support Relationship between Beliefs: Friedman's general belief that the limitations on imports are bad[2] is supported by his specific belief that the limitations on imports will cost jobs in the U.S.

Supporting Cause-Effect Chain: Friedman's specific belief is supported by the cause-effect chain that indicates how a reduction in imports produces a reduction of jobs in export industries.

Attack Relationship between Beliefs: Friedman's specific belief attacks the Reagan administration's belief that the limitations will save jobs.

In general, support relationships are themselves supported by *warrants,* i.e., more basic beliefs which state that conclusions can be drawn from supporting evidences (Flowers et al., 1982; Toulmin, 1982). These warrants are recognized and instantiated by inference rules for recognizing beliefs and belief relationships. In addition, since warrants are just beliefs, they can themselves be attacked. For example, the support relationship between Friedman's general belief that the limitations are bad and his specific beliefs that the limitations on imports will cost jobs, is based on the following principle:

"IF a PLAN-P thwarts a PRESERVATION GOAL, THEN PLAN-P is BAD."

Beliefs of the form "X believes that C causes E" are usually supported by a belief whose contents represent an expansion of the cause-effect relationship. In this case, the warrant of the support relationship is provided by the following general rule:

"IF C causes E1 AND E1 causes E2 AND ... En causes E, THEN C causes E."

For example, Friedman's specific belief about the limitations on imports is supported by a cause-effect chain in ED-JOBS of how reductions in imports produce reductions in exports.

---

[2] OpEd infers Friedman's belief from the affect description "disappointed".

2.3. Argument Units

Argument units (AUs) are abstract reasoning structures which organize: (a) belief, goals, and plans; and (b) support and attack chains of reasoning and relationships in arguments. Seven argument unit have been initially identified: AU-ACTUAL-CAUSE, AU-OPPOSITE-EFFECT, AU-EXPECTATION-FAILURE, AU-HYPOCRISY, AU-ACTUAL-EFFECT, AU-EQUIVALENCE, and AU-RELEVANT-ISSUE.

Each argument unit can be cued by specific constructs which involve: (a) *argument connectives* such as "on the contrary", "far from", "but", and "yet", which signal opposition and expectation failures; and (b) goal, plan, and belief relationships. As a result, following an argument involves recognizing these constructs, accessing the specific conceptualizations they refer to, mapping from them into their appropriate argument unit, and triggering that argument unit's inference rules for recognizing belief, support and attack chains of reasoning and relationships. This recognition process relies on expectations generated after an argument connective is found. These expectations involve specific information about abstract goal, plan, and belief relationships that precede and follow each corresponding argument connective.

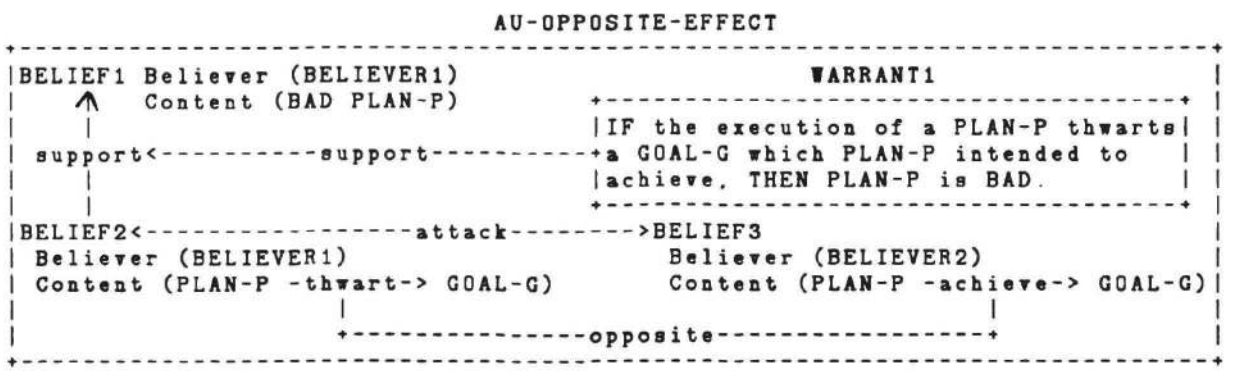
Instantiating argument units helps build argument graphs. For instance, consider AU-OPPOSITE-EFFECT. This argument unit embodies the following chain of reasoning:

> *Although Y believes that executing his PLAN P will achieve GOAL G, SELF does not believe this because SELF believes that executing P will thwart G. Therefore, SELF believes that P is bad.*

where SELF refers to the character who uses AU-OPPOSITE-EFFECT. This chain of reasoning contains a support relationship, a warrant of the support relationship, an attack relationship, and a declarative opposite relationship between two causal dependencies:

> *Support Relationship:* SELF's general belief that Y's PLAN P is bad is supported by SELF's specific belief that P thwarts GOAL G.
>
> *Warrant:* IF the execution of a PLAN P thwarts a GOAL G which P intended to achieve, THEN P is BAD.
>
> *Attack Relationship:* SELF's specific belief attacks Y's belief that Y's PLAN P achieves GOAL G.
>
> *Opposite Relationship:* Thwarting G by executing P is the opposite of achieving G by executing P.

These relationships are illustrated in the figure below.



Argument participants commonly convey AU-OPPOSITE-EFFECT using the following constructs:

1) *<Far from ACHIEVING GOAL-G, PLAN-P THWARTS GOAL-G>*
2) *<PLAN-P NOT-ACHIEVES GOAL-G. On the contrary, PLAN-P THWARTS GOAL-G.>*

where "far from" and "on the contrary" indicate the relation of opposition.

2.4. Building Editorial Memory

How does the comprehension process work and what does memory look like afterwards? Below follows a simplified trace of how OpEd processes ED-JOBS:

```
... measures ... disappointed us
    ===> PLAN execution --cause--> NEGATIVE-AFFECT
       ==infer==> B1: Friedman believes (measures are BAD)

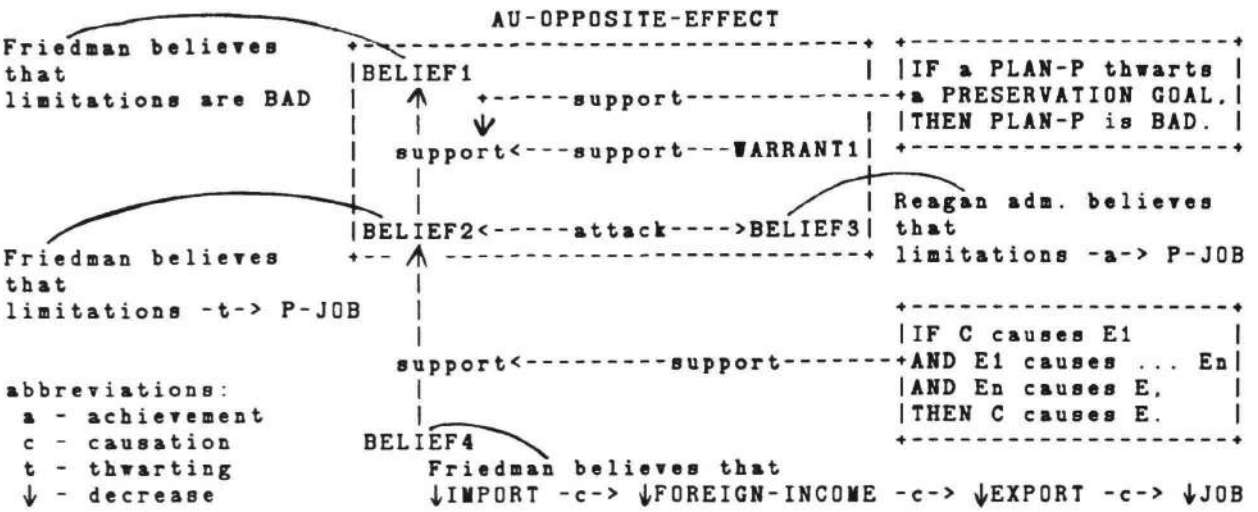
... limits ... bad for the nation
    ===> limits --cause--> GOAL FAILURE U.S.
       ==infer==> B2: Friedman believes (limits--cause--> GOAL FAILURE U.S.)
                  B2 supports B1
```

```
... Far from saving ... limitations cost ...
    ===> limits --achieve--> P-JOB     limits --thwart--> P-JOB
                    |                                  |
                    +----------opposite----------------+
       ==infer==> B2: Friedman believes (limits --thwart--> P-JOB U.S.)
        activate AU-OPPOSITE-EFFECT
        ==infer==> B3: Reagan adm. believes (limits --achieve--> P-JOB U.S.)

... import less ... earn fewer ... less to spend ... fewer jobs
    ===> CAUSE-EFFECT-CHAINO
         ↓import --cause-> ↓foreign-income --cause-> ↓export --cause-> ↓jobs

       ==infer==>  B4: Friedman believes (CAUSE-EFFECT-CHAINO)
                   B4 supports B3
```

The following diagram shows a simplified version of the memory representation currently constructed by OpEd while parsing ED-JOBS:



As the above diagram indicates, the bulk of the editorial memory is contained in the argument graph that depicts support/attack relationships between beliefs and cause-effect chains. Besides organizing several support/attack relationships, AU-OPPOSITE-EFFECT provide necessary inferences for: (a) recognizing and integrating into the argument graph the attack relationship; and (b) recognizing that the Reagan administration believes that the limitations will save jobs. In addition, AU-OPPOSITE-EFFECT captures the point of Friedman's argument and indexes the argument graph for subsequent retrieval.

3. Memory Retrieval

During question answering, the argument graph has to be accessed by search and retrieval processes. Thus, initial entry to the graph must be provided by structures that index beliefs and belief justifications. We have initially identified four indexing schemes based on how recall is done when answering questions about editorials: (1) indexing by containment (i.e., goal, plan, event, and state indexing); (2) indexing by affects; (3) indexing by argument participants; and (4) indexing by argument units.

(1) Indexing by Containment: Each instantiated goal, plan, event, and state indexes:

a) Beliefs and cause-effect chains which contain that goal, plan, event, or state. For example, the goal PRESERVE-JOB indexes both Friedman's specific belief and the Reagan administration's belief about the administration's PROTECTION-PLANs.

b) Structures used in representing that goal, plan, event, or state. For example, the goal PRESERVE-JOB indexes the Reagan Administration as the goal's holder.

(2) Indexing by Affects: Each instantiated affect indexes its associated belief. For example, Friedman's NEGATIVE-AFFECT indexes his belief that PROTECTION-PLANs are BAD.

(3) Indexing by Argument Participants: Each instantiated argument participant indexes:

a) The argument participant's *top beliefs*. A top belief is one that does not support other beliefs.

b) Argument units used by that participant. For example, OpEd accesses via Friedman the instantiation of AU-OPPOSITE-EFFECT.

(4) Indexing by Argument Units: Each instantiated argument unit indexes beliefs contained in its chains of reasoning and relationships. For example, AU-OPPOSITE-EFFECT indexes both Friedman's and the Reagan administration's beliefs about the administration's PROTECTION-PLANs.

Once an argument graph and indexing structures have been instantiated in episodic memory, OpEd demonstrates its comprehension by answering questions about the editorial. Selecting appropriate retrieval strategies depends upon parsing the question and analyzing the conceptual content into one of a number of conceptual question categories (Lehnert, 1978). Different question categories lead to different search and retrieval processes. These processes select indices according to the question's conceptual information. Once an index is selected, these processes will traverse access and memory links in order to locate an appropriate memory which will be retrieved.

Below we discuss a number of search and retrieval processes along with examples of their use during question answering about ED-JOBS. Answer generation in OpEd is by recursive descent through instantiated concepts, using patterns of generation associated with each uninstantiated knowledge structure. For example, the following is an actual answer currently generated by OpEd:

Q: Why have the limitations disappointed Friedman?

A: MILTON FRIEDMAN BELIEVES THAT VOLUNTARY EXPORT RESTRAINTS ON AUTOMOBILES FROM JAPAN NEGOTIATED BY THE REAGAN ADMINISTRATION WILL CAUSE THE LOSS OF JOBS IN THE U.S.

Notice the above answer is a detailed account of what OpEd knows about Friedman's belief. Thus, the answer includes information that humans generally omit, such as the fact that the limits were negotiated by Reagan. However, at this point we are not concerned with linguistic style in question answering. This is an area for future research.

3.1. Retrieval using Goal Indexing

Intended Plan Retrieval: Given an argument participant's goal, retrieve a belief which contains a plan of action to achieve that goal.

Q: How is the Reagan administration going to save jobs?

A: The Reagan administration believes that the limitations on imports will save jobs.

Argument Participant Retrieval: Given a goal, retrieve the argument participant who wants to achieve it.

Q: Who wants to save jobs?

A: The Reagan Administration.

3.2. Retrieval using Plan Indexing

Achieved and Thwarted Goal Retrieval: Given an argument participant's plan, retrieve beliefs which contain the effects of that plan's execution.

Q: What is the effect of the limitations on imports?

A: Friedman believes that the limitations on imports will cost jobs.
The Reagan administration believes that the limitations on imports will save jobs.

Argument Participant Retrieval: There are two cases:

1) Given a plan, retrieve the argument participant who is the actor of that plan.

   Q: Who will implement protectionist measures?

   A: The Reagan Administration.

2) Given a belief about a plan, retrieve the argument participant who holds that belief.

   Q: Who believes that the limitations on imports will save jobs?

   A: The Reagan administration.

Support Retrieval: Given a belief about a plan, retrieve the immediate support for that belief.

Q: Why does Friedman believe that the limitations on imports are bad?

A: Friedman believes that the limitations on imports will cost jobs in the U.S..

3.3. Retrieval using Event Indexing

Consequent State Retrieval: Given an event, retrieve a belief that contains a state or causal chain caused by that event.

Q: What will happen if the U.S. imports less?

A: Friedman believes that if the U.S. imports less, export profits of foreign countries will be reduced. Since this will reduce their total capital for importing, they will import less from the U.S. As a result, U.S. export industries will lose money and, therefore, they will lay off workers.

Plan Realization Retrieval: Given an event, retrieve a belief about a plan which realized that event.

Q: Why will the U.S. import less?

A: Friedman believes that the U.S. will import less because of the limitations on imports.

3.4. Retrieval using State Indexing

Causal Event Retrieval: Given a state, retrieve a belief that contains an event which causes that state.

Q: Why will export profits of foreign countries will be reduced?

A: Friedman believes that if the U.S. imports less, export profits of foreign countries will be reduced.

3.5. Retrieval using Affect Indexing

Affect Motivation Retrieval: Given an affect and an argument participant, retrieve its motivation:

Q: What has disappointed Friedman?

A: Voluntary export restraints on Japanese cars negotiated by the Reagan Administration.

Associated Belief Retrieval: Given an affect and its cause, retrieve the immediate support of its associated belief associated. If the support is not known, then retrieve the associated belief.

Q: Why have the limitations on imports disappointed Friedman?

A: Friedman believes that the limitations will cost jobs in the U.S..

3.6. Retrieval using Argument Participant Indexing

Top Belief Retrieval: Given an argument participant and a plan, retrieve the argument participant's top belief which contains that plan and its immediate support.

Q1: What does Friedman think about voluntary limits on Japanese cars?

A1: Friedman believes that the limitations on imports are bad because they will cost jobs.

Q2: What does the Reagan administration think about voluntary limits on Japanese cars?

A2: The Reagan administration believes that the limitations on imports will save jobs.

Argument Unit or Top Belief Retrieval: Given an argument participant, retrieve the argument units that contain his top beliefs. If he has not used any argument unit, retrieve his top beliefs.

Q: What is Friedman's argument?

A: That voluntary limits on Japanese cars are bad because they will cost jobs in the U.S.. Friedman believes that the Reagan Administration is wrong because the administration believes that voluntary limits on Japanese cars will save jobs in the U.S..

4. Comparison with other Work

In contrast to the *conceptual* process model presented here, Cohen (1983) has postulated a *structural* model for argument understanding. According to Cohen, understanding an argument requires building a tree where argument propositions are connected by a single evidence link. The root of the tree contains the major claim made in an argument. Relations between propositions are determined by using: (a) a *proposition analyzer* which produces a proposition from the input and integrates it into the tree built so far, (b) a *clue interpreter* which analyzes the role of special linguistic connectives (e.g., "as a result", "similarly", etc.), and (c) an *evidence oracle* which accesses a knowledge base and model of the speaker in order to determine whether an evidence relation exists between any two propositions.

Aside from the fact that Cohen's model was not implemented, it is theoretically limited since, in general, arguments do not comply with a coherent tree structure, but rather with a graph (Flowers et al. 1982). Furthermore, Cohen's *non-conceptual* tree structure does not show *how* explicit and implicit conceptualizations contained in a proposition relate to, and provide evidence for, conceptualizations contained in other propositions. Moreover, by using an *oracle*, the model avoids dealing with: (1) how lexical items are mapped from natural language into conceptual structures; (2) how world knowledge is represented and applied during the comprehension process; and (3) how abstract argument strategies are represented and applied. In contrast, OpEd understands complex editorial arguments by building a conceptual graph which captures interactions between goals, plans, events, states, beliefs, and argument

units. This conceptual graph results from recognizing and instantiating those knowledge structures along with causal, attack, and support relationships. In addition, this comprehension process also requires building indices which are subsequently used during question answering.

5. Future Work and Conclusions

We have presented memory organization and retrieval techniques for use by OpEd in understanding and answering questions about short politico-economic editorials. Currently, OpEd can read ED-JOBS and answer a number of questions about its conceptual content. Future questions that need to be addressed include:

* How is editorial memory organized after reading several editorials which present diverse opinions on a specific issue and how is memory retrieval performed in this case?
* How do different ideologies affect editorial comprehension, memory organization and retrieval?
* How are argument units used in order to generate arguments?

Five major points have been emphasized in this paper: (1) the internal conceptual representation of an editorial has several levels of complexity that include instantiations of conceptual structures which hold world knowledge and abstract knowledge about reasoning and argumentation; (2) a major portion of editorial memory is contained in an *argument graph;* (3) instantiating *argument units* helps build the *argument graph* by supplying inferences and argument strategy expectations; (4) search and retrieval processes use indexing structures to access *argument graphs;* (5) the points of an editorial are held by instantiated *argument units* containing top beliefs.

We believe that all arguments are composed of configurations of a fixed number of abstract *argument units.* Thus, we see the process of argument comprehension and generation fundamentally as one of accessing and instantiating these units. With OpEd we intend to explore and test this process model of reasoning and argument comprehension.

References

Abelson, R. P. (1973). The Structure of Belief Systems. In R. C. Schank and K. M. Colby (Eds.), *Computer Models of Thought and Language*. San Francisco, CA: Freeman.

Abelson, R. P. (1979). Differences between Beliefs and Knowledge Systems. *Cognitive Science*, 3, 355-366.

Alvarado, S. J., Dyer, M. G., and Flowers, M. (1985a). *Understanding Editorials* (Tech. Rep. UCLA-AI-85-3). CS Dept. UCLA.

Alvarado, S. J., Dyer, M. G., and Flowers, M. (1985b). *Recognizing Argument Units* (Tech. Note UCLA-AI-N-85-1). CS Dept. UCLA.

Birnbaum, L. (1982). Argument Molecules. *Proceedings of The National Conference on Artificial Intelligence*. AAAI.

Carbonell, J. G. (1981). *Subjective Understanding*. Ann Arbor, Mich: UMI Research Press.

Cohen, R. (1983). *A Computational Model for the Analysis of Arguments* (Ph.D. Thesis). CS Dept. U. of Toronto, Canada.

Cullingford, R. E. (1978). *Script Application*. (Ph.D. Thesis). CS Dept. Yale, New Haven.

DeJong II, G. F. (1979). *Skimming Stories in Real Time*. (Ph.D. Thesis). CS Dept. Yale, New Haven.

Dyer, M. G. (1983). *In-Depth Understanding*. Cambridge, Mass: MIT Press.

Dyer, M. G. and Lehnert, W. G. (1982). Question Answering for Narrative Memory. In J. F. Le Ny and W. Kintsch (Eds.), *Language and Comprehension*. Amsterdam North-Holland,

Flowers, M. (1985). *Memory-Based Reasoning*. (Ph.D. Thesis). CS Dept. Yale, New Haven (forthcoming).

Flowers, M., McGuire, R., and Birnbaum, L. (1982). Adversary Arguments and the Logic of Personal Attacks. In W. G. Lehnert and M. G. Ringle (Eds.), *Strategies for Natural Language Understanding*. Hillsdale, NJ: LEA.

Friedman, M. (1982). Protection That Hurts (Editorial). *Newsweek*. 15 November, p. 90.

Lebowitz, M. (1980). *Generalization and Memory in an Integrated Understanding System* (Ph.D. Thesis). CS Dept. Yale, New Haven.

Lehnert, W. G. (1978). *The Process of Question Answering*. Hillsdale, NJ: LEA.

Riesbeck, C. K. (1983). *Knowledge Reorganization and Reasoning Style* (Res. Rep. #270). CS Dept. Yale, New Haven.

Schank, R. C. (1973). Identification of Conceptualizations Underlying Natural Language. In R. C. Schank and K. M. Colby (Eds.), *Computer Models of Thought and Language*. San Francisco, CA: Freeman.

Schank, R. C. (Ed.) (1975). *Conceptual Information Processing*. New York, NY: North-Holland.

Schank, R. C. and Abelson, R. P. (1977). *Scripts, Plans, Goals, and Understanding*. Hillsdale, NJ: LEA.

Thurow, L. C. (1983). The Road to Lemon Socialism (Editorial). *Newsweek*, 25 April, p. 63.

Wilensky, R. (1983). *Planning and Understanding*. Reading, Mass: Addison-Wesley.

Wilks, Y. and Bien, J. (1983). Beliefs, Points of View, and Multiple Environments. *Cognitive Science*, 7, 95-119.

ANALOGY RECOGNITION AND COMPREHENSION IN EDITORIALS

Stephanie E. August
Michael G. Dyer

Artificial Intelligence Laboratory
Computer Science Department, 3531 BH
University of California
Los Angeles, California 90024

ABSTRACT

Analogical reasoning is an important part of human intelligence. We often employ it as a vehicle for conveying ideas, and we rely upon it whenever we make a decision about a new situation [STER77]. This paper presents a theory of analogy recognition and comprehension, using as a domain letters to the editors of weekly news magazines. Our theory relies on lexical clues and the comparison of conceptual similarities to trigger recognition of the analogies in these letters. Our conceptual representation of an analogy in memory utilizes comparison links to map analogous elements to each other, and to tie together parallel arguments. We demonstrate application of this theory to a prototypical letter. The current status of a program implementing this theory is reviewed, and future research directions are discussed.

1. Introduction

Our research addresses the role of analogy in editorial comprehension, in argumentation, and in question-answer processing, using the domain of editorial letters [AUGU85] [AUGU85a]. The theory presented here is implemented in JULIP, a computer program which accepts as input a conceptual representation of a prototypical editorial letter. JULIP is part of the OpEd project [ALVA85]. The goal of OpEd is to develop a theory about the process of reasoning comprehension in the domain of editorials. The focus of JULIP is on the role of analogy in argumentation. The objective of this task is to recognize the presence of the analogy in the letter to the editor, map analogous elements together, and perform any transformations needed to complete the analogy. Understanding of the analogies presented to JULIP is demonstrated via a question-answer session with the user.

2. Analogical Reasoning in Natural Language Processing -- Some Background

Researchers in linguistics, education, psychology and other academic disciplines have studied the use of analogy and metaphor in depth [LAKO80] [ORTO79] [STER77]. Recent contributions by AI researchers in the area of computational models of analogical reasoning include [CARB83], in which Carbonell extends means-ends analysis to utilize past problem-solving experience in solving new problems. JULIP draws upon Carbonell's work in transferring experience among related problems in solving the current problem. In addition, JULIP relies upon domain specific knowledge in solving problems.

One area of analogical reasoning for which few computational theories exist is the study of analogy from the point of view of its use in editorials, arguments, conversation, debates, narratives, or other aspects of natural language text. Our work falls into this category. Two examples of related work are Winston's work on learning by analogy [WINS82] and Lebowitz' IPP [LEBO80].

In Winston's system [WINS82], a teacher uses precedents and exercises to teach the system rules about relations in a particular domain. While Winston's system is able to perform some analogical reasoning on the narratives, it does not recognize the narratives as being analogous without the assistance of the teacher. Also, the ability to make analogical mappings relies upon the existence of a

This work is supported in part by the Artificial Intelligence Center, Hughes Aircraft Company, Calabasas CA, and by a grant from the W.M. Keck Foundation.

* Also affiliated with the Software Engineering Division, Electro-Optical and Data Systems Group, Hughes Aircraft Company, El Segundo, CA 90245.

common ancestor in the AKO hierarchy of the situation parts to be mapped [WINS78] [WINS80] [WINS82]. Additional domain knowledge and past experience in making the analogical mappings are not utilized in Winston's system. As a result, the system's performance does not improve over time, and only obviously similar stories can be mapped to one another.

IPP [LEBO80] compared new wire service stories to similar events previously stored in memory. Events were indexed in memory according to their similarities and differences. IPP was successful in finding events similar to the new one and was able to form generalizations allowing it to learn about its domain. However, IPP did not form specific analogical mappings and did not deal with disputes, arguments, or beliefs.

In contrast to the work by Winston and Lebowitz, JULIP deals with the role of analogy in arguments. The objectives of JULIP include being able to have the system recognize the presence of an analogy in the input without it being identified as such by the user. The objectives also include representing the analogy itself, utilizing specific analogical mappings to show how the components of the analogy are related to one another, and being able to reason about the purpose of and basis for the analogy itself, as well as its role in an argument.

3. The Issues in Understanding Editorial Analogies

The following hypothetical letter to an editor illustrates the issues which arise in developing a system which will understanding analogies in editorial letters:

HIGH-TECH-1

> Some people are against computers because computers eliminate people's jobs. However, the automobile industry did the same thing to people in the horse carriage industry. Yet consumer demand for autos was strong enough that eventually more jobs were created in the auto industry than were lost in the horse carriage industry. In the end, the economy benefitted by the introduction of the new technology.

The author of HIGH-TECH-1 is arguing that the introduction of computer technology will eventually improve the economy by increasing the number of positions available in the job market. This point is never explicitly made in the text. Instead, it is argued by analogy to a similar situation resulting from the introduction of the automobile. We can identify three basic issues central to the theory encompassed by JULIP:

1) the ability to understand analogies:
   to recognize them, to map together the source or familiar domain and the target or unfamiliar domain, and to transform information available in one domain to fill gaps in the information available about the other domain
2) the ability to understand arguments:
   to identify propositions, and to relate propositions in support or attack relationships
3) the ability to understand the role that analogy plays in the structure of arguments

Our approach to addressing these issues incorporates the natural language comprehension component in BORIS [DYER83], the work by Flowers et al. on representation of beliefs and the structure of arguments [FLOW82], and Alvarado's work on the use of argument units to support belief recognition and inferencing about beliefs [ALVA85] [ALVA85a]. Our memory organization and causal reasoning components are based upon Schank's work [SCHA82] [SCHA77], and the implementation of JULIP'S question-answer processing draws upon Lehnert's work in this area [LEHN78].

4. Recognizing Analogies in Editorials

How is the presence of an analogy in an editorial recognized? Unless the existence of the analogy is recognized, the information associated with it is not available, and the reader will have difficulty following the author's argument.

Analogy recognition mechanisms can be activated in either of two ways. First, *analogy recognition mechanisms can be expectation driven.* In this case, the reader anticipates that the author will use an analogy, and actively seeks it out in the text. Second, *analogy recognition mechanisms can be*

data driven. In this case, the analogy is indicated by the text of the letter, or by the similarity of concepts directly presented in the letter. Both components are necessary for accurate identification of analogies. If recognition were activated only top down, the reader would miss analogies not specifically identified as such in the text. On the other hand, if recognition were only bottom up, the reader would spend a lot of time processing analogies that were never intended.

4.1 Expectation-Driven Analogy Recognition

Understanding an editorial requires that the reader identify the dispute being presented, and the technique being used to support or refute the author's arguments. Analogy is one of these techniques. Consider the following letter:

HIGH-TECH-5

> Would those who complain about computer-related job loss care to do without their cars? Buggy whip makers undoubtedly voiced similar concerns in their heyday. And I wonder how many of those complainers are employed in the auto industry.

A subject shown just the first sentence of this letter keyed in on the complainers, and drew an analogy between the introduction of the automobile industry and the introduction of computer technology. Yet nothing in the text of the first sentence directly relates the two. The subject's familiarity with the arguments in favor of the introduction of new technology enabled him to immediately focus in the comparison of the two events in the author's rhetorical question and caused him to anticipate that the author would argue the point by analogy.

4.2 Data-Driven Analogy Recognition

There are two main data driven indicators of the presence of an analogy: 1) textual clues, and 2) conceptual similarities. JULIP relies upon both of these indicators to identify the presence of an analogy.

The use of textual clues provides the most direct technique for introducing an analogy into an editorial letter. Phrases often used to directly link the source to the target are "the same as", "the same thing", "similar to", and "so it is with". This technique is used in HIGH-TECH-1, for example, when the author writes "the automobile industry *did the same thing* to people in the horse carriage industry".

People can readily detect the presence of an analogy even in the absence of textual clues. Such would be the case if the second sentence of HIGH-TECH-1 read "the automobile industry *caused* people in the horse carriage industry *to lose jobs*". In such cases, people seem to categorize concept in memory as they are encountered in the text. For a computer program to recognize an analogy presented this way, conceptual representations must be categorized as they are built, and linked together in the order in which they are encountered. As new elements are added to each category, similarity measures and other heuristics must be employed to determine whether an analogy is intended.

4.3 Constraining Comparisons: Where Does Mapping Begin and End?

Once a reader realizes that the author might be using analogy to argue a point, she must decide what elements of each domain form the basis for a mapping between the two domains, and what that mapping tells her about the author's point. Mapping of the analogy is driven by the similarities and differences between the domains of the analogy. Aspects of the domains which are similar and related to the point of the analogy support the analogy and are mapped together. Aspects of the domains which stand in contrast to each other and are unrelated to the point of the analogy do not support the analogy and are not mapped together.

5. Analogy in Argumentation

The ability to argue by analogy presumes an ability to argue in general. What framework enables people to argue? Familiarity with basic argument strategies gives people a basis around which to build arguments in support of the points they wish to convey. Alvarado et al. refer to the structure underlying arguments as argument units, and describe techniques for recognizing these units and reasoning about them [ALVA85] [ALVA85a].

5.1 Representing the Analogy in an Editorial Letter

Representing the analogy in an editorial letter requires that the arguments in the editorial be represented, and that the relationship of the arguments to each other in the context of the analogy be maintained in memory. For example, in HIGH-TECH-1, each argument contains two beliefs [AUGU85]. First, in the target domain there is the belief (ARG-1) that the computer industry is bad, because it leads to job loss. Next, there is belief that, by the same line of reasoning, the automobile industry was bad, because it, too, led to job loss (ARG-2). Both of these beliefs are supported by the fact that losing a job is considered to be bad, and, more generally, by the argument rule that

Arg-Rule-1:
X is bad
IF X leads to Y and Y is bad.

These beliefs are represented using the argument structures and rules developed in [FLOW82], [ALVA85] and [ALVA85a]. Comparison links are constructed to map ARG-1 and ARG-2, the computer and auto industries, and the causal relationships justifying the beliefs proposed about these industries.

The author's statement that the automobile industry actually led to an improved economy is viewed as a contradiction to the justification for the belief in ARG-2, and an instantiation of the argument attack strategy that

Arg-Strategy-1
IF Arg-Rule-1 holds and Y is a low-level goal,
THEN attack by showing that Y also achieves a high-level goal.

In HIGH-TECH-1, having jobs in the short term (Y) is seen as being a lower-level goal than increasing the total number of jobs available in the long run. This leads to marking ARG-2 in memory as having been attacked by this new causal relationship. Since ARG-1 is analogous to ARG-2, by the same line of reasoning ARG-1 is marked in the same way. A new belief (ARG-2') is constructed: the automobile industry is good, because it leads to an increased job market. Because of the analogical mapping between ARG-1 and ARG2, and between the industries, a fourth belief (ARG-1') is constructed: the computer industry is good, because it, too, will lead to an increased job market. The analogous components of these two new belief structures are mapped together.

6. Analogy and Question Answer Processing

Demonstrating that an analogy has been completed and understood requires answering the following categories of question:

1) Mapping: Were the key features of the source and target domains linked together?
2) Transforms: Were important features missing from one domain supplied by transforming information available from the other domain?
3) Basis: What was the purpose of the analogy? Why was the comparison made?

The heuristics needed to categorize and perform a memory search for questions in each of these categories are based upon extensions to the theory developed in Lehnert's QUALM [LEHN78]. The heuristics implemented in JULIP extend beyond Lehnert's for two reasons. First, the concepts in an editorial revolve around arguments, rather than around script instantiations. JULIP's memory search heuristics take this into account. Secondly, JULIP's representation scheme utilizes memory links that are not considered by Lehnert in QUALM.

Mapping questions demonstrate whether those similarities in the source and target domains which are related to the point of the analogy have been linked to one another. In JULIP, mapping is noted in a comparison or COMPARED-TO link, a link not considered by Lehnert. The memory search heuristics for mapping questions require retrieving the value found in the COMPARED-TO slot of the named concept.

Understanding an analogy often requires inferring information not explicitly mentioned in the text regarding one domain by transforming information available from the other domain. Transform

questions demonstrate that this inferencing has taken place. For example, in HIGH-TECH-1 the question *What did the auto industry do to people in the horse carriage industry?* should elicit the reply *PEOPLE IN THE HORSE CARRIAGE INDUSTRY LOST JOBS.* The relationship between the auto and horse carriage industries is never explicitly mentioned in the text of HIGH-TECH-1. It is inferred from the previously understood relationship between the computer industry and job loss of people on assembly lines. This is a Causal Consequent question [LEHN78], in which the the question concept is the antecedent of a causal structure. The heuristic used in this case is to retrieve the consequent of that causal structure as the answer to the question. The question category of other transform questions depends upon the type of inference to be demonstrated.

In arguing by analogy, an author states a belief in one domain, then proceeds to argue the point in a second domain. She might or might not repeat the argument in the original domain. When the author does not return to the original domain, the reader has the job of completing the analogy by creating a parallel argument in the original domain. To demonstrate successful comprehension of the author's actual point, a reader should be able to answer questions regarding the completion of the analogy. Such a question in HIGH-TECH-1 is *What will happen as computers eliminate jobs?* The reply *AN EVEN GREATER NUMBER OF NEW JOBS WILL BE CREATED.* would reflect the reader's successful comprehension of the analogy. This question is a request for a co-temporal event, a question category not considered in Lehnert's work. JULIP's memory search heuristics for co-temporal events cause it to search for another concept sharing a slot with the named concept.

Basis questions are similar to Lehnert's Goal Orientation questions, because they ask "For what purpose..." However, they differ in that they have to do with the author's goal of communicating with the reader, rather than having something to do with the events, plans, and goals associated with the components of the analogy themselves. The basic heuristic for retrieving the answer to a basis question regarding two beliefs is to generalize the justification for the beliefs. In the event that contradictory beliefs have been stated, as in the case of HIGH-TECH-1 in which the computer and auto industries are alternately considered to be good and bad, an additional heuristic comes into play: *retrieve the justification from arguments that still hold or were not attacked.*

7. Current Status

JULIP currently works on a hand-coded representation of HIGH-TECH-1, developed by following the parsing strategies described in this paper. This representation is built in memory using ARF [EDWA84]. JULIP accepts queries in English, searches a completed argument graph in memory, and returns the conceptual representation of the results of the query. Our current objective is to demonstrate understanding of editorial analogies given to the program in English through a natural language question-answer session with the user. JULIP is able to answer questions about HIGH-TECH-1 in each of the four categories listed above [AUGU85].

The most immediate plan for JULIP is to develop the parsing and generation components of JULIP that will enable it to handle verbatim input. Our approach will be to translate the theory presented here into the demons and lexical entries needed to support analogy recognition and comprehension in DYPAR, the parsing component of BORIS [DYER83]. The theory will continue to develop as additional editorial letters containing analogies and human protocols for understanding them are analyzed.

8. Problems for the Future

Many issues still need to be addressed in JULIP. Among these issues are question answer processing on the role of analogy in arguments, learning from analogies in editorial letters, and generating analogies.

The question categories currently supported in JULIP relate mainly to the completion of the analogy presented to the system. Additional question-answering heuristics need to be developed to deal with the role of an analogy in an argument. This would enable JULIP to answer, for example, the following questions relating to HIGH-TECH-1: Why was the auto industry mentioned in the argument? Why doesn't the letter's author agree with the argument that the computer industry is bad?

Work on JULIP has concentrated on representation of the analogy in the letter, leaving unanswered the following questions which deal with the issue of learning by analogies from the editorial domain: What conclusion is drawn from the analogy and learned, or retained in LTM? Why is this learned? How does JULIP's prior knowledge state affect the conclusion? How does question-answer processing affect the conclusion? How does the conclusion affect future comprehension?

Future work will explore the possibility of reversing the process of analogy comprehension in order to fulfill a long range goal of this research: to build a system which can generate arguments by analogy on its own.

9. Conclusions

We have shown how the analogy recognition mechanisms in the domain of editorial letters can be both expectation driven and data driven. Comparison links provide a means for representing the mapping of the source and targets domains of an editorial analogy in memory. Both domain knowledge and an understanding of the structure of arguments in general enable JULIP to transform an argument in in one domain of the analogy to the other domain.

JULIP provides a foundation for adding the ability to support analogy recognition and comprehension in the integrated natural language parser OpEd [ALVA85].

References

[ALVA85] Alvarado, S.J., Dyer, M.G., Flowers, M. Understanding Editorials: The Process of Reasoning Comprehension. Technical Report UCLA-AI-85-3. Artificial Intelligence Laboratory, Computer Science Department, University of California, Los Angeles. January 1985.

[ALVA85a] Alvarado, S.J., Dyer, M.G., Flowers, M. Recognizing Argument Units. Technical Note UCLA-AI-N-85-1. Artificial Intelligence Laboratory, Computer Science Department, University of California, Los Angeles, March 1985.

[AUGU85] August, S.E., Dyer, M.G. Analogy Recognition and Comprehension in Editorials. Technical Report UCLA-AI-85-7. Artificial Intelligence Laboratory, Computer Science Department, University of California, Los Angeles. March 1985.

[AUGU85a] August, S.E., Dyer, M.G. Understanding Analogies in Editorials. Proceedings *IJCAI-85,* Los Angeles, August 1985.

[CARB83] Carbonell, J.G. "Learning by Analogy: Formulating and Generalizing Plans from Past Experience", in *Machine Learning,* R.S. Michalski, J.G. Carbonell, and T.M. Mitchell, eds., Tioga, Palo Alto CA, 1983.

[DYER83] Dyer, Michael G. *In-Depth Understanding: A Computer Model of Integrated Processing for Narrative Comprehension.* MIT Press, Cambridge MA, 1983.

[EDWA84] Edwards, Gary R. *A Rule and Frame System, Version 2.2.* Hughes Artificial Intelligence Center, Calabasas CA, 1984.

[FLOW82] Flowers, M., McGuire, R., and Birnbaum, L. "Adversary Arguments and the Logic of Personal Attacks", in *Strategies for Natural Language Processing,* Wendy G. Lehnert and Martin H. Ringle, eds., Lawrence Erlbaum Associates, Hillsdale NJ, 1982.

[LAKO80] Lakoff, G. and Johnson, M. *Metaphors We Live By.* Chicago University Press, 1980.

[LEBO80] Lebowitz, Michael. "Generalization and Memory in an Integrated Understanding System", Research Report #186, Yale University Computer Science Department of Computer Science, November 1980.

[LEHN78] Lehnert, W.G. *The Process of Question Answering.* Lawrence Erlbaum Associates, Hillsdale NJ, 1978.

[MINS75] Minsky, M. "A Framework for Representing Knowledge", in *The Psychology of Computer Vision,* P. Winston, ed., McGraw-Hill, NY, 1975.

[ORTO79] Ortony, A. (Ed.) *Metaphor and Thought.* Cambridge University Press, Cambridge, 1979.

[SCHA77] Schank, R., and Abelson, R. *Scripts Plans Goals and Understanding.* Lawrence Erlbaum Associates, Hillsdale NJ, 1977.

[SCHA82] Schank, R.C. *Dynamic Memory.* Cambridge University Press, Cambridge, 1982.

[STER77] Sternberg, R.J. *Intelligence, Information Processing and Analogical Reasoning: the Componential Analysis of Human Abilities.* Lawrence Erlbaum Associates, Hillsdale NJ, 1977.

[WINS82] Winston, P.H. "Learning new principles from precedents and examples", *Artificial Intelligence,* 19:3, November 1982, p.321.

Investigations of Information Utilization during Fixations in Reading[1]

Harry E. Blanchard
University of Illinois at Urbana-Champaign

During the reading of connected text, visual information is acquired during fixations. The question addressed in this paper is when during these fixation is visual information acquired? In order to deal with this question, it is necessary to clarify what we mean by the "acquisition" of information. Acquisition can be distinguished into two processes, registration and utilization. Registration refers to visual information becoming available to the brain. Utilization refers to the visual information being used to further the comprehension of the text being read.

At least four different patterns of utilization are possible. (1) Utilization immediately follows registration. This expresses the common assumption that the first 50 ms of each fixation is devoted to information acquisition and the remainder of the fixation period to other processing activities. (2) Utilization from different regions at different times. Some segments of a word or text may be utilized at different times than other segments. One possible pattern of this kind is the left-to-right scan (e.g. Geyer, 1970). This states that letters are utilized, in a left to right sequence, beginning at the left side of the field of clear vision and proceeding to the right. (3) Utilization occurs continuously throughout the fixation. (4) Utilization occurs at a specific point in time. Utilization could occur at a specific point in time, as in the first alternative, but not necessarily immediately after registration. Two possibilities exist: utilization could occur at, or after, the end of a fixation or utilization could occur during the fixation, with the specific time of its occurrence being variable. This paper presents evidence which narrows down these possibilities to the last alternative.

Blanchard, McConkie, Zola, & Wolverton (1984) provided evidence on two hypotheses: utilization follows registration and the left-to-right scan. The experiments reported here provide evidence on the left-to-right scan and continuous utilization hypotheses. These experiments all use a variation of Blanchard et al.'s (1984) technique.

Blanchard et al. (1984): Experimental Procedure and Findings

How can the time course of utilization be experimentally determined? Blanchard et al. (1984) manipulated the visual input available during fixations. If the visual pattern is different at different times during the fixation, then what the reader reports having seen can specify the time at which visual information was utilized. Fast control over the stimulus during the process of reading continuous text was achieved through eye movement contingent display control. In this technique, the subject reads text from a cathode ray tube (CRT) linked to a computer and an eyetracker. The CRT provides the ability to change the text presented in any way desired very quickly (approximately 4 ms). Eye movements are monitored while the subject is reading. The signal from the eyetracker is collected by a computer, which

1. This research was supported by the National Institute of Education under Contract 400-76-0116 to the Center for the Study of Reading.

identifies, on line, whether the subject is fixating or making a saccade. The text displayed on the CRT can be changed contingent upon what the subject's eyes are doing. In the manipulation used by Blanchard et al. (1984), the text was changed by switching a single letter on the line.

Passages were written in which two words fit into the same place in the text, e.g. The underground caverns were meant to house hidden (tombs,bombs), but then the construction was stopped because of lack of funds. Changing a letter, from t to b, changes the meaning of the text. This critical letter was changed partway through each fixation that the subject made in the vicinity of the letter. The word containing the critical letter is the critical word. At the beginning of a fixation, one word was in the text (tombs). Partway through the fixation, the line of text was replaced by a mask of X's for 30 ms, and the text was then returned to the screen for the remainder of the fixation. However, after the mask, the critical letter was different than it had been in the early part of the fixation, thus changing the critical word to bombs. During the saccade, bombs was changed back to the original word, tombs, and the cycle began again. In this way, the same sequence of display changes was repeated for each fixation-saccade cycle. This sequence of changes caused different information to be present in the early and late parts of the fixation.

The purpose of the mask in this experiment was to eliminate apparent movement which was localized at the critical letter position when the letter changed. Localized apparent movement attracts attention to the location of the movement. The mask causes apparent movement at all letter positions equally, thus removing the effect caused by a localized change.

After reading each passage, the subject was given four test words, one after another, and indicated for each whether or not it had been in the text. Two of the test words were the critical, changing words, e.g. tombs and bombs in the above example. Which word or words were reported indicated whether the critical letter was utilized before the mask, after it, or throughout the fixation.

Given these experimental conditions, the hypothesis that utilization immediately follows registration predicts that only the first word present during the fixation should be reported. Blanchard et al.'s (1984) results do not conform to this prediction. Sometimes subjects reported reading the first word, sometimes both words, and sometimes only the second presented word. Each of these reports occurred with nearly equal frequency (29% for the first, 36% for the second, and 35% for both). Clearly, utilization does not inevitably occur at the beginning of the fixation -- sometimes it occurs later in the fixation.

The Left-to-Right Scan Hypothesis

Blanchard et al. (1984) conducted several tests on the left-to-right scan hypothesis, and found no evidence for it. Two additional experiments were specifically designed to test this hypothesis.

In Experiment 1, the same technique as Blanchard et al. was used, except now there were two critical letters in a critical word instead of just one. Texts were used in which four words contrasting by two letters all fit into the same position in the text, for example, Ruth's great aunt is definitely the most (mushy,musty,gushy,gusty) person she has ever met. During fixations

in reading, the two critical letters were changed (with an interpolated mask) so that one of the four possible words was present during the early part of each fixation (mushy) and a second word during the latter part of each fixation (gusty). The other two words were never presented. If a left-to-right scan were taking place, it would be expected that at least sometimes the first critical letter from the first word and the second critical letter from the second word would be utilized, causing the subject to report reading a word that was never present on the screen (musty in this example). The frequency of this phenomenon would depend on the speed of the scan and on where the scan begins (and, therefore, on where the fixation was with respect to the critical letter).

The results were much like Blanchard et al. (1984) in that the subjects sometimes reported the first present word, sometimes the second, and sometimes both. However, they did not report the musty-type non-presented words which would be expected from a left-to-right scan hypothesis with a frequency greater than that expected purely by error. Also, there was no pattern of the frequency of such reports varying with where the fixations were with respect to the position of the critical letters. Thus, no support was found for a left-to-right scan.

In Experiment 2, the same basic procedure as in Experiment 1 was used, except that now every letter in the critical word was changed after the mask was removed during fixations. Passages were written in which a pair of critical words differing at every letter position fit into the same position in the text, for example: Sandy spent a long time preparing the (melon,cakes) for dessert and completely forgot about the hors d'oeuvres. In this case, if a left-to-right scan takes place, there should be some instances where some letters from first presented word and some letters from the second presented word are utilized together. The subject would perceive a non-word as a result.

The results of changing an entire word during a fixation resemble the original results of Blanchard et al.'s (1984) single letter switch, in that sometimes a single word was reported and sometimes both critical words were reported. However, the relative frequency of reporting both critical words presented during the fixation doubled with respect to the single letter change experiment. This may be due either to the greater possibility of localized movement being perceived despite the mask or to the greater disruption to the perceptual processes prior to utilization. With respect to the scan hypothesis, however, subjects did not report seeing non-words.

It is possible that instead of seeing a nonword, the perceptual system may allow some kind of error recovery, where perhaps both critical words are consciously perceived. In this case, instances where only one critical word is reported would be expected in cases where the scan traversed all the letters of the critical word competely before or after the mask replaced the text. These instances would have to be cases where the scan started very close to or far away from the critical word. This translates to the prediction that the probability of reporting one or both critical words should vary as a function of fixation location. This was not the case, however.

The results of these experiments rule out the left-to-right scan, with the possible exception of very rapid scans, faster than 10 ms per letter.

The Memory Process Explanation

The next possible pattern is continuous utilization throughout the fixation. This would seem to be ruled out by Blanchard et al.'s (1984) basic results, in which subjects reported reading only one critical word, indicating that utilization did not occur for the entire duration of the fixation. However, there is a way in which continuous utilization could be made compatible with these results. The reports of reading a single word could be explained through the action of memory processes rather than through perceptual processes. Under this alternative hypothesis, both critical words present during a fixation are always perceived (utilized), but one of the critical words is more susceptible to forgetting than the other. Cases where the subject reports reading only a single word are caused by forgetting processes operating after utilization.

Two experiments were conducted to test the memory process hypothesis. In Experiment 3, the display changes and texts were similar to those used by Blanchard et al. (1984): a single letter was changed during fixations. However, subjects in Experiment 3 were instructed to press a button while they were reading if they observed a word change. These were all non-naive subjects: they were aware of what a letter change looked like. The memory process explanation requires that there should be no correlation between the indications of seeing a letter change and whether subjects report one or both critical words. The results are not consistent with this explanation. When subjects did indicate seeing a letter change, they reported both words on the recognition test 92% of the time, and when they did not indicate a change, they reported a single word 80% of the time. Performance on the recognition test appears to accurately reflect what subjects are detecting during their on-going reading. Therefore, the results do not support the memory process hypothesis.

In Experiment 4, the single letter change technique was again used to test the memory process explanation. On half of the passages in this experiment, the text was removed from the screen completely during a saccade taking the eyes away from the critical word. When this happened, subjects were instructed to verbally report the last few words that they remembered reading, and to report any letter changes they had detected. This "turn-off-the-text" technique is an alternative method to the recognition test. It greatly reduces non-immediate effects such as memory processes by obtaining reports very soon after the critical word was fixated. On half the texts, this turn-off-the-text procedure was used, but on the remainder, subjects proceeded through the text without interruption and then performed the recognition test. This allows comparison of the two testing methods and assessment of the influence of non-immediate effects on subjects' reports.

There was essentially no difference between the recognition test and the immediate verbal reports, in that the percentage of reports of one critical word and of both critical words are nearly the same. This is in direct contradiction to the memory process explanation, which predicts that subjects should always be able to report both critical words present, because they do not have an opportunity to forget one of the words. At the very least, the probability of reporting both critical words should increase with the immediate verbal reports, but this is not the case. It seems safe to conclude that the pattern of reporting which was observed in the recognition tests of this and previous studies largely reflects processes that are occurring during

perception, or, at least, during the very early stages of processing. These results do not support the memory process explanation, and so are not consistent with the hypothesis that utilization occurs continuously throughout the fixation. The results are consistent with the position that utilization occurs during a specific, delimited point in time during the fixation.

Conclusion

These experiments have clearly narrowed down the possible patterns of utilization described above. Utilization does not typically occur immediately after registration, letters are not utilized in a left-to-right scan, and utilization does not occur continuously throughout the fixation.

This leads to the last possibility, that utilization occurs at a delimited point in time during the fixation. There are two possible versions of this hypothesis. In one, the pattern of sometimes reporting a single word is attributed entirely to the pattern masking characteristics of the interpolated mask. The mask is viewed as a backward mask for the first presented word and a forward mask for the second presented word. One or both critical words will be reported, depending on the effectiveness of the mask for each critical word during an individual fixation. This involves a kind of indirect competition between the two words present during a fixation. Visual information is available from both words when utilization takes place, although either one or both words may be perceived. For this to occur, utilization must take place at the end of the fixation or after it is terminated, in order for information from both words to be available.

The other possibility is that reports of reading a single critical word directly indicate when utilization occurs during a fixation. That is, the fact that the subject reports the word present early in the fixation or the word present later in the fixation corresponds to the actual point during the fixation at which utilization occurred. If this is so, then the time of utilization would appear to vary, occurring sometimes early and sometimes later in the fixation. This was proposed by Blanchard et al. (1984). It can be further speculated that the variable time of utilization is influenced by the current need of concurrent language processes. When on-going language comprehension is ready to accept new input, then utilization will occur. This activity will not be in direct correspondence with the saccade-fixation cycle, so the time of utilization will vary independently of the making of a new fixation. Further research examining these two possibilities is being conducted using the letter change during fixations technique.

References

Blanchard, H. E., McConkie, G. W., Zola, D., & Wolverton, G. S. (1984). Time course of visual information utilization during fixations in reading. Journal of Experimental Psychology: Human Perception and Performance, 10, 75-89.

Geyer, J. J. (1970). Models of perceptual processes in reading. In H. Singer & R. B. Ruddell (Eds.), Theoretical models and processes of reading (pp. 47-94). Newark, DE: International Reading Association.

Two Endorsement-based Approaches to Reasoning About Uncertainty

Paul Cohen
Computer and Information Science Department
University of Massachusetts
Amherst, Massachusetts 01003

§1 ABSTRACT

Two approaches to reasoning about uncertainty are discussed. A parallel certainty inference model superimposes reasoning about the credibility of inferences on a deductive framework. A second approach identifies and implements representativeness as the general determinant of credibility in classification tasks. These approaches are steps in the evolution of endorsement-based reasoning, a view of uncertainty in terms of structured objects that represent characteristics of evidence.

§2 INTRODUCTION

This paper is about evidence. It is also about uncertainty, since uncertainty is a state of mind that arises when evidence is incomplete, inaccurate, irrelevant, and so on. Since these and other characteristics of evidence give rise to uncertainty, reasoning about uncertainty ought to be reasoning about characteristics of evidence. But this is difficult for Artificial Intelligence (AI) programs, which are not provided with explicit representations of characteristics of evidence. Most frequently, all knowledge about uncertainty is summarized in a single degree or range of belief.

Endorsements are explicit representations of characteristics of evidence. This paper describes two stages in the theory of reasoning with endorsements. It draws together results from several sources,[1] and is necessarily condensed. The endorsement-based view is prescriptive, in the sense of saying how reasoning about uncertainty *ought* to be done by intelligent computers. Some prescriptions have been implemented, others remain open research problems. One version of endorsement-based reasoning, described below, also purports to describe human reasoning under uncertainty.

Three prescriptions for intelligent reasoning about uncertainty guide the development of endorsement-based reasoning:

> **Orientation towards action.** It is striking that humans, though uncertain about many or most decisions *act as if certain* about these decisions. AI programs are

[1] Cohen and Grinberg (1983); Cohen (1985); Sullivan and Cohen (1985); Cohen, Davis, Day, Greenberg, Kjeldsen, Lander and Loiselle (1985).

less able because they lack representations of *reasons* for uncertainty. Decision is trivialized by reliance on *degrees* of belief, exclusive of reasons, since decision strategies have access only to the weight of evidence, not its other characteristics. For example, given only the degree of belief in a hypothesis, one cannot decide to minimize uncertainty by attempting to prove its converse. Nor can one decide to "wait and see," in the expectation that more evidence is forthcoming. Both decision strategies require knowledge about *why* a hypothesis is uncertain: Is its converse more credible? is it currently unsupported but expected to gain support soon? An orientation to action under uncertainty mandates this kind of explicit knowledge about uncertainty.

Justification. Humans can justify their decisions. AI programs can generally explain how they derive factual conclusions, but not why they believe or disbelieve the conclusions. Once again, the lack of explicit representations of characteristics of evidence is at fault.

Advocacy and consensus. Humans can view the same evidence from different perspectives – arguing for and against a position to convince oneself or another. Avron Barr (1985) points out that, in addition to advocacy, humans may argue to find a consensus. The goal here is not to bludgeon an opposing position with the club of evidence, but to seek a new interpretation of the evidence that accommodates two apparently opposing views. Both advocacy and consensus focus on how evidence supports arguments. The weight of evidence is certainly important, but no more so than the source, cost, reliability, and other characteristics of evidence.

Guided by these considerations, we developed two forms of endorsement-based reasoning about evidence.

§3 PARALLEL CERTAINTY INFERENCE

The parallel certainty inference model divides reasoning under uncertainty into two parallel streams. In one, deductive inference is mediated by modus ponens; and in the other, representations of the credibility of deductive inferences are propagated. This split is necessary because deduction is not defined for propositions that are neither true nor false. Therefore, to reason deductively under uncertainty, one must add representations of uncertainty to logical (true or false) propositions. Degrees of belief in expert systems are the best-known example of parallel certainty inferences. For example, MYCIN uses modus ponens to backchain through a rulebase of implications, and uses quasi-Bayesian combining functions to propagate degrees of belief from one conclusion to the next (Buchanan and Shortliffe, 1984, Chapter 11). Since we believe that numbers are inadequate representations of characteristics of uncertainty, we developed a parallel certainty inference model in which numbers were replaced by endorsements.

Cohen (1984) and Sullivan and Cohen (1985) studied the role of endorsements in plan recognition. The problem is this: Given a set of plans, each composed of elementary actions, determine the plan a user has in mind by watching his actions. For example, imagine the given "library" contains only three plans, each composed of three elementary actions:

Plan	Steps
plan1	a b c
plan2	b d e
plan3	d f g

Further assume, for simplicity, that plan steps are not "shared": step **b**, for example, does not simultaneously continue plan 1 and start plan 2.

How can we interpret the user actions **a** followed by **b**? The first action suggests the user intends plan 1, since it is unique to plan 1. However, it might have been a mistake: the user might actually intend plan 2. The second action, **b** appears to support the plan 1 interpretation, since it is the next step in the plan. However, **b** is ambiguous: the first action might have been a mistake and the user might be starting plan 2. Both interpretations are uncertain. Sullivan and Cohen represent the uncertainty with mnemonic endorsements:

Step	Interpretation	Endorsements
1: **a**	(start plan1 **a**)	(**a** unique to plan 1 +) (**a** could be a mistake -)
2: **b**	(continue plan1 **b**)	(**a b** continuity is desirable +) (**b** ambiguous -) (**b** could be a mistake -)
b	(start plan2 **b**)	(**a b** discontinuity is undesirable -) (**b** ambiguous -) (**b** could be a mistake -)

Asked to argue that the actions **a, b** are evidence of plan 1, one can read the positive endorsements (that's what the + means) that **a** is unique to plan 1, and is continued by **b**. Asked to argue against this interpretation, one reads the negative endorsements: **a** might be a mistake, **b** is not unique to plan 1. The plan 2 interpretation of the actions accrues no positive endorsements, but the negative ones can be used to argue against it: the user prefers not to start one plan while in the middle of another (i.e., discontinuity is undesirable); **b** is ambiguous, and might have been a mistake.

These endorsements seem to explicitly capture the sources of uncertainty in the plan

interpretation task. Yet they are only tokens: their meanings must be supplied from somewhere. "Discontinuity is undesirable" stands for the intuition that a user prefers to finish one task before starting another. Imagine replacing the mnemonic endorsements with arbitrary symbols, and you will see how important it is to specify the semantics of endorsements, and not rely on their mnemonic power. How much knowledge would a program need to reason about the symbol x43 as well as humans reason about the possibility that an action is a mistake? This knowledge imparts meaning to the token "could be a mistake." Sullivan and Cohen found that the meaning of endorsements was adequately specified by their applicability conditions -- when to add them to a proposition -- and by rules specifying when to add and delete endorsements from propositions in the light of new evidence. The latter kind of knowledge played an important role in combining evidence.

The task of combining endorsements in light of evidence is best introduced with an example. Given the plans above, we can interpret action **a** as evidence for plan 1, subject to the qualification that it could be a mistake. Now, if the next action is **b**, is the qualification still valid? Might our uncertainty be tempered by the fact that **b** is consecutive with **a** under the plan 1 interpretation of both actions? If the third action is **c**, the concluding step in plan 1, can there be any doubt that **a** was not a mistake? If it is reasonable to diminish uncertainty due to a possible mistake given subsequent consistent evidence, then we might erase the negative endorsement "could be a mistake - " given such evidence. This is exactly the method used by Sullivan and Cohen. They specified *semantic combining rules*, so called because they defined the meaning of endorsements in terms of the way endorsements combine:

If (plan N: step i could be a mistake -) and
    (plan N: steps i j continuity is desirable +)
Then erase (plan N: step i could be a mistake -)

(Though the implementation of this and other semantic combining rules literally erased endorsements, a better implementation would withdraw support for them. This reasoning, in the style of Doyle's reason maintenance (Doyle, 1983), keeps an important record of the fate of hypotheses as evidence is accrued.)

Semantic combining rules provide one mechanism for implementing advocate and consensus reasoning. Consider the justification offered by one advocate for the plan 1 interpretation of **a**, **b**: "The actions are consecutive -- evidence that plan 1 is intended. True, **a** might be a mistake, and **b** is ambiguous, but, for me, the likelihood that **a** is a mistake given the consecutive action **b** is small." Another advocate uses the same evidence and endorsements but takes another view: "True, the actions are consecutive, but **a** might have been a mistake, and since **b** is ambiguous, I cannot argue that both belong to plan 1." We can model these different views in terms of semantic combining rules. The first advocate is governed by the rule shown above; the second by a more conservative rule:

If (plan N: step i could be a mistake -) and
(plan N: steps i j continuity is desirable +) and
(plan N: step j has only one interpretation +)
Then erase (plan N: step i could be a mistake -)

Consensus is more difficult to model, but involves comparing the semantic combining rules: The advocates disagree because the ambiguity of **b** is significant to one and not to the other. Several intermediate, consensus positions can be worked out; for example, the third clause in the previous rule might be rewritten "only one interpretation of step j currently has support," which would lead to agreement on the plan 1 interpretation of **a, b**, since all the endorsements of the plan 2 interpretation of this input are negative, as discussed above.

§4 REPRESENTATIVENESS

Our second approach to endorsement-based reasoning addressed two related issues: However informative our endorsements appeared, they were still comments added post hoc to deductive inferences. This was an inevitable consequence of the parallel certainty inferences, but it leaves open the questions "where do endorsements come from, and what do they mean?" At the same time, we became interested in associative, as opposed to deductive, reasoning. Expert and commonsense reasoning derives its power from empirical associations – propositions that "go together" in the mind. We thought that the manner in which the propositions go together – the associations between them – could tell us something about the credibility of inferences based on those associations. Unfortunately, the nature of an association (be it causal, temporal, etc.) is lost when it is represented as an implication; and most empirical associations are represented this way, as production rules. The failure of this approach, it seemed, was that implications are interpreted deductively – as logical forms – and thus the parallel certainty inference model is inescapable. Yet, if the associations that underlie inferences are not discarded in the translation from empirical association to implication, then they might be used to assess the credibility of the inferences. By focusing on associative reasoning, we eliminate the parallel certainty inference model, and the questions it raises about the meaning of endorsements. At the same time, we gain a set of associations that are a natural basis for statements about the credibility of interpretations – endorsements.

We addressed a single associative task and its inherent uncertainty. The task was classification, prevalent in AI systems. Partial matching occurs whenever the criteria for a classification are not met exactly. For example, given the empirical association *a person with queasiness, fatigue, aching limbs, and a fever has flu*, what can we say of an individual with a poor appetite, headache, marginal fever, and a twitch in the left eye? Flu might be considered if poor appetite is evidence of queasiness, headache is evidence of aching limbs, and so on, although no reasonable interpretation of the twitch seems possible. The partial matching problem has two forms: Not all the

criteria for a classification are met, and those that are may be met approximately. The individual above did not complain of fatigue – one of the criteria for flu – and did not complain of other criteria exactly as stated. Yet all complaints but the twitch could be interpreted as evidence for flu.

One form of the partial matching problem – inexact matches between evidence and classification criteria – reminded us of Tversky and Kahneman's notion of representativeness:

> Many of the probabilistic questions with which people are concerned belong to one of the following types: What is the probability that object A belongs to class B? What is the probability that event A originates from B? What is the probability that process B will generate event A? In answering these questions, people typically rely on the representativeness heuristic, in which probabilities are evaluated by the degree to which A is representative of B, that is, by the degree to which A resembles B. (Tversky and Kahneman, 1982, p.4)

One's certainty in a diagnosis of flu depends on the degree to which the symptoms are representative of flu. Returning to the previous example, is poor appetite (a symptom) representative of queasiness (a criterion for flu)? Is a headache representative of aching limbs? We accepted the idea that representativeness mediates credibility in classification tasks, and set about implementing judgments of representativeness. What follows is a necessarily abbreviated discussion of the work presented in Cohen et al. (1985).

Our approach is to represent propositions as structured objects (frames) and to measure representativeness in terms of the associations that hold between structures. For example, consider this structure for the concept *tobacco*:

Tobacco

PART-OF cigarettes
CAUSE cancer

Is the inference *cigarettes cause cancer* credible? If so, it must be justified by the PART-OF association between tobacco and cigarettes. Consider a similar case:

Industrial emission

HAS-PART: carbon dioxide
CAUSE : acid rain

Is it credible to infer that carbon dioxide causes acid rain? If not, it must be because carbon dioxide is only one part of industrial emissions, and thus cannot credibly be implicated as a cause, given other possible causes. The credibility of the first example

is due to the PART-OF relation, but the HAS-PART relation does not support causal arguments credibly. We say that the one relation *preserves representativeness* and the other does not. More formally:

CAUSE(A,B) PART-OF(A,C)	but,	CAUSE(A,B) HAS-PART(A,C)
CAUSE(C,B)		** CAUSE(C,B)

The ** indicates an unrepresentative, or less credible, conclusion. This style of reasoning has much in common with Collins' plausible reasoning (1978). The central idea is that the credibility of a conclusion given evidence is determined by the associations that relate evidence and conclusion in a network of concepts. Representativeness, or credibility, is a function of associations. In the previous example, cigarettes may be considered to cause cancer, given the evidence that tobacco causes cancer, because the agent in the evidence (tobacco) is a PART-OF the agent in the conclusion (cigarettes).

We have developed a program to reason associatively in a classification task, and to qualify its conclusions as representative or unrepresentative based on the associations that underlie classifications. The domain of the program, called GRANT, is research funding. A proposal (P) states a researcher's interests. GRANT finds a set of funding agencies (A) that are representative of the proposal. The agencies judged representative of a proposal by GRANT were judged in turn by an expert to be those most likely to fund the proposal. I will briefly describe the structure, function, and testing of GRANT.

GRANT has two kinds of knowledge about the world. One is a network of about 1000 research concepts, and the other is a set of rules, like those above, for finding representative connections between concepts. Research proposals and the interests of funding agencies are represented as structures attached to the network of research concepts. For example, an agency that wants to study the effects of *diet* on *cardiovascular disease* is linked to the network by these topics. A researcher may propose to study the effects of dietary sodium on atherosclerosis. Is this representative of the agency? That is, is the agency likely to fund the proposal? The answer depends on the associative relations between the topics subsumed by the agency and the proposal. As it happens, dietary sodium is PART-OF diet, and atherosclerosis ISA cardiovascular disease. Both relations have been shown to preserve representativeness, so the proposal is representative of the agency's interests.

Most of the knowledge engineering effort on the GRANT project has been devoted to discovering, from a human expert in the domain of funding, the associations that link research concepts, and the rules that state whether the associations and their combinations preserve representativeness. Combinations of associations are called *path endorsements*. A set of *traversal rules* restricts GRANT to consider the associative

pathways that are adequately endorsed. Given the rules, a program can find sources of funding for which a proposal is representative, simply by starting in the associative network of research concepts at the proposal and spreading activation through the network over representative associations until it "bumps into" topics associated with one or more funding agencies.

The use of traversal rules constrains blind spreading activation, and finds the agencies most likely to fund a research proposal. We tested this claim with 23 research proposals. For each, we ran GRANT in two modes. One, called *minimum-distance search* (MD for short) spread activation without regard for representativeness until it found, on average, 15 agencies. The other mode, called *endorsement constrained* (EC for short) searched until it found all representative agencies. For each proposal, the agencies found by MD search were given to an expert, who was asked to select the agencies most likely to fund the proposal. For the 23 proposals, the expert selected an average of 2 agencies from the average of 15 found by MD search. EC search found, on average, 80% of the agencies judged good by the expert. However, about 32% of the agencies found by EC search were not judged good by the expert. Thus EC search has a respectable hit rate (80%) and an adequate false positive rate (32%). Its false positive rate, compared with that of MD search under a variety of stopping conditions, is much superior. This suggests that the rules that govern EC search do, in fact, discriminate between representative and unrepresentative alternatives. Moreover, they effectively capture the determinants of the likelihood that an agency will fund a proposal.

Path endorsements have not yet been shown to mediate explantation and advocacy and consensus reasoning. Endorsements and combining rules of the kind discussed in Sullivan and Cohen are better suited to these purposes. However, the semantics of path endorsements is much clearer than those added to a parallel certainty inference model. Current research is devoted to attaining advocacy and consensus reasoning in GRANT.

§5 REFERENCES

Barr, Avron. 1985. Systems that don't understand. In the proceedings of Cognitiva: Artificial Intelligence and Neuroscience. Centre d'Etudes des Systemes et des Technologies Avancees, Paris.

Buchanan, B.G. and Shortliffe, E.H. 1984. Rule-Based Expert Systems: The MYCIN Experiments of the Stanford Heuristic Programming Project. Addison Wesley, 1984.

Collins, A. M. Fragments of a theory of human plausible reasoning. TINLAP-2. p.194-201.

Cohen, P.R., 1984. Progress report on the theory of endorsements: a heuristic

approach to reasoning about uncertainty. IEEE Workshop on Principles of Knowledge-Based Systems. p. 139.

Cohen, P.R., 1985. Heuristic reasoning about uncertainty: An artificial intelligence approach. Pitman Research Notes. London. (Also published as Stanford University Technical Report 83-986, 1983)

Cohen, P.R., Davis, A., Day, D., Greenberg, M., Kjeldsen, R., Lander, S., and Loiselle, C. 1985. Representativeness and uncertainty in classification systems. AI Magazine. Fall 1985.

Cohen, P.R., and Grinberg, M.R. 1983. A theory of heuristic reasoning about uncertainty. AI Magazine, Summer, 1983.

Doyle, J. Some theories of reasoned assumptions: Essays in rational psychology. CMU-CS-83-125. Carnegie-Mellon University.

Sullivan M., and Cohen, P.R. 1985. An endorsement-based plan recognition program. IJCAI 1985, forthcoming.

Tversky, A., and Kahneman, D. 1982. Judgment under uncertainty: Heuristics and Biases. In Judgment Under Uncertainty: Heuristics and Biases, Kahneman, D. Slovic, P. and Tversky, A. (Eds.) Cambridge University Press.

Teleology + Bugs = Explanations

Gregg C. Collins

Yale University

In previous work, Schank and Collins (1982) have argued that learning requires the ability to explain the failures of expectations derived from the knowledge structures used in planning and understanding. Building a computational theory based on this idea, however, depends first of all on our ability to understand what kind of explanation of an expectation failure is required for learning, and what a system would need to know in order to construct such an explanation. One approach to this rather formidable task is to analyze actual instances of such explanation performed by people. This paper will look at one such instance in detail, and attempt to sketch the general method which might underlie it.

The example we will analyze involves explaining expectation failures arising from the behavior of a broken vending machine. This problem can be viewed as a kind of "debugging," and in fact prior work on that topic and on the analysis of device models in general (see, e.g., Sussman & Brown, 1974, McDermott, 1976, Stevens & Collins, 1978, Forbus & Stevens, 1981, de Kleer & Brown, 1981) will be relevant to the analysis. However, a big difference in this case is that we begin without a model of how a vending machine operates. Instead, we begin with our everyday knowledge of how to use one when it is functioning properly, plus an understanding of its purpose as seen by its owners and designers. In fact I will argue that we *conetruct* a model of how the machine works as a necessary part of explaining what is wrong with it. Our ability to reason teleologically about things (i.e. with respect to their "purposes") (deKleer & Brown, 1981, Sussman & Brown, 1974), combined with the information which can be derived from an analysis of the ways in which the device fails to fulfill its purposes, can take us surprisingly far in constructing a model of a machine about which we originally had little specific knowledge. The model must be developed to the point where it can be used to construct an explanation of the cause of the bugs we observed.

The actual episode in which the failures occurred was this: I went up to an ordinary Coke machine, which I had previously used without any problem, and deposited a quarter. The quarter came back in the coin return box. I tried several more times using different quarters, with the same result each time, and with no apparent effect on the machine. I then tried other denominations of coins, which the machine accepted. I tried a quarter again after having deposited some other coins, and it was again rejected. When I had deposited 30 cents in nickels and dimes, the machine lit up, signalling that I could now have a Coke, which it gave me. The usual price for a Coke in that machine was 55 cents. The same behavior occurred subsequently when I tried the same machine again.

The machine was behaving in a way that consistently violated two of the expectations derived from my get-a-Coke *script* (Schank & Abelson, 1977). I expected to be able to use quarters in the machine, and I expected to get the Coke for the advertised price of 55 cents. What was wrong? The explanation that immediately presented itself to me was that the machine consisted of some sort of ramp between the input slot and the coin return box down which coins slid or rolled. Off of this ramp would be chutes for quarters, dimes, and nickels, separately, and at the top of each chute would be some sort of sensing device which would increment a counter whenever a coin went by. What must be happening, I thought, was that a quarter was stuck at the top of the quarter chute, blocking other quarters and forcing them to slide on down to the coin return. This quarter would be at least partially inside the sensing device, and would be "counted" once on each cycle of the machine, so that Cokes effectively would cost 25 cents less than they normally did. I believe this to an interesting example the explanation was not completely obvious, implying that the reasoning required was not trivial, yet there seems to be a fairly unanimous consensus that the explanation is the right one among people to whom I subsequently presented the problem, implying that special knowledge or abilities were not required to produce the explanation, but that it was in fact only a slightly clever instance of whatever it is people normally do in constructing explanations.

Analysis of the example

I will now present an analysis of the processes and prior knowledge which are necessary to produce the above explanation in terms of what a program capable of constructing it would need to know and do. The description will be presented as a series of steps starting with the initial knowledge of the expected behavior of the machine, and leading to a model of the machine which is complete enough to, first, explain how the vending machine *should* accomplish the functions which were perturbed, and second, how the perturbations could arise from plausible malfunctions of those mechanisms.

Step one: Modelling what should have happened

Initially what I knew was a rote plan, or *script*, for using a Coke machine, which told me what to do and what observable behavior to expect from the machine in response. One expectation told me that if I inserted dimes, nickels, or quarters in the machine, they would remain within, as long as the cost of a Coke had not been exceeded. Another told me that it would cost me the advertised price to get the Coke. In both cases, the actual observation differed from the expectation.

What we need in the final explanation of the discrepancy is a model of *how* the things we expected to happen are accomplished. What we have, probably, are just the expectations themselves. What I want to show is that we can bridge the gap by asking ourselves the teleological question: what purposes of the machine were affected by its failure to operate as expected?

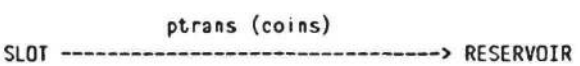
We know of two purposes which are relevant to the observed bugs:

1. The vending machine is meant to collect a customer's money.
2. It is meant to give the customer a Coke when a certain amount of money has been collected.

Transforming this teleology into the mechanism required for the explanation is a matter of distributing the teleology among a set of interacting components which are themselves characterized, at first, *only* by their teleology. The interactions among the components can be represented by links which denote various types of causal relations. Two causal link types are needed in this example, corresponding roughly to Conceptual Dependency primitives PTRANS and MTRANS (Schank & Abelson, 1977). These are intended to indicate, respectively, the transfer of some physical object, and the transfer of information, between two components.

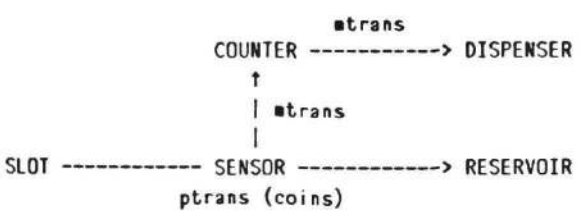
In order to distribute the teleology of the mechanism among interacting components, we must first determine suitable components. It may not require inventing new component types; rather, it may suffice to select items from an existing stock of devices described in terms of their teleology. However, our knowledge of the devices includes more than just their teleology, in particular, knowledge of how such devices are commonly implemented. We are therefore not only following deKleer in asserting that teleology is a crucial aspect of our vocabulary for device description, but further claiming that this serves as an organizing principle for memory about devices.

Given an appropriate stock of simple components, our knowledge of the existence of the coin slot from the rote plan enables us to construct the following model accounting for the first teleological fact about the vending machine, namely that it takes the customer's money:



A "reservoir" is simply a component which collects something. The appropriate causal links between the components is part of our knowledge of the components themselves. Reservoirs, we know, must have a link enabling an inflow of whatever substance they store. The nature of this link depends on what that substance is. In the case of a physical object like a coin, the mechanism by which it is input must of course be physical transfer, i.e., PTRANS.

We can extend this model to account for the second teleological fact, namely, that the vending machine should dispense a Coke when a certain amount of money has been collected:



A "dispenser" is a component which "gives one something." Inferring the existance of the "counter" and the "sensor" is a bit more complex: we must know that when control of an operation, in this case dispensing, is dependent on detecting a certain amount of something, then a sensor is required for detection, and a counter is required for storing the quantity detected so far. In fact we could abstractly view a counter as a reservoir for a quantity, and analogously with the coin reservoir, determine that since quantity is information, and causally appropriate input link is transfer of information, i.e., MTRANS. The sensor must be inserted in the path between the slot and the reservoir because the coins which are collected to satisfy the first purpose of the vending machine are the same coins which must be counted to satisfy the second purpose.

Step two: Locating the bug

Once we have constructed this simple model of the Coke machine, we must attempt to locate the sources of the observed bugs in the model. At this point we need to invoke domain-independent rules about locating bugs. The first, and most important is the well-known "single-fault assumption," which is the assumption that multiple bugs have a single source.

We can make use of this assumption to locate the bug due to a nice property of our representation, namely that each link and the devices at its endpoints can be seen as a "nearly decomposable" (Simon, 1969) subsystem. Thus, if one can assign bugs to different subsystems in the representation, the actual fault must lie at the intersection of those subsystems. In this case, one bug is clearly in the SLOT-RESERVOIR system, since coins end up in the coin-return instead of the reservoir. The other bug is in the COUNTER-SENSOR system, since the machine seems to be counting wrong. Applying single fault assumption tells us that the source of both bugs should be at the point where the sensor phyically contacts the SLOT-RESERVOIR connection. Thus a major part of the ultimate explanation stems from our ability to apply the single-fault assumption to our model in this way.

Step three: Modelling the bug

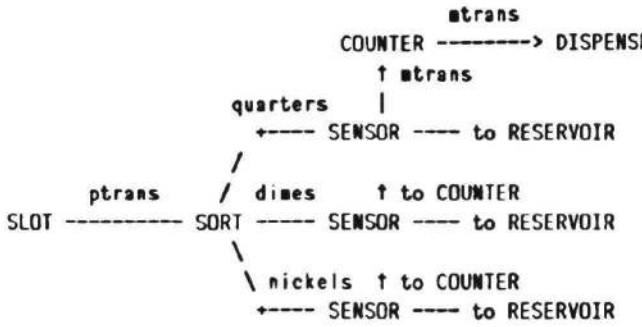
At this point, although we have located the source of the bugs, we have not yet explained them. We can look to two sources for the underlying causes which might explain a bug: the properties of the correctly functioning device, and the properties of the fault which is actually producing the bug. The observed symptoms which constitute the bug can generally be viewed as resulting from some relatively straightforward interaction between these two sources. One way of sorting out the interaction is to isolate characteristics of a bug which can be attributed predominantly to either the device alone or to the fault alone.

For example, we might view the entire description of the first bug, "rejects quarters," as such a characteristic. Viewing this bug in isolation from the second bug, we might then attribute it either to the device alone (the machine has a reason for rejecting quarters, e.g., when it is out of change) or to the fault alone (for example,the coin slot is blocked so that nothing as large as a quarter can be inserted). Since neither of these is particularly convincing -- for example, I would have noticed if the coin slot were in fact blocked -- we have reason to assume that "rejects quarters" is not best seen as being directly attributable to either the device alone or the fault alone, but rather is the result of a complex interaction. In such a case, we must attempt to isolate features of the bug at a finer level of detail -- e.g., that different denominations of coins are being discriminated -- which may be attributable to either the normal device alone or to the underlying fault alone.

Depending on which attribution is made, differing constraints will be imposed any subsequent explanation. If it is assumed that the characteristic in question results from the underlying fault, then the constraint is imposed by the single-fault assumption: all buggy behavior explained in this way must be traced to a single underlying cause. If, on the other hand, a property of the correctly functioning device is adduced to account for a characteristic, then the constraint is imposed by another domain-independent debugging rule, the *teleological* assumption. This rule asserts that any hypothesized property of a properly functioning device must have a teleological justification. We always assume that every component of the mechanism must serve some purpose, and if no purpose can be found for some component which is hypothesized to play a role in explaining a bug, then the explanation is probably wrong.

A critical feature of the buggy behavior which must be explained in this example is the fact that different denominations of coins are treated differently by the first bug, so that quarters are rejected while nickels and dimes are not. Applying the procedure above, we must decide whether this distinction is solely a property of the bug, or whether it is owing to a discrimination that the device would make under normal circumstances anyway. Would discriminating among the different denominations of coins serve some purpose in the normal function of the vending machine? In fact we know such a distinction must be present in the SENSOR-COUNTER subsystem, since the physical differences between coins indicate the differences in the amount of money they represent. If we assume that such a distinction is also present in the SLOT-RESERVOIR subsystem (where the bug resides), then the design of the sensor is simplified, since a single complex sensor which must distinguish among different denominations can then be replaced by three simple sensors which merely indicate the presence of a coin. Such a simplification provides evidence that the assumption is plausible. But it should be clear that deducing this consequence requires fairly sophisticated knowledge about the desirability of and means for accomplishing complex sensing by separating the functions of discrimination and detection.

Upon assuming that the machine is in fact designed to make such a discrimination in the SLOT-RESERVOIR subsystem, then we can extend our model, using an additional SORT component capable of routing distinct items into distinct paths:



Another aspect of the buggy behavior is the fact that quarters come out of the machine at the coin return slot. Disregarding (for the moment) the *explanation* for this, the mere fact of it implies that there is a deficiency in the model shown above: there is no path for the quarters to travel between the SLOT-RESERVOIR subsystem and the coin return. This requires a PTRANS link from some point in the SLOT-RESERVOIR subsystem to the coin return. This link must connect with the SLOT-RESERVOIR subsystem before the quarter sensor, since otherwise quarters would be counted. Furthermore, using the single fault assumption we can infer that whatever is wrong with the quarter sensor is precluding quarters from taking the normal path through it, forcing them instead to take this alternative path to the coin return. The obvious explanation, of course, is that something is *stuck* in the quarter sensor. Exactly why this is obvious is not clear. We can only conclude that there exists a BLOCKED PATH memory structure which has been indexed by the problem description "a problem ahead in a path is precluding something from continuing along that path," and which indicates what the problem might be depending on the nature of the path. In particular, for physical paths, the BLOCKED PATH structure must indicate that the problem might be that something is stuck in the path.

At this point, we have a complete explanation of why the vending machine is rejecting and ignoring quarters. However, the explanation has not yet been adequately tied to the teleology of the machine, since we do not yet know the purpose of the alternative path the quarters are taking to the coin return. Our explanation of this particular aspect of the buggy behavior will not be complete until we have satisfied the constraint of teleological justification.

In this case, the particular purpose which I inferred was suggested by knowledge about the SORT component of the model. Often when sorting items, one wants an "other" category to take care of objects which have failed the sort. This would be a useful for a vending machine to handle the probability that objects other than the appropriate coins will be deposited; the action of routing such things to the coin return would serve the purposes of removing unwanted items from the machine, and returning them to their owners. Thus our explanation of the first bug meets the teleological constraint.

We turn now to the second bug, which is that the machine is dispensing Cokes for only 30 cents, rather than the usual 55 cents. From the single fault assumption, we already determined that the problem was located at the quarter sensor. Furthermore, the BLOCKED PATH memory structure enabled the additional conclusion that something was stuck in the quarter sensor, in order to explain the first bug. If, again invoking the single fault assumption, we assume that the rest of the SENSOR-COUNTER subsystem functions properly, the problem must be that the quarter sensor is reporting that a quarter has been collected which has in fact not been collected.

The question then is what is causing this to happen? The obvious answer for what might cause a quarter sensor to report a quarter is: a quarter. In this case it would have to be a quarter which was not in fact collected. Since we have already concluded that the problem is something stuck in the quarter sensor, we can now conclude that what is stuck in the sensor is in fact a quarter, and *this* is the quarter which the machine counts once towards the price of each Coke dispensed.

We are now almost finished with respect to the two issues which have driven our construction of the model: each part in our model is teleologically justified, and we have explanations of the bugs which people generally deem adequate. In fact, the mechanisms and explanations match quite closely what we set out to produce.

Step four: Building the machine

Given that we are satisfied with our model's explanatory and teleological characteristics, one question remains: can we build it? That is, can we think of actual physical mechanisms which will do the jobs we have assigned to the various components?

Some of the implementations are straightforward: the PTRANS connections can be viewed as tracks down which the

coins roll or slide. The SORT process can be implemented by attaching chutes of various sizes off of a main channel, arranged smallest to largest, such that the coins fall into the chute matching their size. Most of the protocol subjects imagined this kind of sorting device, and most got the idea from remembering children's piggy banks which sort coins in this fashion. Thus the SORT component allows us to index this memory, given that coins are the items to be sorted. Other components, such as the counter, were harder.

The real significance of the attempt to actually "construct" the model, however, lies in the fact that conjectures about the physical implementations of device components not only raise additional demands to elaborate the model (when it seems hard or impossible to implement it as is), they also suggest possible elaborations of the model through fortuitous remindings of additional uses a device component may serve, once it has already been proposed for a different use.

Conclusion

Let us consider again the important points made about the process of explanation in this domain:

1. Teleological knowledge must be transformed into a device model consisting of teleologically-defined components connected by the appropriate causal links.
2. Domain-independent knowledge of debugging must be used to locate the source of the problem in the model.
3. The model must be sufficiently elaborated to explain all observed features of the bugs.
4. All components added to the model in order to explain the buggy behavior must be justified teleologically.
5. The teleological components of the model must, ultimately, be replaced with "real" components -- e.g., relays and wheels -- as a demonstration that the teleological model is implementable.
6. Knowledge about real components in memory must be indexed by the purposes with which they have been associated by experience; conversely, given a real component, one must be able to access knowledge about its possible purposes.

Two striking facts seem to emerge from this analysis. First, it is surprising how elaborate and detailed a model can be constructed in a situation where so little is initially known. When teleology and observed bugs are treated as twin constraints on the process of model building, complex and fairly convincing models can be constructed given only a rudimentary knowledge of device components. Second, given the simple representation constructed in step one, plus the single fault assumption, it seems difficult *not* to conclude that the problem lies in the quarter sensor, which is the cornerstone of the ultimate explanation. However, many people failed to reach this conclusion, despite the fact that they at least considered using the single fault assumption in attacking the problem. This argues that quite a bit of power lies in the simple representation scheme illustrated here.

Acknowledgments: I thank Larry Birnbaum for useful discussions and for comments on an earlier draft of this paper.

References

de Kleer, J. and Brown, J.S. 1981. Mental Models of Physical Systems and their Aquisition. In J. Anderson (eds), Cognitive Skills and their Aquisition, Erlbaum.

Forbus, K., and Stevens, A. 1981 Using Qualitative Simulation to Generate Explanations. Technical Report 4490, Bolt Beranek and Newman.

McDermott, D.V. 1976. Flexibility and Efficiency in a Computer Program for Designing Circuits. Technical Report TR-402, Massachusetts Institute of Technology.

Schank, R.C. and Abelson, R. 1977. Scripts, Plans, Goals and Understanding. Lawrence Erlbaum Associates, Hillsdale, New Jersey.

Schank, R.C. and Collins, G.C. 1982. Looking at Learning. In Proceedings of the ECAI, pages 10-16. ECAI, Paris, France.

Simon, H.A. 1969. The Sciences of the Artificial. The M.I.T. Press.

Stevens, A. and Collins, A. 1978. Multiple Conceptual Models of a Complex System. Technical Report 3923, Bolt Beranek and Newman.

Sussman, G.J. and Brown, A.L. 1974. Localization of Failures in Radio Circuits: A Study in Causal and Teleological Reasoning. Technical Report AIM-319, Massachusetts Institute of Technology.

Step four: Building the machine

Given that we are satisfied with our model's explanatory and teleological characteristics, one question remains: can we build it? That is, can we think of actual physical mechanisms which will do the jobs we have assigned to the various components?

Some of the implementations are straightforward: the PTRANS connections can be viewed as tracks down which the coins roll or slide. The SORT process can be implemented by attaching chutes of various sizes off of a main channel, arranged smallest to largest, such that the coins fall into the chute matching their size. Most of the protocol subjects imagined this kind of sorting device, and most got the idea from remembering children's piggy banks which sort coins in this fashion. Thus the SORT component allows us to index this memory, given that coins are the items to be sorted. Other components, such as the counter, were harder.

The real significance of the attempt to actually "construct" the model, however, lies in the fact that conjectures about the physical implementations of device components not only raise additional demands to elaborate the model (when it seems hard or impossible to implement it as is), they also suggest possible elaborations of the model through fortuitous remindings of additional uses a device component may serve, once it has already been proposed for a different use.

Conclusion

Let us consider again the important points made about the process of explanation in this domain:

1. Teleological knowledge must be transformed into a device model consisting of teleologically-defined components connected by the appropriate causal links.
2. Domain-independent knowledge of debugging must be used to locate the source of the problem in the model.
3. The model must be sufficiently elaborated to explain all observed features of the bugs.
4. All components added to the model in order to explain the buggy behavior must be justified teleologically.
5. The teleological components of the model must, ultimately, be replaced with "real" components -- e.g., relays and wheels -- as a demonstration that the teleological model is implementable.
6. Knowledge about real components in memory must be indexed by the purposes with which they have been associated by experience; conversely, given a real component, one must be able to access knowledge about its possible purposes.

Two striking facts seem to emerge from this analysis. First, it is surprising how elaborate and detailed a model can be constructed in a situation where so little is initially known. When teleology and observed bugs are treated as twin constraints on the process of model building, complex and fairly convincing models can be constructed given only a rudimentary knowledge of device components. Second, given the simple representation constructed in step one, plus the single fault assumption, it seems difficult *not* to conclude that the problem lies in the quarter sensor, which is the cornerstone of the ultimate explanation. However, many people failed to reach this conclusion, despite the fact that they at least considered using the single fault assumption in attacking the problem. This argues that quite a bit of power lies in the simple representation scheme illustrated here.

Acknowledgments: I thank Larry Birnbaum for useful discussions and for comments on an earlier draft of this paper.

References

de Kleer, J. and Brown, J.S. 1981 Mental Models of Physical Systems and their Aquistition. In J. Anderson (eds), Cognitive Skills and their Aquisition, Erlbaum.

Forbus, K., and Stevens, A. 1981. Using Qualitative Simulation to Generate Explanations. Technical Report 4490, Bolt Beranek and Newman.

McDermott, D.V. 1976. Flexibility and Efficiency in a Computer Program for Designing Circuits. Technical Report TR-402, Massachusetts Institute of Technology.

Schank, R.C. and Abelson, R. 1977. Scripts, Plans, Goals and Understanding. Lawrence Erlbaum Associates, Hillsdale, New Jersey.

Schank, R.C. and Collins, G.C. 1982. Looking at Learning. In Proceedings of the ECAI, pages 10-16. ECAI, Paris, France.

Simon, H.A. 1969 The Sciences of the Artificial. The M.I.T. Press.

Stevens, A. and Collins, A. 1978. Multiple Conceptual Models of a Complex System. Technical Report 3923, Bolt Beranek and Newman

Sussman, G.J. and Brown, A.L. 1974. Localization of Failures in Radio Circuits: A Study in Causal and Teleological Reasoning. Technical Report AIM-319, Massachusetts Institute of Technology.

A MODEL FOR UNDERSTANDING THE POINTS OF STORIES*

Marcy Dorfman

Coordinated Science Laboratory and
Center for the Study of Reading
University of Illinois at Urbana Champaign
Urbana, IL 61801

ABSTRACT

This paper describes a proposal and some preliminary evidence in support of a model for understanding the points of simple stories. The proposed model differs from existing systems in that it includes, in addition to a representation of the plans and goals of each of the story characters, a model of the beliefs and intentions of the author and the reader. It is hypothesized that readers use story-specific information in conjunction with their own beliefs about the story events in order to make inferences relevant to the point of the story that the author intended. Evidence from adult readers is presented in support of each of the components of the model and their interaction. The proposed model has relevance for psychological and computational research on story understanding. The work also has implications for more general discourse situations in which understanding is predicated on the knowledge of shared beliefs.

1. INTRODUCTION

During the past decade, a great deal of research in cognitive science has been devoted to understanding the nature of stories and story understanding abilities. Some of this work has focused specifically on the plans and goals of story characters [Rumelhart, 1977; Schank and Abelson, 1977; Bower, 1978; Wilensky, 1978], while other research has attempted to describe characteristics that make stories interesting [Wilensky, 1983], enjoyable [Brewer, 1982; Brewer and Lichtenstein, 1982], and thematically significant [Dyer, 1982; Lehnert, *et al.*, 1982; Schank, *et al.*, 1982]. Despite recent progress in analyzing stories, current story processing models have been concerned far more with the plans, goals, motivations, and emotions of story characters than with the beliefs and intentions of readers and authors. An important assumption underlying the research presented here is that, in order to understand the *thematic content* or the *point* of a simple story, a human reader or a story understanding system must be able to represent its own beliefs, as well as the beliefs and intentions of the story's author. In this paper, we present a model of story understanding that consists of several kinds of information that readers use in making inferences relevant to the points of stories. Following a description of the model, we discuss the results of several experiments that provide tentative support for each of the components of the model and their interaction.

2. DESCRIPTION OF THE MODEL

2.1. Story Outcome Component

In order to understand the points of simple stories, a reader or a story understanding system must be able to infer the author's purpose in writing stories. Consequently, an important component of any theory capable of generating the points of stories is a model of what the author thinks or believes. In didactic stories, such as fables, the author's beliefs are usually reflected by the story's outcome, or by the positive or negative consequences experienced by each of the characters at the conclusion of the story. For example, in the fable *The Tortoise and the Hare*, the reader can infer from the story's outcome that the author believes that it is morally correct to be diligent and hard-working like the tortoise, and morally wrong to be boastful and over-confident

* This research was supported in part by the Air Force Office of Scientific Research under grant F49620-82-K-0009.

like the hare. Given this characteristic of didactic stories, it is possible to account for many aspects of story points by a theory that focuses on the story's outcome. Within this framework, a simple procedure for generating the points of stories can be described as follows:

(1) IDENTIFY OUTCOME: Identify the positive or negative outcome of each story character.

(2) IDENTIFY ACTION: Identify the morally significant action(s) that lead to the story outcome.

(3) GENERALIZE: Provide a general description for each morally significant action that occurs in the story.

(4) STATE POINT: Assert that people should perform morally significant actions that have positive outcomes, and should not perform morally significant actions that have negative outcomes.

A representation of *The Tortoise and the Hare* in terms of this simple framework is shown in Figure 1.

CHARACTER	OUTCOME	SPECIFIC ACTION	GENERAL ACTION	POINT
TORTOISE	+	Persistent in running a race against the hare.	DILIGENT	BE DILIGENT
HARE	–	Negligent in running a race against the tortoise.	OVER-CONFIDENT	DON'T BE OVER-CONFIDENT

Figure 1. An outcome-based analysis of *The Tortoise and the Hare*

2.2. Just World Component

In section 2.1 we described a simple procedure for generating the points of stories based on information relevant to the story's outcome. While this procedure is adequate for many simple stories, an outcome-based approach does not always generate the best point for stories in which the outcome-based point is inconsistent with the point inferred to be the author's. For example, consider the case in which the outcome of *The Tortoise and the Hare* is reversed, so that the hare rather than the tortoise wins the race. Based on the outcome of each character, the story outcome component would generate the following set of points:

PEOPLE SHOULD BE OVER-CONFIDENT
PEOPLE SHOULD NOT BE DILIGENT.

Given this set of points, the reader can make one of two assumptions:

(1) the outcome-based point is the point of the story that the author actually intended;

(2) the outcome-based point is *not* the point of the story that the author actually intended. The author's real purpose in writing the story is to point out that there are instances of moral injustice in the world. That is, sometimes lazy and over-confident people are rewarded, while hard-working and diligent people are not.

In order to arrive at the latter, or *ironic* interpretation of the story, the reader must compare the actions and outcomes of each of the story characters to his or her own beliefs about moral justice. Since the model described here is intended to account for the comprehension of didactic stories, such as fables, the reader's belief system is based on Lerner's [1980] concept of a *just world*. In the context of the story world, the just world hypothesis predicts that stories will be morally satisfying when "good" characters experience positive outcomes and "bad" characters experience

negative outcomes. Unlike the simple procedure based on story outcome, the just world component generates story points by evaluating story outcome information relative to the positive or negative valence of each of the character's actions. An example in which the story outcome component and the just world component interact to produce an ironic story point is provided below.

2.3. Story Outcome vs. Just World Interpretation

Ironic story points result when the action-outcome mappings represented by the story outcome component are inconsistent with the action-outcome mappings represented by the just world component. For example, in the reversed-outcome version of *The Tortoise and the Hare*, the hare is rewarded for being boastful and over-confident, and the tortoise is punished for being hard-working and diligent. Since the relationship between action valence and outcome valence is inconsistent with the beliefs specified by the just world component, the model concludes that the author intended the text to be ironic, and produces a story point that reflects the immorality or injustice of the story's outcome. A schematic representation of the interaction between the two components is shown in Figure 2.

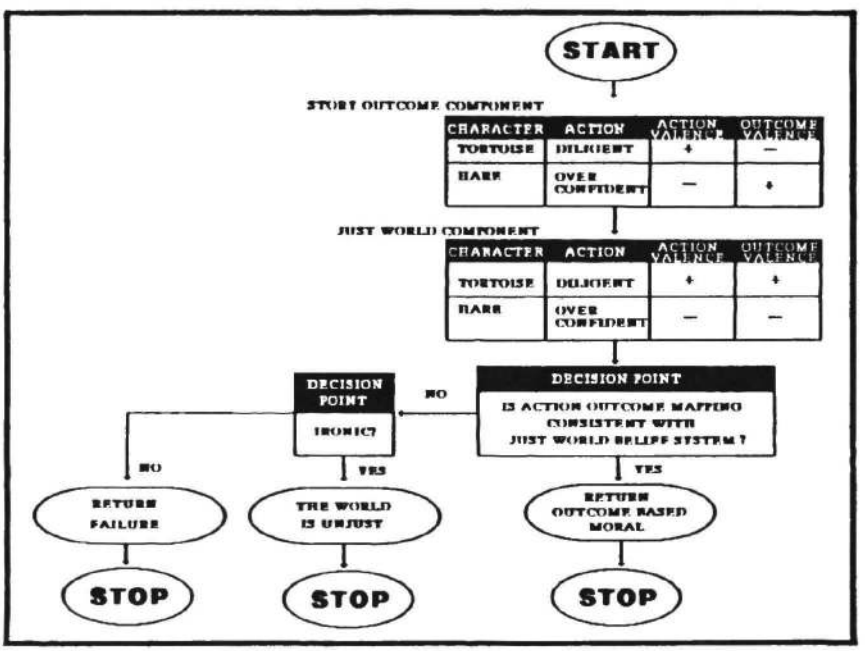


Figure 2. An ironic reading for the reversed-outcome version of *The Tortoise and the Hare*.

3. SUMMARY OF EXPERIMENTAL FINDINGS

In order to test the proposed model, we asked subjects to read and state the points of three types of stories. Type 1, or *canonical* stories, were selected from the Penguin edition of *Fables of Aesop* [1982] by two coders who independently rated each story for clarity of point. In each of the Type 1 stories, positive or negative action valence was consistently paired with the positive or negative outcome valence for each of the characters in the story. These stories served as the base narratives for the manipulated story types. Type 2, or *reversed-outcome* stories, were constructed stories in which the outcome of the focus character was reversed. In these stories, good characters were punished for performing good deeds (like helping another animal in trouble), and bad characters were rewarded for doing evil deeds (like being greedy or deceptive). Thus, in Type 2 stories, character valence and outcome valence were inconsistently paired. Type 3, or *no point* stories, were constructed by pairing a neutral event sequence with a positive or negative outcome for each of the focal characters in the story. In Type 3 stories, there was no causal relationship between the actions performed in the story and the positive or negative outcomes for each of the

story characters. As a result, type 3 stories were predicted to be the most unnatural story type for the reader.

If readers use information relevant to action valence and outcome valence in generating the points of stories, we would expect subjects to have more difficulty stating the points for Type 2 and Types 3 stories than for Type 1 stories. Table 1 indicates that this prediction is correct. A greater percentage of subjects were able to generate points for Type 1 stories than either of the manipulated story versions. In addition, subjects found the reversed-outcome and no-point stories less clear, less prototypical, and less likable than their canonical counterparts. A possible interpretation of the results is that subjects were simply better at generating points for stories with which they were familiar. However, an analysis of subjects' familiarity ratings indicates that the percentage of yes/no responses produced by the subjects does not vary as a function of familiarity, and that subjects are more likely to be responding to consistencies or inconsistencies in action valence and outcome valence included in the test stories.

STORY TYPE	YES (%)	NO (%)
CANONICAL (TYPE 1)	97.11	2.89
REVERSED-OUTCOME (TYPE 2)	77.33	22.67
NO-POINT (TYPE 3)	45.50	54.50

Table 1. The percentage of adults who stated points for each story type.

Based on the proposed model, we also predicted that readers would be more likely to generate *ironic* interpretations when the mapping between action valence and outcome valence was inconsistent with the reader's beliefs about moral justice. Table 2 lists percentages of point types for Type 1 and Type 2 stories. Subjects' points were coded as *outcome-based* if the point of the story was based solely on information relevant to the story's outcome. Points were coded as *ironic* if the reader's point was in opposition to the outcome-based point, or commented on the immorality or injustice of the story's outcome. Points that were neither outcome-based nor ironic were coded as *other*, and cases in which readers were not able to state a point for the text were coded as *no point*.

STORY TYPE \ POINT TYPE	OUTCOME-BASED	IRONIC	OTHER	NO POINT
CANONICAL (TYPE 1)	80.55	1.38	15.27	2.77
REVERSED-OUTCOME (TYPE 2)	30.55	31.94	12.50	25.00

Table 2. The percentage of point types across each story type.

Table 2 indicates that most readers generated outcome-based points for the Type 1 stories, in which the story outcome component and the just world component were in agreement. However, in the case of the Type 2 stories, at least as many subjects generated ironic points as outcome-based points. This finding suggests that, when action valence is inconsistent with outcome valence, the reader's beliefs play a significant role in evaluating the author's intentions and in making inferences relevant to the point of the story.

4. CONCLUSION

We have presented a model for understanding the points of simple stories that departs from previous research in some important ways. While existing models have focused on the plans and goals of story characters, the model presented here uses story-specific information to make inferences relevant to the author's purpose in writing the story. In order to reconstruct the author's intentions, the reader is hypothesized to rely on information relevant to the positive or negative actions of the story characters, the positive or negative valence of the story outcome, and his or her own beliefs about the universe of action described in the story. While this work is restricted to a highly specific domain, the mechanisms described in the model and supported by the data seem to characterize an important part of the point generation process. The findings have implications for psychological and computer-based models of story understanding. The results are also relevant to more general discourse situations in which the interpretation of meaning is contingent on the knowledge of shared beliefs.

5. ACKNOWLEDGEMENTS

I wish to thank Professor William Brewer for his comments on an earlier version of this paper.

6. REFERENCES

Bower, G. H. [1978] Experiments on story comprehension and recall. *Discourse Processes, 1*, 1978, 211-231.

Brewer, W. F. [1982]. Plan understanding, narrative comprehension, and story schemas. *Proceedings of the National Conference on Artificial Intelligence*, 262-264.

Brewer, W. F., & Lichtenstein, E. H. [1982]. Stories are to entertain: A structural-affect theory of stories. *Journal of Pragmatics, 6*, 473-486.

Dyer, M. G. [1982]. *In-depth understanding*. Cambridge: MIT Press.

Fables of Aesop. [1982]. New York: Penguin Books.

Lehnert, W. G., Dyer, M. G., Johnson, P. N., Yang, C. J., & Harley, S. [1983]. BORIS - An experiment in in-depth understanding of narratives. *Artificial Intelligence, 20*, 15-62.

Lerner, M. J. [1980]. *The belief in a just world: A fundamental delusion*. New York: Plenum Press.

Rumelhart, D. E. [1977]. Understanding and summarizing brief stories. In D. LaBerge and S. F. Samuels (Eds.), *Basic processes in reading: Perception and comprehension*. Hillsdale, NJ: Erlbaum.

Schank, R. C., & Abelson, R. P. [1977]. *Scripts, plans, goals, and understanding*. Hillsdale, NJ: Erlbaum.

Schank, R. C., Collins, G. C., Davis, E., Johnson, P. N., Lytinen, S., and Reiser, B. J. [1982]. What's the Point? *Cognitive Science, 6*, 255-275.

Wilensky, R. (1978). Understanding goal-based stories. New Haven, CT: Yale University Department of computer Science, Research Report No. 140.

Wilensky, R. (1983). Story grammars versus story points. *The Behavioral and Brain Sciences, 6*, 579-623.

A Framework for Concept Formation[1]

J. Daniel Easterlin
Pat Langley

Irvine Computational Intelligence Project
Department of Information and Computer Science
University of California, Irvine 92717

INTRODUCTION

Our approach to concept formation differs from the traditional view. In this paper, we outline an alternative view of concept learning, and argue that the goals of the learner play a central role in this process. We propose that goals act to determine significant features of the world, and that without such goals as a basis, concept formation is a semantically empty data summarization task. We begin by examining the components of the concept formation process. After laying this foundation, we review previous approaches to concept formation in these terms, rejecting two of the assumptions upon which this work has been based – the presence of a tutor and the "all–or–none" character of concepts. This leads us to propose an alternative model of the concept formation process, in which goals and prototypes figure prominently.

THE COMPONENTS OF CONCEPT FORMATION

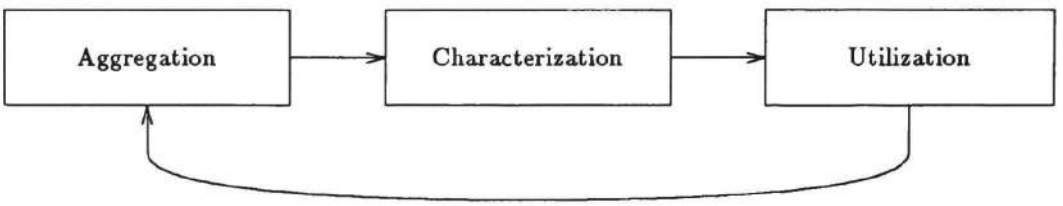


Figure 1. The Components of Concept Formation.

Previous research in machine learning suggests that the process of concept formation can be divided into three distinct components. The first of these – aggregation – involves grouping instances of the concept into collections. The second component – characterization – involves generating some description of the instances in the aggregate. The final subprocess – utilization – involves making use of the resulting description. Let us examine each of these components in more detail.

Aggregation

Aggregation is a process of collection, in which objects or instances of some concept (possibly still to be learned) are grouped together into a set. Aggregation is not a process of description, but involves collecting entities into an aggregate, from which a description or characterization can subsequently be formed. In the task of learning from examples as studied by machine learning researchers, the aggregation process is made trivial (Hunt, Marin & Stone, 1966). The tutor provides explicit aggregation of the examples into sets of positive and negative instances, and some characterization of the positive instances is generated. In contrast, in the task of learning search heuristics, aggregation must be performed by the learning system itself (Mitchell, Utgoff & Banerji 1983). Instances that led to the successful solution of a problem are aggregated as positive instances of the responsible rule's use, while instances that led away from the solution path are aggregated as negative instances. Thus, aggregates are generated by the learner on the basis of performance, rather than relying on a tutor, as in the case of learning from examples. The utility of discussing aggregation as a distinct process in concept formation is that it focuses attention on what constitutes significance for the system. As a result, one begins to question the plausibility of existing aggregation techniques, and to explore alternative methods.

Characterization

Characterization is the process that is usually discussed in machine learning under the name "concept learning". It involves constructing a description for an aggregate of entities, based on individual descriptions of each entity. This may occur either incrementally or non–incrementally, depending on whether instances are presented simultaneously or one at a time. Researchers in machine learning have proposed a number of computational methods for characterization, and these constitute their contributions to concept formation.

[1] This research was supported in part by the IBM Corporation, and in part by a gift from Hughes Aircraft Company.

Utilization

The utilization process integrates the characterization or concept description with the performance element. In the case of recognition, it contains a matching process for identifying positive instances of the description that were constructed during the characterization process. Following recognition, some action may be taken or a metric may be applied to test the adequacy of the description in recognizing the instance. For the most part, concept descriptions are used to recognize positive instances of the concept when they occur in the problem domain. Since most AI learning research has assumed "all–or–none" concepts, the recognition process has typically involved a "complete matching" mechanism, in which all conditions must be satisfied before instances of the concept are recognized.

TRADITIONAL APPROACHES TO CONCEPT FORMATION

We are interested in concept formation as it occurs in complex, reactive environments that are similar to the real–world. In this section, we review the components of the concept learning process, and find that such environments lead one to reject some important assumptions upon which earlier machine learning work has been based.

Tutors and Aggregation

Previous research on concept learning has assumed careful guidance by a tutor, despite the intuition that humans learn most of their concepts through experience with the world. For example, children clearly learn concepts such as "dog" and "chair" before they know the words for these concepts. In the traditional approach to learning concepts from examples, a tutor trivializes the aggregation problem, by providing positive and negative instances of the concept to be learned. In contrast, learners in the real world must aggregate instances in some other manner.

All–or–None Concepts and Characterization

According to the classical view described by Smith and Medin (1981), a concept is defined by necessary and sufficient conditions, and machine learning researchers have used a similar notion of concepts. In other words, for an object to be recognized as a positive instance of some concept, it must satisfy all of the conditions specified in the concept description, and additional features have no effect. However, most of our everyday concepts are "fuzzy", with exemplars being better or worse, rather than instances or non-instances (Rosch & Mervis, 1975). For instance, a robin is a better instance of the concept "bird" than a penguin, and some chairs are better than others (e.g., ones that are missing a leg). Such concepts cannot be described in terms of necessary and sufficient conditions, so some other representation is required.

Smith and Medin (1981) have outlined two alternatives to the all–or–none framework. The "prototype" (or "probabilistic") approach, first proposed by Rosch and Mervis (1975), assumes that there exists an abstract representation of each concept, and that instances are judged to be better or worse examples depending on the *degree* to which they match this representation. Smith and Medin also discuss the *exemplar* approach, in which concepts are represented not as abstract structures, but as disjunctions of many specific instances. Both approaches have their advantages, and evidence exists for both theoretical frameworks. In this paper, we will focus on the prototype–probabilistic approach for a simple reason – this approach is computationally much more tractable.

The most common AI methods for characterization are generalization and discrimination. Upon closer examination, we find that these methods encounter serious difficulty when applied to "fuzzy" concepts. The problem is that both methods rely on a strong distinction between positive and negative instances: generalization finds structures held in common among positive instances, and discrimination finds differences between positive and negative instances.

Complete Matching and Utilization

Recognition involves determining the most appropriate concept to describe the current data. This task is considerably simplified by the assumption that concepts are defined by a set of necessary and sufficient conditions. However, we have already argued that real–world concepts cannot be defined in this manner. Thus, complete matching of object to characterization must be rejected.

AN ALTERNATIVE APPROACH TO CONCEPT FORMATION

In the previous section, we argued that real–world concepts are not learned from a tutor, and that they cannot be described in terms of necessary and sufficient conditions. If we hope to account for the process of concept formation, this forces us to propose new techniques for aggregating sets of instances, for characterizing the resulting aggregates, and for recognizing the best concept for a given situation.

Goals and Aggregation

By rejecting the traditional assumptions of tutor-provided instances, we must find some other solution to the problem of aggregation. We believe that the *goals* of the learner play a major role in this process, and presume that at each point in time, the agent has one or more active goals (possibly organized as a hierarchy of goals and subgoals). In describing their means–ends analysis theory of human problem solving, Newell and Simon (1972) distinguish between three types of goals. Although each of these goal types can be used to direct the aggregation process, the most obvious examples involve apply–operator goals, in which one wants to apply an operator to some object or state. For instance, suppose the agent is tired, and decides to apply the operator *sit–down*.† This operator requires some object upon which to sit, and the agent will scan its immediate environment for a likely candidate. The important point is that by applying its operator to candidate objects, the agent will discover that some objects produce better results than others. These will be good instances of "sittable" objects, while others (such as chairs with wobbly legs, or with a tack on their seat) will be poor instances. In any case, objects to which the operator has been applied that more or less satisfy the goal are passed on to the characterization process.

Our attention to the importance of goals arises primarily from recognizing aggregation as a distinct process demanding a supporting basis. As this basis, goals play a dual role – they identify which objects should be grouped together for input to the characterization mechanism, and they provide a test indicating the degree to which the desired state has been achieved. Thus, they tie objects and operators to experience by indicating their relative value in satisfying goals. Certain objects and operators are rated higher than others, since the application of particular operators to particular objects manifests properties of those objects that are instrumental in satisfying the posted goals, while others are not. Objects are thus rated higher to the degree they manifest functional properties in achieving goals. This provides the feedback necessary to identify some combinations of objects and operators in experience as more significant to the learner than others.

In summary, we believe that the learner's goals direct the aggregation process. Furthermore, objects and operators are grouped together according to the degree to which their interaction satisfies goals. It is through this interaction between operators and objects that objects manifest properties which are functional in satisfying goals. Thus, not only do goals identify significant objects and operators, but they further suggest the existence of significant functional properties within an object.

The Representation of Concepts

Traditional approaches to characterization assume the all–or–none nature of concepts, which simply does not hold for many everyday object concepts. As a result, we must find another solution to the characterization problem. Our approach must be able to represent "fuzzy" concepts, and to incrementally modify these descriptions in response to new instances of the concept. In real–world concepts, some features and relations are more important than others. Thus, our representation must include some measure of each feature's *criteriality*. We specify this in terms of a weight ranging from zero to one, with zero denoting low importance and one indicating high importance. Of course, these numbers have little meaning detached from the utilization process. In our framework, conditions (features or relations) with high weights contribute more to the overall degree of match than conditions with low weights. Moreover, conditions with very high weights (near one) must be matched for a reasonable overall match to result. As a result, the notion of all–or–none concepts emerges as a special case of this scheme, in which all conditions have weights of one.

Since we are concerned with object concepts, we believe that structures similar to Binford's (1971) generalized cylinders will prove adequate. This representation has the advantage of combining structural relations between the components of an object with numeric features of those components. This is an important characteristic, since the real–world has both structural and numeric aspects. For instance, a prototypical chair might be represented with the components of four legs, a seat, and a back arranged in particular spatial relations to each other. In addition, each component would be described by numeric features, such as length, diameter, and orientation (normalized for the overall size of the object). In addition, the use of numeric features leads to a novel interpretation of the weights on each feature. With each numeric feature, one can associate a *mean* value of the positive instances that have been observed, and a *standard deviation* of those values. High standard deviations imply that a wide range of values of the feature are satisfactory, while low standard deviations imply that only a narrow range of values for the feature is acceptable. Thus, one might use the *inverse* of the standard deviation for a feature as its associated weight. This would give low criteriality to features with widely varying values, and high criteriality to features with nearly constant values. For instance, the legs of a chair are nearly always half the length of the entire chair's

† Obviously, the action *sit-down* is not primitive in any sense; it is a high–level operator (or macro–operator) that must be acquired from experience.

height. Thus, this feature would have a low standard deviation and be highly criterial, giving it an important role in judgements of prototypicality.

However, recall that we are assuming the characterization process receives more than prototypical instances as input. Rather, it is given the degree to which each object satisfies the agent's goals. We would like our learning mechanism to use this information in creating the concept description. In order to do this, we require more than a feature's mean value; we require a function relating features and operators to the "goodness" of an object in satisfying goals. We propose constructing this function by considering operators in addition to physical features as relevant to the object's goodness and including them as a special type of feature in the concept description. By regressing goodness against the values of all feature types, we derive a relation between important physical features of the object, those operators that operate on the features, and values of goal satisfaction. Thus, we have a description of the object that expresses the object's functional properties as they relate to goal satisfaction and the object's physical characteristics for use in recognition. The representation is an equation providing both a means for predicting values of goal satisfaction and a measure of the goodness of fit for such predictions. Returning to our interest in the criteriality of features for predicting goal satisfaction, the percentage of variance in goal satisfaction accounted for by an individual feature can be taken as a measure of that feature's criteriality in the concept definition.

The Characterization Process

Let us now turn to the mechanism of characterization, by which the learner goes from instances of some concept to a description of that concept. Within the current framework, we are assuming that the aggregation process has determined which object and operators should be incorporated into the concept description, and that aggregation also provides the degree to which the object and operators satisfy the relevant goal. The task of characterization is to modify the existing description to better predict the "goodness" of the current object. We also assume that instances are processed incrementally, since the agent generally interacts with one object at a time (or a few at most). Thus, each instance leads to only minor modifications in the concept description. Before a concept description can be altered, it must first be created, and issues arise about the nature of such initial descriptions. Since early descriptions are based on a single instance, one might make each feature very criterial by having a weight of one. Through experience, as additional instances are observed and variation among feature values (including operators) occurs, constraints on the feature values become looser.

Once a stable description has been formed, the feature values of new instances are used to modify the regression coefficients associated with each feature. By retaining the number of instances that have been observed so far, one can easily compute a revised equation that includes the new feature value. This accommodates gradual changes in the concept description over time. For example, if the learner began to see chairs with longer legs, his coefficients for the "length of leg" features would slowly be revised. Thus, this method can respond to changing environments, unlike most traditional approaches to concept learning.

However, if the agent encounters an object with feature values that fall far outside previous experience, this is an occasion to generate a disjunctive version of the current concept. For instance, if one sees a chair in which the legs are substantially longer than expected (such as a baby's high–chair), then it is natural to distinguish this from other chairs that more closely match one's expectations. Such variants are stored near to the initial concept, but are characterized independently of the original version. Note that this implies the order of presentation is relevant to learning. If gradual changes in feature values are observed, a single concept will be learned; however, if instances with extreme values are alternated, disjunctive concepts will be acquired instead.

Goal-Indexed Partial Matching and Utilization

Traditional AI approaches to concept learning assume that complete matching can be used for recognition. However, in rejecting the notion of necessary and sufficient conditions, we are inevitably led to replace this with some form of partial matching mechanism. Hayes–Roth (1978) has argued that partial matching is computationally expensive, and the best known algorithm is exponential in the general case. Therefore, we would like to take advantage of constraints to make the task manageable.

Recall that we are assuming different weights on the various conditions composing the concept description. To a certain extent, we can constrain the partial matching process by attempting to match more criterial conditions (those with higher weights) first, and leaving less criterial features and relations until later. This leads to a best–first search through the space of partial matches, and is much more attractive than an exhaustive version. Like all heuristic search approaches, the method is not guaranteed to find the optimal solution (in this case the best partial match), but it will nearly always find a satisfactory one with considerably less effort.

However, recall also that the agent must choose between hundreds and thousands of competing concepts, and it is unlikely that the above method will suffice. Fortunately, in this framework concepts are created because their instances have been instrumental in achieving the learner's goals. Thus, it is natural to organize concepts around the goals they help satisfy. If we index concepts by the goals with which they are associated, then the agent can use its currently active goals as probes to retrieve potentially relevant concepts. As a result, the partial match is constrained to those concepts likely to aid in achieving the current goal, presumably a few instead of thousands.

Since it is central to the recognition process, we should say a little more about the partial matching mechanism. Given the description of some object and the characterization of some concept, the matcher returns a mapping between the two structures, along with the degree to which the match was successful. If the match was high, then the agent can infer that the object will prove ideal for satisfying the goal under which its concept was indexed. If the match is only fair, then it may still want to use the object, provided no better objects are found in the immediate vicinity. Furthermore, since information about which operators to apply is included in the object concept, guidelines for instrumental use of the object derive not from additional problem solving, but directly from the concept's content. Thus, inferences regarding the object's functionality co-occur with recognition.

Goals ⇒ Aggregation
Incremental weighting ⇒ Characterization
Goal–indexed partial matching ⇒ Utilization

Figure 2. An alternative approach to concept formation.

CONCLUSION

In the preceding pages, we identified three components of the concept learning process – aggregation, characterization, and utilization – and found that earlier work relied on two assumptions that made each of the tasks manageable. The first involved the presence of a tutor, who made the aggregation problem trivial by providing positive and negative instances of the concept to be learned. The second involved the notion that concepts are all–or–none in nature, so that they can be described by a set of necessary and sufficient conditions. Since we were concerned with concept formation in real–world settings, we rejected these two assumptions. However, this forced us to propose new methods for dealing with the three components of concept formation. In response, we proposed an alternative framework in which goals were used to aggregate experience. In this approach, goals are also used to index and retrieve potentially relevant concepts, reducing the task of partial matching against prototypes to reasonable proportions. Finally, we proposed a method for incrementally characterizing concepts, based on weighting numeric features and operators in terms of their observed variance. This gave us both physical criteria for recognizing future instances of the concept and information about the functional properties of objects. Taken together, we believe that these methods constitute a viable alternative to traditional approaches to concept formation, and in our future work we plan to instantiate the framework as a running system, and to test its learning abilities in a complex, reactive environment.

REFERENCES

Binford, T.O. (1971) Visual perception by computer. In *Proceedings IEEE Conference on Systems Science and Cybernetics.*

Hayes–Roth, F. (1978) The role of partial and best matches in knowledge systems. In D.A. Waterman & F. Hayes–Roth (Eds.) *Pattern–directed inference systems*, NY: Academic Press, 557–576.

Hunt, E.B., Marin, J., & Stone, P. (1966) *Experiments in induction.* NY: Academic Press.

Mitchell, T.M., Utgoff, P.E., & Banerji, R. (1983) Learning by experimentation: Acquiring and refining problem solving heuristics. In R.S. Michalski, J.G. Carbonell, & T.M. Mitchell (Eds.) *Machine learning*, Palo Alto, CA: Tioga Publishing Company, 163–190.

Newell, A., & Simon, H. (1972) *Human problem solving.* Englewood Cliffs, NJ: Prentice–Hall.

Rosch, E., & Mervis, C.B. (1975) Family resemblances: Studies in the internal structure of categories. *Cognitive Psychology* 7, 573–605.

Smith, E.E., & Medin, D.L. (1981) *Categories and concepts.* Cambridge, MA: Harvard University Press.

The Problem of Existence

Kenneth D. Forbus
Department of Computer Science, University of Illinois
1304 W. Springfield Avenue, Urbana, Illinois, 61801
Arpanet: forbus@UIUC

Abstract: Reasoning about changes of existence in objects, such as steam appearing and water disappearing when boiling occurs, is something people do every day. Discovering methods to reason about such changes in existence is a central problem in Naive Physics. This paper analyzes the problem by isolating an important case, called *quantity-conditioned existence*, and presents a general method for solving it. An example generated by an implemented program is exhibited, and remaining open problems are discussed.

1. Introduction An important feature of the physical world is that objects come and go. If we boil water steam appears, and if the boiling continues long enough the water completely disappears. Modeling changes in existence is a central problem in qualitative physics, yet most theories avoid it. de Kleer & Brown (1984) and Williams (1984) define it away by basing their formalisms on *system dynamics*. In system dynamics, the model builder constructs a network of "devices" to represent the system under study. Many systems are not represented naturally by system dynamics (such as boiling water), and they do not address the crucial issue of generating the initial device network from what a person sees when walking around in the everyday world. Kuipers (1984) represents a system by a collection of constraint equations; objects are only represented implicitly by the names chosen for variables in the equations, so his system provides no help on this issue either. Simmons (1984) provides a means of specifying that objects appear and vanish in his representation of occurrences of processes, but in a way that precludes discovering changes in existence not explicitly foreseen by the modeller. Weld (1984) provides a similar notion in his elegant theory of discrete processes, but with similar limitations. No general solution currently exists. Given the range of phenomena (including state changes, chemical reactions, and fractures in solids) this is not too surprising. This paper presents a solution to an important special case, based on the framework provided by Qualitative Process theory (Forbus 1981; 1984a). First we describe a general logic of existence, extending notions of histories introduced by Hayes (1979) and then introduce the idea of quantity-conditioned existence. Next we describe a *temporal inheritance procedure* for reasoning about changes in existence, and illustrate its operation with an example. Finally, we suggest a direction in which to look for further solutions to the problem of existence.

2. A Logic of Existence Objects in the world are represented by *individuals*. In general criteria for what constitutes an individual depends on the domain. *Histories* represent how objects change over time. The history of an object describes its "spatio-temporal extent" and is annotated with the properties that hold for the object at various times. We futher assume the extensions described in (Forbus 1984a).

We begin by distinguishing between two related notions of existence. The first is logical existence, which simply means that it is not inconsistent for there to be some state of affairs in which a particular individual exists. A square circle is something which logically cannot exist. The second notion is physical existence, which means that a particular individual actually does exist at some particular time. Clearly an individual which physically exists must logically exist, and an individual which logically cannot exist can never physically exist. An example of an individual which logically exists but which (hopefully) never physically exists is the arsenic solution in my coffee cup.

The predicate **Individual** indicates that its argument is an individual. Being an individual means that its properties and relationships with other things can change with time, and that it may not always physically exist. The relation **Exists-In(i, t)** indicates that individual **i** exists at, or during, time **t**. The import of this relationship is the creation of a *slice* to represent the properties of **i** at **t**. A slice of an object **B** at time **t** is denoted by **at(B, t)**. All predicates, functions, and relationships between objects can apply to slices to indicate their temporal extent, i.e., the span of time

they are true for.

Hayes' original treatment of histories did not address the interaction between existence and predication. What is the truth of a predicate applied to a slice when the individual is not believed to physically exist at the time corresponding to that slice? Allowing all predicates to be true of an individual when it doesn't physically exist has the problem that every fact **F** which depends on a predicate **P** must now also be explicitly justified by a statement of existence, such as

P(at(obj, t)) and Exists-in(obj, t) → F

rather than just

P(at(obj, t)) → F

To avoid this, we simply indicate that certain predicates which depend on physical existence imply that the individual does exist at that time, i.e.

P(at(obj, t)) → Exists-in(obj, t)

This allows the implications of the predication to be stated simply, while also providing a useful constraint for detecting inconsistencies. However, taxonomic constraints must be specified carefully. Consider the statement that an object is either rigid or elastic. If we simply assumed

For-All sl ∈ slice, Rigid(sl) or Elastic(sl)

we would be asserting the existence of the object at the time represented by that slice, since one of the alternatives must be true. These statements must always be placed in the scope of some implication which will guarantee existence, such as

For-All sl ∈ slice Physob(sl) → [Rigid(sl) or Elastic(sl)]

to avoid inappropriate presumptions of physical existence.

Situations describe a collection of objects being reasoned about at a particular time. A situation simply consists of a collection of slices corresponding to a set of objects existing at a particular time.[1]

An individual's existence is *quantity-conditioned* if inequality information is required to establish or rule out its existence. An example is Hayes' (1979) contained-liquid ontology. In this ontology a liquid exists in a container if there is a non-zero amount of it inside. It can be extended to a *contained-stuff* ontology that models solids and gasses as follows. Let the function **amount-of-in** map from states, substances, and containers to quantities, such that **A[amount-of-in(sub,st,c)]** is greater than zero exactly when there is some substance **sub** in state st in container **c**.[2] Let the function C-S denote an individual of a particular substance in a particular state inside a particular container. For instance, a coffee cup typically contains two individuals, denoted **C-S(coffee, liquid, cup)** and **C-S(air, gas, cup)**. The individual denoted by C-S exists exactly when the appropriate **amount-of-in** quantity is greater than zero. (See Forbus (1984b) for details.) Other kinds of material objects also seem to be captured by quantity-conditioned existence, including objects subject to sublimation, evaporation, or other changes in amount which do not cause "structural" changes. Examples include contained powders, heaps of sand, and ice cubes. A block of wood, however, provides a counter-example. Under certain conditions the block's existence can be modeled as quantity-conditioned, for instance when sanding or grinding down surfaces of it. But several ways of changing the block's existence cannot be so modeled -- consider sawing the block in half or bending it until it breaks. We will return to this issue later.

---

[1] Qualitative Process theory provides a way to determine what objects must be considered together for accurate prediction. Here we assume situations contain slices for all objects that exist at the time in question.

[2] In QP theory a quantity consists of an amount and a derivative, and the function A maps a quantity into its amount. Similarly, the function D maps a quantity into its derivative.

3. Modeling Changes of Existence In QP theory, processes are the ultimate source of all changes that happen in the physical world. Processes act by causing changes in continuous parameters of objects. A liquid flow, for instance, causes the amount of one liquid to increase and the amount of another to decrease. These changes in parameters will cause inequality relationships[3] to change. These in turn can lead to changes in the collection of active processes, as when the pressures in two containers equalize as a result of flow between them. They can also cause individuals whose existence is quantity-conditioned to appear and vanish. These changes are computed by a temporal inheritance procedure that determines what the world looks like after a change. This procedure solves the frame problem for simulation within the QP ontology.

A few remarks will make the procedure clearer. First, the statements which must be true for a process to act are divided into *quantity conditions* (which refer to inequalities and other relations defined within QP theory) and *preconditions* (all other statements a process depends on). We assume the facts stated in preconditions remain unchanged, although the procedure can be easily modified to track such changes (and the implemetation does so). Second, we assume that unless we know otherwise, individuals which exist remain in existence. Finally, the inequality relationships in the Quantity Spaces can be divided into two classes, those relationships in the current state which might change and those which cannot. Call the set of inequality relationships which might change Ω. Importantly, assuming that a particular change occurs implies that the relationships between the quantities it mentions change and that no other inequalities from Ω change.

Think of the facts which comprise a situation as consisting of a collection of assumptions and consequences of those assumptions. Finding the results of a change involves carefully modifying the assumptions. Two factors complicate this. First, the procedure which generates possible changes[4] is local, and thus sometimes hypothesizes changes which are not actually possible. The procedure described below filters out impossible changes. Second, the indirect consequences of the known changes will invalidate a subset of the previously held assumptions. For instance, an assumption about the level of water in a cup relative to some other height is moot if the water in the container no longer exists. The procedure correctly detects moot assumptions.

In what follows, "When consistent, assume **P**" means "if you don't already believe $\neg$ **P**, assume P. Otherwise, do nothing." The temporal inheritance procedure is:
(1) Assume that individuals whose existence is not quantity-conditioned remain in existence and that all preconditions remain the same.
(2) Assume the inequalities represented by the hypothesized change are true, and that all other relationships in Ω remain true.
(3) When consistent, assume quantity-conditioned individuals remain in existence.
(4) When consistent, assume that inequalities not in Ω hold.
If any required assumption leads to a contradiction, then assert that the proposed change is inconsistent. The algorithm is subtle, and is best understood by analyzing an example.

4. An Example Consider an open can partially filled with water sitting on a stove, such that the burner of the stove provides a heat path between them. Assume the water is initially below its boiling temperature and cooler than the stove. Figure 1 shows the possible behaviors (the envisionment) produced by GIZMO.[5] In the envisionment, **IS** indicates the set of quantity-conditioned individuals that exist in a situation. The set of active processes is indicated by **PS**. Possible changes are indicated by the prefix **QH**. The function **Ds** maps from a quantity to the sign of its derivative, which corresponds to the intuitive notion of direction of change (i.e., **-1** indicates decreasing, **0** indicates constant, and **1** indicates increasing). The Process Vocabulary used here consists of heat-flow and boiling, as described in Forbus (1984b).

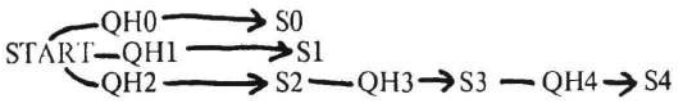
---

[3] In QP theory, numerical values are represented by collections of ordering relations called *Quantity Spaces*.

[4] Limit Analysis generates possible changes by looking at quantity space information and the signs of derivatives to determine all the possible ways the inequalities can change. While several domain-independent constraints, such as continuity, reduce the number of hypothesized changes domain-dependent information is sometimes required. The temporal inheritance algorithm provides a way to use such information.

[5] GIZMO implements the basic operations of Qualitative Process theory, including facilities for making predictions and interpreting measurements taken at a single instant. See Forbus (1984b) for details.

Figure 1 --- Predicted behaviors



Abbreviations:
T = temperature A-of = amount-of
TB = boiling temperature ST = stove
WC = C-S(water, liquid, can) HF1 = heat-flow(stove, WC, burner)
SC = C-S(water, gas, can) HF2 = heat-flow(stove, SC, burner)

Start: IS: {WC}, PS: {HF1}, Ds[T(WC)] = 1
S0: IS: {WC}, PS: {}, A[T(WC)] = A[T(ST)], all Ds values 0
S1: IS: {WC}, PS: {}, A[T(WC)] = A[T(ST)], A[T(WC)] = A[TB(WC)], all Ds values 0
S2: IS: {WC, SC}, PS: {HF1, HF2, Boiling}, Ds[T(WC)] = Ds[T(SC)] = 0 Ds[A-of(WC)] = -1, Ds[A-of(SC)] = 1
S3: IS: {SC}, PS: {HF2}, Ds[T(SC)] = 1
S4: IS: {SC}, PS: {}, all Ds values 0

QH0: A[T(WC)] < A[T(ST)] becomes =. QH1: QH0 and QH2 occur simultaneously.
QH2: A[T(WC)] < A[TB(WC)] becomes =. QH3: A[A-of(WC)] > zero becomes =.
QH4: A[T(SC)] < A[T(ST)] becomes =.

In the initial state START GIZMO deduces that heat flow occurs, since there is assumed to be a temperature difference between the stove and the water. It also deduces that boiling is not occurring, since we assumed no steam exists since **amount-of-in** for that combination of state and substance was zero. Either the heat flow will stop (if the temperature of the stove is less than or equal to the boiling temperature of the water, represented by changes **QH0** and **QH1**, respectively) or boiling will occur (if the temperature of the stove is greater than the boiling temperature, represented by change **QH2**). If boiling occurs (situation S2) then steam will come into existence. Ignoring flows out of the container, the next change is that the water will vanish (**QH3**), ending the boiling. The heat flow from the stove to the steam will continue, raising the steam's temperature until it reaches that of the stove (change **QH4**, resulting in the final state S4). We can see the impact of different aspects of the temporal inheritance method by seeing how this description would change if it were different. Failing to distinguish between changed and inherited quantity conditions (i.e., those in Ω and those in its complement) would rule out **QH0** since we would inherit the initial assumption of no steam. Inheriting beliefs concerning quantity-conditioned individuals before updating changed inequalities would preclude **QH3**, leaving us with water that was boiling away but never completely vanishing.

5. Discussion Quantity-conditioned existence provides a simple solution to the problem of existence for several important classes of material objects in Naive Physics (i.e., contained stuffs). It appears that quantity-conditioned existence can be extended to reason about all changes in existence caused by processes which affect the amount of something without affecting its gross structure. However, it cannot model all changes in existence; banging a rock with a hammer, for instance, results in the rock breaking into several new rocks. The reasons rocks break as they do

concern exactly where they are struck and the details of their microstructure. There is no simple description of this change by means of a small set of quantities because geometry is intimately involved. We should not be too discouraged, however, because it is not clear just how deep commonsense models of fracture really are. While we have rough ideas about the number and shape of pieces that result from breaking certain objects, we often cannot make very detailed predictions about exactly what pieces will result when an object breaks. Even traditional Materials Science cannot make such predictions in full detail for an arbitrary piece of material, so we shouldn't expect Naive Physics to do so.

The centrality of geometry in the open problems above suggests that another class of answers to the problem of existence lies in *qualitative kinematics*, the theory of places and their spatial relationships which, together with qualitative dynamics (e.g., Qualitative Process theory) may be viewed as providing the large-scale structure of Naive Physics. Configural information becomes even more important when considering more abstractly defined objects (such as a truss or a force balance), so it appears that a theory of qualitative kinematics might solve a large class of existence problems. The need for such a theory is growing clearer, and hopefully this paper will stimulate more work in this area.

6. **Acknowledgements** Brian Falkenhainer and Jeff Becker provided useful suggestions. This research is sponsored by the Information Sciences Division of the Office of Naval Research.

7. **References**

de Kleer, J. & Brown, J.S., "A qualitative physics based on confluences", *Artificial Intelligence* **24**, 1984

Forbus, K. "Qualitative reasoning about physical processes" Proceedings of IJCAI-7, Vancouver, B.C., 1981

Forbus, K. "Qualitative Process Theory" *Artificial Intelligence* **24**, 1984

Forbus, K. "Qualitative Process theory" TR-789, MIT Artificial Intelligence laboratory, July 1984

Hayes, P. "Naive Physics 1 -- Ontology for Liquids", Memo, Centre pour les Etudes Semantiques et Cognitives, Geneva, Switzerland. 1979

Kuipers, B. "Commonsense reasoning about causality: deriving behavior from structure" *Artificial Intelligence*, **24**, 1984

Simmons, R. "Representing and reasoning about change in geologic interpretation" MIT AI Laboratory technical report TR-749, December, 1983

Weld, D. "Switching between discrete and continuous process models to predict genetic activity", MIT AI Laboratory technical report TR-793, May, 1984

Williams, B. "Qualitative analysis of MOS circuits", *Artificial Intelligence*, **24**, 1984

CROSS-MAPPED ANALOGIES: PITTING SYSTEMATICITY AGAINST SPURIOUS SIMILARITY

Dedre Gentner & Cecile Toupin
Dept. of Psychology Dept. of Psychology
University of Ill. University of Cal.,Berkeley
Champaign, IL 61820 Berkeley, CA 94720

An analogy can be viewed as a device for conveying that two domains share significant relational structure even though they may not share surface similarity. The value of an analogy lies in its ability to give a causal or explanatory coherence to a new domain through the transference of a mutually-constraining set of relations. In Gentner's (1980, 1982, 1983) structure-mapping theory, this is called the systematicity principle. Intuitively, systematicity reflects people's tacit preference for coherence and deductive power in analogy. Syntactically, systematicity is realized as a preference for mapping relations that are governed by higher-order constraining relations that can themselves be mapped. The use of systematicity appears to be a central aspect of adult competence in comprehending analogies. In this research we investigate two questions concerning the nature of this competence: (1) the role of systematicity in the on-line mapping process and (2) the developmental course of the use of systematic knowledge in analogical mapping.

The first question is exactly how systematicity enters into the mapping process. Is it simply a passive desideratum which conveys the complexity, utility, or aptness of an analogy once it has been correctly interpreted ? In accordance with structure-mapping theory, we suggest that systematicity plays a decisive role in the mapping process itself. More specifically, our aim is to show that the presence of systematic structure in the base domain can help people keep the mapping process on track. Moreover, this affect should be most pronounced for difficult mappings. That is, the less transparent the object correspondences are between base and target, the more important will be the ability to take advantage of systematic structure. Without systematicity, spurious similarities between domains are likely to mislead in the mapping process.

To illustrate these principles, we offer a brief analysis of Rutherford's analogy between the solar system and the hydrogen atom. Consider the case where a person hears the Rutherford analogy for the first time. According to structure-mapping theory, the transference of knowledge from the base domain (e.g. the solar system) to the target domain (e.g., the hydrogen atom) involves a mapping process in which the objects from the base are placed in correspondence with objects in the target, e.g., sun --> nucleus, planet --> electron. Then predicates are mapped from the base to the target according to the following three mapping rules:

(1) The relations between objects in the base tend to be mapped across. For example, the lower-order relations:

i. MORE MASSIVE THAN (sun, planet) -->
MORE MASSIVE THAN (nucleus, electron)
ii. REVOLVE AROUND (planet, sun) -->
REVOLVE AROUND (electron, nucleus)

(2) The particular relations mapped are determined by systematicity, as defined by the existence of higher-order constraining relations which can themselves be mapped. For example,

ii. CAUSE [MORE MASSIVE THAN (sun, planet),
REVOLVE AROUND (planet, sun)] -->
CAUSE [MORE MASSIVE THAN (nucleus, electron),
REVOLVE AROUND (electron, nucleus)].

Note that the lower-order relation HOTTER THAN (sun, planet) is not part of the systematic structure and can be dropped.

3) Attributes of objects are dropped. For example:
iv. [YELLOW (sun)] --/-> [YELLOW (nucleus)]

By applying these principles the appropriate relational similarity between the solar system and atom is obtained, as

depicted in Figure 1.

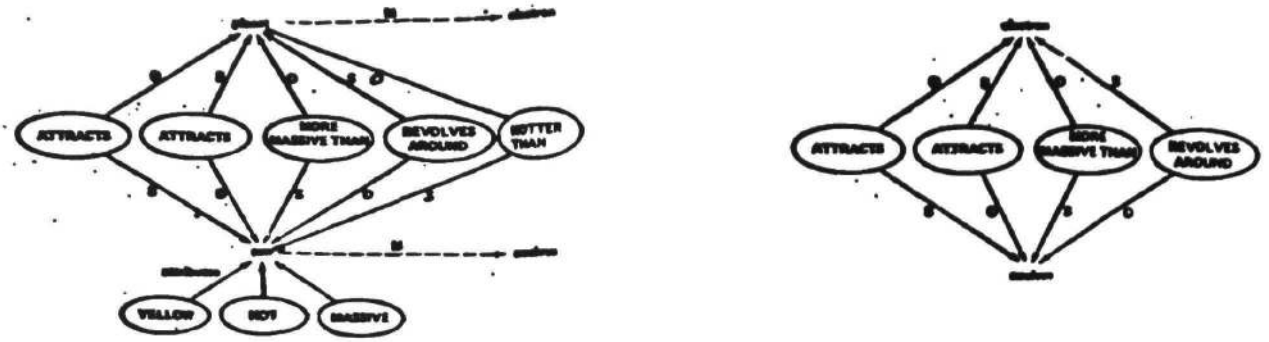
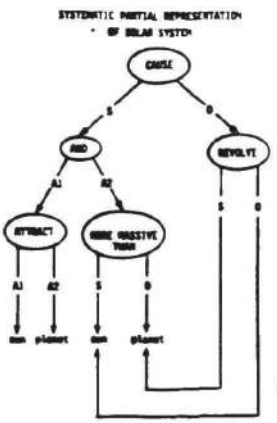


FIG1. Partial depiction of the analogy between solar system and hydrogen atom, showing a person's presumed initial knowledge of the solar system and the mapping of that knowledge to the atom.

By the structure-mapping account, two factors should enter into the success of the mapping process: (1) the transparency of the object correspondences and (2) systematicity. Higher-order relations help to guide the mapping of lower-order relations and provide a check on the correctness of the lower-order mappings. To illustrate how these two factors can interact in the mapping process, we now consider the case where the object correspondences are not wholly transparent, and a mapping error is made. Let us contrast two cases: (1) the systematic case: the learner's initial knowledge of the solar system includes the higher-order relation that there is a causal relation between (a) the fact that the planets revolve around the sun and (b) the fact that the sun is more massive than the planets: and (2) the nonsystematic case: the learner knows facts (a) and (b) but does not know the higher-order relation between them. Figure 2ab depicts a systematic and a nonsystematic representation of the s lar system.

- a. Systematic representation



- b Nonsystematic representation.

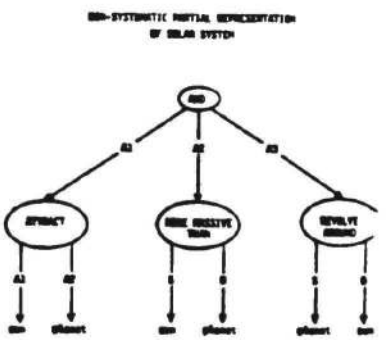


FIG2. More detailed depictions of a person's representations of the solar system

Now lets suppose both learner's are given the solar system/atom analogy, and both make the same error. They become momentarily confused and reverse the object correspondences while mapping the MORE MASSIVE THAN relation, resulting in MORE MASSIVE THAN (electron, nucleus). For the learner who has a systematic model of the base domain, the presense of a higher-order causal relation can be used to spot and correct the spurious mapping of the lower-order relation. In this example, when the predicate structure is transfered from base to target on the basis of the erroneous lower-order relation (proposition 1), as follows.

1. MORE MASSIVE THAN (electron, nucleus)
2. REVOLVE AROUND (electron, nucleus)
3. CAUSE [MORE MASSIVE THAN (electron, nucleus), REVOLVE AROUND (electron,nucleus)]

the learner is in a position to detect a relevant inconsistency. Namely that this chain violates the causal constraint that he knows holds true in the base domain i.e., CAUSE[MORE MASSIVE THAN (sun,planet),REVOLVE AROUND (planet,

sun)]. The inconsistency can be resolved by rechecking the object mappings, and reversing the order of objects in the MORE MASSIVE THAN predicate. Thus systematic knowledge of the base domain allows the learner to detect and repair an incorrect local mapping.

In the nonsystematic case, the learner has only two lower-order relations, one of which is erroneous:

1. MORE MASSIVE THAN (electron, nucleus)
2. REVOLVE AROUND (electron, nucleus)

There is nothing is this derived representation of the target domain that can alert him to an error in mapping these relations. Without systematic structure to map from the base domain, the learner simply has an unrestrained set of lower-order predicates. Note that the correct object correspondences - sun/nucleus and planet/electron - are not additionally supported by any salient attribute similarities. Thus unlike the learner who has systematic knowledge, he is unlikely to notice and repair the mapping error.

This line of reasoning leads us to two predictions concerning analogical mapping: 1) systematic knowledge of the base domain leads to more accurate analogical mapping and 2) the affect of systematicity will be strongest when the appropriate object correspondences are least transparent.

The Development of the Use of Systematicity

In this research we investigated the development of the use of systematicity in analogical mapping. We wished to discover when children are able to benefit from the presence of a set of mutually-constraining relations in carrying out an analogy. The method we used was designed to avoid some of the problems that commonly arise when assessing young children's analogical ability e.g., conflating developmental differences in children's ability to reason analogically, with differences in their command of vocabulary, knowledge about domains and pragmatic understanding of when nonliteral comparisons are permissable. The analogical domains in our study consisted of children's stories with simple plot structures. The analogical mapping step involved transferring the story plot from one set of characters to another. In performing this task the child was required only to act out the stories using toy dolls and animals. Thus the success of the child's mapping did not depend upon accessing any privledged information about the base domain, the child's verbal skills, or his/her pragmatic understanding of nonliteral comparisons.

In order to test our predictions concerning the use of systematic structure as a guide to analogical mapping two variables were manipulated: systematicity, and mapping difficulty. In varying the first factor, children were given either systematic or nonsytematic base scenarios to map. Higher-order relational information was embedded in the systematic stories in in two ways. In the systematic stories, the protoganist was attributed a story-relevant habit or relational trait(e.g.,"The chipmunk was very jealous"), whereas in the nonsystematic version a story-irrelevant trait was substituted (e.g."The chipmunk was very good-looking"). The second difference was that a final sentence expressing a moral and linking the protoganist's initial character trait to the story outcome was added only in the systematic story (e.g."The chipmunk realized that he shouldn't be so jealous, because it is better to have more friends). With the exception of these differences, the wording and hence the plot structure of the two scenario types was identical.

The degree of transparency of the object correspondences, or mapping difficulty, was varied using what we called a cross-mapping technique. With this manipulation, the correspondences between the original set of characters (the base set) and the test characters (the target set) could be either very easy with similar-looking characters playing the same roles (S/S); or moderately difficult, with different-looking characters playing the story roles (D); or extremely difficult, with characters that looked similar to the original characters but playing different roles than the ones they were assigned in the original story (S/D). An example will help to clarify the differences between these three mapping conditions. Suppose that

in the original story the hero was a chipmunk, the hero's friend was a robin and the villain was a horse. Then the roles in the three mapping conditions might be as follows:

	ORIGINAL	S/S	D	S/D
HERO	chipmunk	squirrel	elephant	zebra
FRIEND	robin	bluebird	shark	squirrel
VILLAIN	horse	zebra	cricket	bluebird

The prediction was that the children's accuracy in enacting the target story would be greatest for the S/S mapping condition and lowest in the S/D condition, where the natural object mappings had to be resisted. A further question was which of the two experimental variables - systematicity or object-mapping - would show up earlier developmentally. Our expectation was that effects of mapping condition would show up before effects of systematicity. A more interesting question, from our theoretical perspective, was whether systematicity would interact with mapping difficulty. For if the presense of systematic higher-order relations helps the child preserve the relational structure she is trying to map, then the more difficult the mapping the greater the potential effects of systematicity.

Method

Subjects. The subjects were 72 children, 36 four-to-six-year-olds, and 36 eight- to ten-year-olds, recruited from local preschools in Cambridge, Mass. Approximately equal numbers of males and females were included within each of the two experimental conditions (systematic and nonsystematic), and within each age group.

Materials. Nine short stories were constructed consisting of (1) an introductory section, which introduced two out of the three story characters and their relationship (2-3 sentences) (2) an event sequence, depicting a series of actions with an outcome (10 - 15 sentences) and (3) a moral (in systematic versions only). There were two versions of each story: systematic and nonsystematic. The two story types differed only in their introductory sections and in whether they contained a moral.

Story-telling Stimuli. Sixty-three toy dolls and animals were used to depict the characters. Of these, there were 27 pairs of animals that were independently judged by three judges to be 'similar-looking', and nine animals that were judged to be 'different looking' from one another and from any of the paired animals. A small number of props were additionally used to aid in the story-telling e.g., plastic food, and felt pieces to mark locations.

Systematicity Condition. Subjects in each age group were randomly assigned to either the Systematic or the Nonsystematic Condition so that there were equal numbers in both conditions. Subjects in each of these conditions received all nine stories of either the systematic or nonsystematic type, respectively.

Mapping Conditions. For each base story that the child heard, one of three sets of characters were inserted as the actors in the story. Relative to the 'target' set of characters, which were the same for all subjects, the characters depicted in the base story were either similar-looking characters in the same story roles (S/S); different-looking characters in the story roles (D); or similar-looking characters in different story roles (S/D). We decided to vary the characters in the original base story that the children heard , while keeping the target story to be retold by the subjects the same (instead of vice versa) in order to achieve strict comparability on the test/scoring phase.

Each child received three stories in each of the three mapping conditions, for a total of nine stories. The assignment of stories to mapping condition was counterbalanced across groups of children. The mapping conditions (S/S, D and S/D) were presented in six different orders, according to a Latin Square Design. There were two story-presentation orders; an ordering of all nine stories was presented in a forward or reverse sequence.

Procedure

The experimental procedure was the same for each story and was divided into two parts: the Story Phase and the Test Phase. Prior to beginning the experiment subjects received a practice session to acquaint the subjects with the nature of their task. Once the

child demonstrated the ability to perform the transfer task with a four-line story about two characters. without help, the experiment proceeded.

Story Phase. For each story, the experimenter began by introducing each of the three story characters by presenting their toy facsimiles and naming them (e.g., "Here is the moose.") With the characters in view, the base story was then read aloud. and the subjects were instructed to listen very carefully. Then props were introduced and the children were asked to act out the story on their own. verbalizing as much as possible. Any omissions and errors were corrected by the experimenter during this phase. The Test Phase began once the subject demonstrated the ability to act out the story correctly without help.

The Test Phase . The experimenter then asked the subject to act out the story again. but with three new characters. The three original story characters were removed from view and the new test characters were introduced and named. The experimenter then read aloud the introductory section of the story just read . substituting the names of the new 'target' characters for the names of the original story characters, thereby identifying the relationship between and character traits of two of the three characters. This introductory section was repeated if desired. Then the subject was told to tell the rest of the story. During the test phase, the experimenter did not provide the subject with any information regarding mapping assignments, omissions or errors.

The Story and Test Phases were carried out in the same way for each story. Children were given three stories in a test session. with a two-minute play task between stories. Each child participated in three test sessions. spaced at least a day apart.

Scoring. For each story, the sentences were grouped into six core propositions representing the major events and the outcome. These six did not include the moral in the systematic stories. Thus the same six propositions were scored for the systematic and nonsystematic conditions. In scoring, propositions were treated as wholes. A proposition was scored as correct if the child either verbally or nonverbally depicted the correct actor(s) and action(s) contained in the proposition. Two types of errors were scored: omissions and incorrect answers. A proposition was scored as an omission if the child both verbally omitted any actor or action contained in the proposition AND failed to convey the information through nonverbal actions. Similarly, a proposition was scored as incorrect if any actor(s) or action(s) contained in that proposition was incorrectly identified both verbally AND through nonverbal actions.

Results

The results are shown in Figure 3. All of our expectations were confirmed: (1) the object-mapping condition had strong effects on the transfer accuracy for both age groups. (2) systematicity benefited only the older group and (3) the benefits of systematicity were strongest in the most difficult mapping condition.

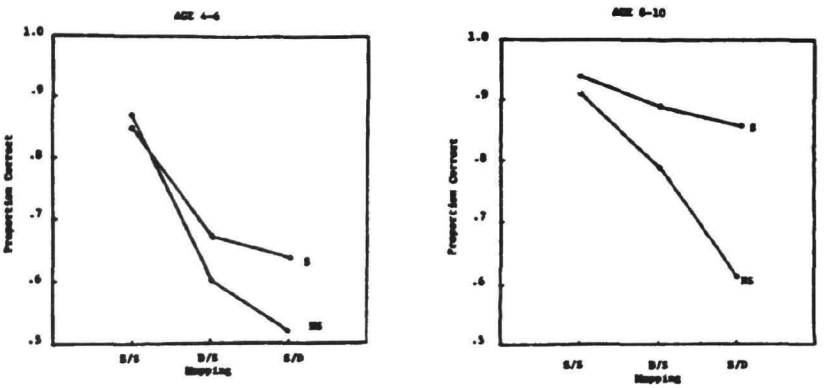


FIG 3. Results: Proportion of statements in the target stories correctly enacted under different kinds of mapping conditions given systematic versus nonsystematic versions of the original story, for four-to-six year-olds and eight-to-ten year-olds.

A 2x2x3 mixed-measures analysis of variance of Age (Between) x Systematicity (Between) x Mapping Condition (Within) showed main effects of Age [$F(1,68)=14.93$, $p < .01$]. Systematicity

[F(1,68)=6.28, p < .05], and Mapping Condition [F,(2,136)= 29.01, p < .000001]. There was also the predicted interaction between Systematicity and Mapping Condition [F(2,136)=3.89, p <.05]. Although both Mapping Condition and Systematicity show main effects, their developmental patterns differ. Planned comparisons confirmed that Mapping Condition had significant effects on both age groups. As predicted children performed best with the easy S/S mapping, intermediate with the D mapping, and worst with the misleading S/D mapping. In contrast, Systematicity showed significant effects only in the older group. For the older children, performance was significantly better on systematic stories [t(34)=2.48, p < .01]. This was not true for the younger children; they derived no significant advantage for systematic plot structure [t(34) = 1.08, NS]. Moreover, planned comparisons within the older group revealed that systematicity was significant only in the S/D condition. This confirmed our third prediction that systematicity should have its greatest effects on the most difficult mappings. This suggests that systematicity indeed plays a role in the mapping process: children (at least by the age of eight) can use higher-order constraints to help keep the lower-order predicates straight. We found informal support for this claim in the self-corrections that the older children occasionally made. A child would begin to make an error, acting out an event with the wrong character, and then stop herself with a remark like "No wait, it's the greedy one who got stuck in the well, because he ate too much." Here the child is using higher-order causal relations to check the correctness of a lower-order predicate.

Discussion

In this research we found effects of both systematicity and object-mapping difficulty on the accuracy of children's analogical mappings. These results have implications both for theories of analogical processing and for accounts of the development of analogy and metaphor. While the principle of systematicity has become increasingly prominent in computational approaches to analogy , there has been little evidence about how systematicity enters into the analogical process. Structure-mapping theory , and the results of this research , suggest that both systematicity and the naturalness of object correspondences contribute to the correctness of the mapping process. Two developmental questions were posed here: (1) whether there are developmental differences in the effects of difficult object correspondences in analogical mapping and (2) whether children develop in their ability to profit from systematic relational structure in dealing with difficult object mappings. One rather nice aspect of our methodology is that it allows an indirect measure of the child's ability to use systematicity. Research on the development of metaphor has shown repeatedly that children do not articulate their interpretation of metaphors in the same manner as adults. However, it is unclear what conclusions should be drawn from these results concerning children's ability to perform metaphorical and analogical transfer. In the present methodology, children simply acted out stories with a new set of characters. Thus, although they were not required to verbalize the relational structure they were carrying across, their ability to make the transfer was clear from the accuracy of their reenactment. Given that the child can act out the original story (which was in all cases true), we found (1) children of both ages were affected by the naturalness of the object mappings (2) systematicity benefited only the older children and (3) systematicity had its greatest effect when the object mappings were most difficult.

References

Gentner, D. (1980). The structure of analogical models in science (BBN Rrt No. 4451). Cambridge, MA: Bolt Beranek and Newman Inc.

Gentner, D. (1982). Are scientific analogies metaphors? In D. Miall (Ed.), Metaphor: Problems and perspectives. Brighton, England: Harvester Press Ltd.

Gentner, D. (1983). Structure-mapping. A theoretical framework for analogy. Cognitive Science, 7, 155-170.

Information, Uncertainty, and the Utility of Categories

Mark A. Gluck
Stanford University

James E. Corter
Columbia University

INTRODUCTION

Do categories which encode certain types of regularities or invariances in the world have a special psychological status? Many studies have shown that some categories or groupings of instances are easier than others to learn and recall as coherent concepts or generalizations. For example, within a hierarchically nested set of categories (such as a taxonomy of animals), there is some level of abstraction--called the "basic level"--that is most natural for people to use (Rosch, Mervis, Gray, Johnson, & Boyes-Braem, 1976). For example, in the hierarchy *animal-bird-robin*, *bird* is the basic level category. The preferrential status of basic level categories can be measured in a variety of ways. Basic level names are generally learned earlier by children (Rosch et al., 1976; Daehler, Lonardo, and Bukatko, 1979), and arise earlier in the development of languages (Berlin, Breedlove, & Raven, 1973). People tend to spontaneously name pictured objects at the basic level, and can name them faster at this level than at subordinate or superordinate levels (Rosch et al., 1976; Jolicoeur, Gluck, & Kosslyn, 1984).

Structural Explanations of the Optimality of Categories

Recent findings suggest that the superiority of basic level categories is due to *structural* properties of the categories, that is, to the distribution of features across instances and non-instances (Murphy & Smith, 1982; Hoffmann & Ziessler, 1983). Previous attempts to characterize these structural regularities have focused on measures of the degree to which basic level categories capture regularities or invariances in the world. For example, it has been suggested that basic level categories maximize the average *cue validity* of features for a category (e.g. the conditional probability of a category given a feature) or the average *category validity* (e.g. the conditional probability of features given a category), or some combination of the two, such as the product of the cue validity and category validity (Rosch & Mervis, 1975; Medin, 1983; Jones, 1983). These measures of category structure, however, are purely extensional and ignore the contexts and needs of people who are creating and using concepts. Recent evidence suggests that conceptual structures are highly unstable and vary greatly with context (Murphy & Medin, 1985; Barsalou, 1983). We suggest here a *context-sensitive* measure of the utility of categorizations. Our primary goal is to develop a model of the structural basis of basic levels effects; we present evidence that this measure does predict the occurance of basic levels in several experiments. We are motivated in this effort by the belief that the existence of basic levels reflects an underlying cognitive heuristic that people use for organizing information about the world.

For their guidance and comments, we are indebted to Gordon Bower, Misha Pavel, W. K. Estes, Ed Smith, Doug Medin, Joachim Hoffmann, and Greg Murphy. The assistance of Katie Albiston and Audrey Weinland is also gratefuly acknowledged. Please address correspondence to: Mark A. Gluck, Department of Psychology, Stanford University; Bldg. 420, Stanford, CA 94305.

The Informational Value of Categories

We suggest that the degree to which certain concepts are favored over others may be related to how useful these concepts are for encoding and communicating information about the properties of things in the world. In other words, the most useful categories are those that are, on the average, optimal for communicating information (hence reducing uncertainty) about the properties of instances. We will show how to formalize this idea in situations where the relevant attributes are well defined.

We consider two specific definitions of uncertainty and show the implications of each for Category Utility. First, we utilize the standard definition of uncertainty from information theory (Shannon & Weaver, 1949), and show what it implies about Category Utility. Second, we consider a hypothetical communication game in which one person attempts to transmit information about an item's attributes to another person. Within this game, we interpret uncertainty as an inability to predict attributes, and analyze how category membership information can be used to transmit information about the attributes of objects or events.

We will describe our theory of Category Utility within the context of a finite population of items, each of which is describable in terms of a set of multi-valued nominal attribute dimensions. Each attribute dimension (e.g. *eye color*) is assumed to have a set of possible values (e.g. *green, brown, blue*), one of which occurs in every instance. A category of instances can be described by specifying the *distributions* of attribute values for instances in the category. For example, a specific category of faces may have 40% green eyes, 50% brown eyes, and 10% blue eyes.

In information theory (Shannon & Weaver, 1949), the *uncertainty* of a set, F, of n messages (i.e. $F = f_1, f_2, \cdots, f_n$) is given by

$$U(F) = -\sum_{i=1}^{n} P(f_i)\log P(f_i).$$

We consider an attribute dimension to be a set of messages regarding the possible values of the attribute dimension. Consider also a partition, C, of a population of objects into two sets: those which are members of a category c and those which are not. Given information that an item is a member of category c, the uncertainty of the values of attribute dimension F will be:

$$U(F|c) = -\sum_{i=1}^{n} P(f_i|c)\log P(f_i|c),$$

where $P(f_i|c)$ is the conditional probability that a member of category c has value f_i on attribute dimension F. If instances of c occcur with probability $p(c)$ and instances of *not-c* occur with probability $(1-p(c))$, then the expected reduction in uncertainty when one is told the category or not-category information is:

Category Utility(C,F)

$$= \left[P(c)\sum_{i=1}^{n} P(f_i|c)\log P(f_i|c) + (1-P(c))\sum_{i=1}^{n} P(f_i|\ \textit{not}\ c)\log P(f_i|\ \textit{not}\ c) \right] - \sum_{i=1}^{n} P(f_i)\log P(f_i).$$

This measure of Category Utility is identical to the standard notion of the *information transmitted* between the message sets C and F.

In certain applications, we may be interested in defining the informational value of category c separately from that of *not–c*. The Category Utility of category c alone is given by:

$$\textbf{Category Utility(c,F)} = P(c)\left[-\sum P(f_i)\log P(f_i) \; - \; -\sum_{i=1}^{n} P(f_i|c)\log P(f_i|c)\right].$$

The Guessing-Game Measure of Category Utility

The information-theoretic measures of Category Utility given in the preceding section have close connections to expected performance in a feature prediction task. If we consider the expected score of someone guessing the values of each attribute dimension of an item, we can compare their expected score when they know nothing about the item to their expected score when they are told whether the items belong to c or *not–c*.

Assuming that the receiver adopts a ***probability-matching strategy*** (e.g. the receiver guesses value f_i with a probability equal to his expectation of the likelihood of f_i occuring given c or *not–c*) their expected increase in score given the category message can be shown to be given by:

$$\textbf{Category Utility(C,F)} = \left[P(c)\sum_{i=1}^{n} P(f_i|c)^2 + (1-P(c))\sum_{i=1}^{n} P(f_i|\ not\ c)^2\right] - \sum_{i=1}^{n} P(f_i)^2.$$

If the receiver is assumed to have no information about the *not–c* distribution, then the expected increase in score is

$$\textbf{Category Utility(c,F)} = P(c)\left[\sum_{i=1}^{n} P(f_i|c)^2 - \sum_{i=1}^{n} P(f_i)^2\right].$$

The information-theoretic and expected-score measures of Category Utility are closely related both in mathematical form and in terms of how they order categories as to relative goodness because $log(p)$ approximates to p for small numbers. Futhermore, assumptions about alternate guessing strategies have little effect on the predicted orderings of categorization utility as long as the strategy predicts that the receiver will do best when one attribute value is certain and will do worst when all attribute values are equally likely. In all our empirical applications to date, the most significant discrepancy between results of the information-theoretic and the feature-prediction versions are a few cases in which a tie in the goodness of two categories was broken by use of the other version.

APPLICATIONS TO CATEGORIZATION EXPERIMENTS

Hoffmann and Ziessler (1983) replicated the basic level phenomena using a series of three artificial category hierarchies. The hierarchies were differentiated by the degree to which exemplars of categories at different levels share common attribute values. Thus a different level was expected to be basic in each of the hierarchies.

Subjects were assigned to learn one of the three hierarchies. They were taught to associate each item with category names at three levels of generality (e.g. exemplar, intermediate, superordinate). Following this, subjects were presented with a picture of one of the items, paired with a concept name. They were asked to verify, as quickly as possible, whether or not the picture was an example of the named category. In a second task, they were asked to recall the correct name at a given level of abstraction. Reaction times for both the verification and naming studies indicated that the basic level was at the superordinate level for one hierachy, at the intermediate level for another, and at the exemplar level for the third hierarchy.

We evaluated the ability of our Category Utility index to predict, for each hierarchy, which level would be optimal or basic. The optimal level is operationally defined to be the level at which people are quickest to verify that an object is a member of the category. According to our theory, the average Category Utility of the categories at a given level should be highest for this optimal level. Utilizing a straightforward encoding of the drawings using three attribute dimensions (outline, edge, and bottom) with two, four, and four attribute values respectively, we calculated both the average Category Utility(c,F) of the individual categories and the average Category Utility(C,F) of the partitions induced by each individual category. For comparison, we also calculated the average cue validity, category validity, and the product of these two (G. Jones' (1983) "collocation" measure).

In these studies, cue validity and the collocation measure invariably identified the highest level as best. Category validity failed to distinguish between any of the levels. In summary, these measures were insensitive to the manipulation of attributes across the three hierarchies; each failed to predict the basic level in at least two out of three studies. The average Category Utility(c,F) index correctly predicted the ordering of reaction times for the three levels in *each* of the three hierarchies, with the exception of giving equal ratings to the basic and intermediate levels in the first hierarchy. The average Category Utility(C,F) index correctly predicted the ordering of reaction times in all three hierarchies.

In addition, in the analyses of a similar experiment by Murphy & Smith (1982), Category Utility was also the only measure which successfully predicted the basic level.

DISCUSSION

The results from these experiments indicate that Category Utility is able to predict the psychologically preferred level of categorization in these verification and naming experiments. None of the alternative measures did nearly as well. An additional advantage to the measure is that it is context sensitive: Category Utility is computed as an expected *decrease* in uncertainty given some context population. Thus, this affords a way of measuring how the utility of a category or generalization can change depending on the context in which it is analyzed. This is particularly important from a psychological standpoint because of evidence indicating that the usefulness of categories and concepts is highly context dependent (Murphy & Medin, 1985; Barsalou, 1982, 1983).

References

Barsalou, L. W. (1982). Context-independent and context-dependent information in concepts. *Memory & Cognition, 10,* 82-93.

Barsalou, L. W. (1983). Ad hoc categories. *Memory & Cognition, 11,* 211-227.

Berlin, B., Breedlove, D. E., & Raven, P. H. (1973). General principles of classification and nomenclature in folk biology. *American Anthropology, 75,* 214-242.

Daehler, M. W., Lonardo, R., & Bukatko, D. (1979). Matching and equivalence judgments in very young children. *Child Development, 50,* 170-179.

Hoffmann, J. & Ziessler, C. (1983). Objectidentifikation in kunstlichen Begriffshierarchien. *Zeitschrift fur Psychologie, 194,* 135-167.

Jolicoeur, P., Gluck, M., & Kosslyn, S. (1984). Pictures and names: Making the connection. *Cognitive Psychology, 16,* 243-275.

Jones, G. V. (1983). Identifying basic categories. *Psychological Bulletin, 92,* 174-177.

Medin, D. (1983). Structural principles of categorization. In B. Shepp & T. Tighe (Eds.), *Interaction: Perception, Development and Cognition.* Hillsdale: Erlbaum.

Murphy, G. (1982). Cue validity and levels of categorization. *Psychological Bulletin, 91,* 174-77.

Murphy, G. & Medin, D. (1985). The role of theories in conceptual coherence. *Manuscript.*

Murphy, G. L. & Smith, E. E. (1982). Basic level superiority in picture categorization. *Journal of Verbal Learning and Verbal Behavior, 21,* 1-20.

Rosch, E. & Mervis, C. (1975). Family resemblances: Studies in the internal structure of categories. *Cognitive Psychology, 7,* 573-603.

Rosch, E., Mervis, C., Gray, W., Johnson, D., & Boyes-Braem, P. (1976). Basic objects in natural categories. *Cognitive Psychology, 8,* 382-439.

Shannon, C. & Weaver, W. (1949). *The Mathematical Theory of Communication.* Chicago: University of Illinois Press.

A MODEL OF QUESTION ANSWERING

Arthur C. Graesser
Memphis State University

David Koizumi
California State University, Fullerton

George Vamos
University of Southern California

C. Scott Elofson
California State University, Fullerton

Abstract

This short report summarizes a new model of question answering that we have developed and tested. The model specifies how humans answer many different kinds of questions (including why, how, when, where, enablement, consequence, and significance questions) after comprehending narrative passages. For example, if a narrative passage contained the episode the dragon kidnapped the maidens, the why-question for this episode would be why did the dragon kidnap the maidens? and possible answers would be because the dragon wanted to eat the maidens and because the dragon was lonely. According to the model, the major information sources for answers to questions include the passage structure and the generic knowledge structures that are associated with the content words in the query (i.e., DRAGON, MAIDEN, KIDNAPPING). After these knowledge structures are activated in working memory, there are convergence mechanisms which narrow down the node space to a set of relevant answers to a given question. The convergence mechanisms involve four major components. First, there is an arc search procedure, associated with each question category, which specifies what categories and paths of arcs are sampled when knowledge structures are tapped for answers. Second, there are a set of heuristics for establishing priorities among knowledge structures. Third, there is an intersecting node identifier which segregates those nodes in a given knowledge structure which overlap (match) a node in at least one other knowledge structure in working memory. Fourth, there is a constraint propagation component which prunes out erroneous nodes during the evaluation of (a) the intersecting nodes and (b) the nodes that radiate from intersecting nodes. The model has been tested by simulating question answering protocols collected from human subjects.

This paper summarizes a model which accounts for the answers that adults give to questions after they comprehend simple narrative passages. For example, suppose that the comprehender reads a passage about a dragon who kidnaps some maidens in a forest and some heroes who rescue the maidens from the dragon. (Of course, the setting and plot would be a bit more embellished.) After the passage is read, the comprehender may be asked a number of questions about the episodes in the passage. The following actions could be asked about the episode the dragon kidnapped the maidens):

Why did the dragon kidnap the maidens?
How did the dragon kidnap the maidens?
When did the dragon kidnap the maidens?
Where did the dragon kidnap the maidens?
What enabled the dragon to kidnap the maidens?
What are the consequences of the dragon kidnapping the maidens?
What is the significance of the dragon kidnapping the maidens?

The proposed model explains how adults answer these seven categories of questions (why, how, when, where, enablement, consequence, and significance). The model specifies the knowledge structures that are tapped for answers and the process of converging on the relevant answers to specific questions.

During the last decade, researchers in artificial intelligence have vigorously investigated the process of question answering in the context of narrative text and mundane world knowledge (Bobrow & Winograd, 1977; Dyer, 1983; Lehnert, 1978; Schank & Abelson, 1977). However, research in cognitive psychology has only recently begun to emerge (see Graesser & Black, 1985). Perhaps the most comprehensive psychological model of question answering was introduced and developed by Graesser (Graesser, 1981; Graesser & Clark, 1985; Graesser & Murachver, 1985; Graesser, Robertson, & Anderson, 1981). The present model constitutes a major extension and modification of Graesser's earlier work on question answering.

A Rich Data Base for Testing Models of Question Answering

Graesser has collected an extensive set of question answering protocols and other relevant data from adult subjects (Graesser & Clark, 1985). These data provide a testbed for evaluating alternative models of question answering. First, Graesser has has mapped out passage structures for four short narrative passages which vary in cohesiveness. The passage structures include explicit statements and knowledge-based inferences that are needed for establishing conceptual connectivity between explicit statements. The bridging inferences were extracted from subjects empirically (for details about methods, see Graesser and Clark, 1985). Second, Graesser mapped out generic knowledge structures (GKSs) associated with explicit content words in the four narrative passages and with higher-level GKSs which are triggered by patterns of information. Again, the content of each GKS was extracted empirically from human subjects (see Graesser and Clark, 1985 for details). Third, Graesser collected question answering protocols from adult subjects after they comprehended the four narrative passages. Graesser queried each statement in the passages with seven questions (why, how, when, where, enablement, consequence, and significance). Thus, there was an answer distribution for each specific question. These data (and other data which we will not discuss here) provided a rich data base for discovering question answering mechanisms and for testing models of question answering. The model proposed in this paper is grounded in a rich foundation of data collected from adults.

The knowledge embodied in each passage structure and GKS was translated into a conceptual graph structure. A conceptual graph structure is a set of categorized statement nodes which are interrelated by a network of categorized, directed arcs. In the representational

system adopted by Graesser and Clark (1985), a statement node is a proposition-like description that is assigned to one of five categories: Event, State, Goal, Action, and Style specification. There are nine arc categories: Reason, Outcome, Initiate, Manner, Consequence, Implies, Property, Set Membership, and Referential Pointer. It is beyond the scope of this report, however, to define these node and arc categories. The important point is that the empirically extracted knowledge was structured according to a representational system developed by Graesser and Clark (1985).

When considering the four narrative passages, an average passage contained 125 statement nodes, with 25 explicit nodes and 100 inference nodes. A typical passage activated 35 GKSs. Approximately two-thirds of the GKSs associated with a passage corresponded to a content word in the text. For example, the passage statement the dragon kidnapped the maidens would activate three GKSs: DRAGON, KIDNAP, and MAIDEN. The other third of the GKSs were pattern-activated GKSs, that is, they were triggered by patterns of information rather than explicit words in the text (e.g., FAIRYTALE, CONFLICT, FEAR). An average GKS contained 166 statement nodes in its conceptual graph structure. Once again, the nodes in these structures had been extracted empirically from human subjects.

One Problem with Graesser's Previous Models of Question Answering

Graesser's previous models (Graesser & Murachver, 1985) went a long way in explaining the answers that humans produce when they answer the seven categories of questions. The models identified the major information sources for the answers to the questions. Specifically, 75-80% of the answers to questions tap nodes in either (a) the passage structure or (b) the GKSs corresponding to the content words in the query. For example, suppose that the question is why did the dragon kidnap the maidens? in the context of narrative passage N. The answers to this question would come from four knowledge structures: passage structure N, the GKS for DRAGON, the GKS for KIDNAP, and the GKS for MAIDEN. Graesser's previous models also identified the arc search procedures associated with the different question categories. An arc search procedure specifies the categories of arcs that are sampled when searching through knowledge structures for answers. For example, answers to why-Action questions sample forward Reason arcs (leading to superordinate goals) and backward Initiate arcs (corresponding to events, states, and actions that initiate goals in animate agents). The arc search procedures of other question categories are different. Graesser reported that 98% of the obtained answers to questions would be generated if the theoretical arc search procedures were applied to the relevant knowledge structures (given that the answer node can be found in these knowledge structures).

There was one major problem with Graesser's previous models of question answering. The models generated too many theoretical answers to specific questions when theoretical question answering procedures were applied to the relevant knowledge structures. For example, the model would generate 50 answers to a question (out of thousands of nodes) whereas adults would generate only 10. Thus, there was a convergence problem in the models--the models did not satisfactorily converge on a small set of answers that are relevant to the question. Our new model of question answering does a better job capturing convergence mechanisms.

The New Model of Question Answering

The new model of question answering incorporates many of the assumptions and components of Graesser's earlier models. However, the new model addresses the problem of convergence more satisfactorily. Listed below are the six major components or properties of the new model.

Working memory. During the process of answering a question, knowledge structures enter a limited capacity working memory and interact with each other. Sometimes one structure directly communicates with another knowledge structure according to a private line or party line. Alternatively, one structure may post constraints that are broadcasted to all other knowledge structures in working memory.

Activation of knowledge structures in working memory. During question answering, a given knowledge structure is activated in working memory through pattern recognition processes. Graesser and Clark (1985) specified what knowledge structures occupy working memory during passage comprehension and during question answering. We adopted these assumptions in the present model. Specifically, when a question is asked (e.g., why did the dragon kidnap the maidens?) the knowledge structures in working memory include (a) the passage structure (actually, a proximate substructure from the passage structure), (b) the GKSs corresponding to the content words in the question (i.e., DRAGON, KIDNAP, and MAIDEN), and (c) occasionally some pattern-activated GKSs (e.g., EVILNESS, FAIRYTALE).

Arc search procedure. For each question category, there is an arc search procedure which specifies the legal paths of arcs that may be pursued when the procedure is applied to a knowledge structure. We adopted the arc search procedures specified in Graesser and Clark (1985) because they proved to be satisfactory in accounting for the question answering protocols.

Priorities among knowledge structures in working memory. We adopted a set of heuristics for establishing priorities among knowledge structures in working memory. These heuristics were needed for resolving conflicts when knowledge structures interact in working memory. For example, one heuristic is that the passage structure has priority over GKSs in working memory. Another heuristic is that the GKS associated with the verb has priority over the GKSs associated with nouns in the query. It should be noted that the verbs convey action/event information that is central to the plot in narrative passages.

Intersecting nodes. Nodes that intersect (i.e., match, overlap) between/among knowledge structures have a special status in the question answering process. The analyses revealed that the intersecting nodes had a much higher likelihood of being produced as answers to questions than did the nonintersecting nodes. Moreover, nodes that were proximate to intersecting nodes were produced as answers with a higher likelihood than were distant nodes. When we inspected the knowledge structures in working memory, we found that the likelihood of producing an answer decreased exponentially as a function of the distance from the nearest intersecting node.

Constraint propagation. When knowledge structure X has priority over knowledge structure Y in working memory, constraints from X are imposed on knowledge structure Y and thereby prune out nodes in Y from consideration (as answers to the question). We identified many of the criteria for pruning out nodes in the constraint propagation mechanism. One criterion is direct contradiction. A node in structure X directly contradicts a node structure Y, so the node in Y is pruned from consideration (as well as all nodes that radiate from the pruned node, away from the nearest intersecting node). Other criteria include time frame incompatibilities, argument frame incompatibilties, and conflicts in resources when agents try to execute plans (see Wilensky, 1983).

It is important to emphasize that the assumptions of our model of question answering were discovered and/or tested by examining the rich, qualitative database that was collected in Graesser and Clark's (1985) earlier study. Moreover, parts of the model have been simulated on computer in LISP and in PROLOG.

References

Bobrow, D.G. & Winograd, T. (1977). An overview of KRL, a knowledge representation language. Cognitive Science, 1, 3-46.

Dyer, M.G. (1983). In-depth understanding: A computer model of integrated processing for narrative comprehension. Cambridge, Mass: M.I.T. Press.

Graesser, A.C. (1981). Prose comprehension beyond the word. New York: Springer-Verlag.

Graesser, A.C. & Black, J.B. (1985). The psychology of questions. Hillsdale, N.J.: Erlbaum.

Graesser, A.C. & Clark, L.F. (1985). Structures and procedures of implicit knowledge. Norwood, N.J.: Ablex.

Graesser, A.C. & Murachver, T. (1985). Symbolic procedures of question answering. In A.C. Graesser and J.B. Black (Eds.), The psychology of questions. Hillsdale, N.J.: Erlbaum.

Graesser, A.C., Robertson, S.P., & Anderson, P.A. (1981). Incorporating inferences in narrative representations: A study of how and why. Cognitive Psychology, 13, 1-26.

Lehnert, W.G. (1978). The process of question answering. Hillsdale, N.J.: Erlbaum.

Schank, R.C. & Abelson, R. (1977). Scripts, plans, goals, and understanding. Hillsdale, N.J.: Erlbaum.

Wilensky, R. (1983). Planning and understanding. Cambridge, Mass.: Addison-Wesley.

The Time Course of Anaphora Resolution

Raymonde Guindon

Microelectronics and Computer Technology Corporation

Abstract

Anaphors, such as definite noun phrases and pronouns, are important contributors to discourse coherence. Anaphora resolution is the process of determining the referent of an anaphor in a discourse or dialogue. Models of discourse and sentence comprehension have made different claims about the temporal relationship between the occurence of the syntactic and semantic analyses of the sentence and the process of anaphora resolution. The **end-of-sentence** hypothesis holds that anaphora resolution occurs at the end of the sentence, after the syntactic and semantic analyses are completed. The **immediacy assumption** holds that anaphora resolution occurs as soon as an anaphor is encountered and is completed as much as possible before further words are processed. The **cognitive lag** hypothesis assumes that anaphora resolution starts when the anaphor is encountered but is completed while processing further words in the sentence. A study is described that traces the activation of a referent by its anaphor over a complete sentence. It demonstrates that anaphora resolution does not await the complete syntactic and semantic interpretations of the sentence. An anaphor starts activating its referent as soon as the anaphor is encountered and the referent stays activated until the end of the sentence. This result supports a particular version of the immediacy assumption. This is also interpreted in terms of a limited cache that stores the items currently in focus and that is updated at sentence or clause boundaries.

Introduction

Temporal Characteristics of Anaphora Resolution

An important aspect of discourse comprehension models, whether in cognitive psychology or in computational linguistics, is the assumed temporal characteristics of anaphora resolution. Two specific questions to be answered are: 1) When does anaphora resolution occur during the comprehension of a sentence? 2) When does it occur in relation to the syntactic and semantic analyses? This study provides preliminary evidence on these questions.

There are three main hypotheses regarding the temporal relationship between the syntactic and semantic analyses of the sentence and anaphora resolution. The **end-of-sentence** hypothesis holds that anaphora resolution occurs at the end of the sentence, after the syntactic and semantic analyses are completed (Bever & Hurtig, 1975). The **immediacy assumption** holds that anaphora resolution occurs as soon as an anaphor is encountered and is completed as much as possible before further words are processed (Just & Carpenter, 1980). The **cognitive lag** hypothesis assumes that anaphora resolution starts when the anaphor is encountered but is completed while processing further words in the sentence (Ehrlich & Rayner, 1983). A similar issue has arisen in designing natural language interfaces, with a tendency to make syntax and semantics work "in tandem" (e.g. Cascaded ATN, Psi-Klone (Bobrow & Webber, 1980)). In most natural language understanding systems, though, anaphora resolution occurs after the syntactic and semantic analyses of the sentence are completed (e.g. see Sidner, 1984).

1- This research was performed at the University of Colorado, Boulder, as part of the author's doctoral dissertation. Walter Kintsch, Peter Polson, Alice Healy, Richard Olson, and Andrjez Ehrenfeucht are kindly thanked for their help with this research.

Short-term memory and cache management

The design of this study and the interpretation of its results are based on a model of discourse comprehension by Kintsch and van Dijk (1978). A sketch of an enhanced version of this model is presented.

Memories The memory of the system is divided into three components: a small, very fast memory called short-term memory (STM); a relatively larger main or operating memory (OM); a vast store of general world knowledge called long-term memory (LTM).

The total **STM** contains only 7±2 chunks at any one time (Simon, 1974; Miller, 1956). STM is itself divided into two other memories to which these chunks are allocated. First, the **buffer** is used to store the incoming clause or sentence before further processing. Second, the **cache** is used to hold over, from one sentence or clause to the next, the information necessary to provide global and local coherence. It contains a subset T of the previous most topical text items and a subset R of the most recent text items. Retrieval times from short-term memory are very fast. Items stored in the cache are in focus (Guindon, 1985).

The **operating memory** is conceptually that subset of the world knowledge in long-term memory which is deemed relevant to the processing of the current part of the text. It also contains the growing structure corresponding to the text read so far. It contains the less topical and less recent information from the text. Operating memory and long-term memory can contain a very large number of entities. However, retrieval times are much longer than for short-term memory. Items not in focus are in operating memory (Guindon, 1985).

Cache Management Anaphora resolution proceeds in a number of partially concurrent steps. As a new sentence is read into the buffer and an anaphor is encountered, its referent is searched in the cache. If the referent is not found in the cache, operating memory is searched. If the referent is found in operating memory, it is reinstated in the cache and is now in focus. The cache management strategy is applied at each sentence or clause boundaries. As a consequence, once an item is in focus it will stay activated until at least the end of the sentence or clause.

Overview of the On-line Activation Technique and the Study

A technique called **on-line activation** was developed to trace the activation of a referent by an anaphor during a whole sentence. The technique is *on-line* because the activation is measured at various short intervals. It is an *activation* technique because the recognition latency of an old test word is measured; the anaphor acting as a prime and the test word being the referent. This allows a **direct** measure of the activation of a **specific** referent by an anaphor.

Using the on-line activation technique, subjects read passages in which an anaphor referred to an antecedent which was not in focus, that is, the antecedent was not in the cache. The antecedent was removed from focus by introducing a topic shift. An example text is presented in Table 1. Only one of the sentences 5a and 5b was presented in a text during the study. The passages were presented using rapid serial visual presentation (RSVP), one word at a time in the center of a CRT. The presentation time was 300 msec per word.

In addition to reading each text, the participants were also given the task to recognize whether some specially marked words, presented surreptitiously within the text, had appeared before in the text or not. These specially marked words are called test words. Some of the test words were old, some were new. In each text, one of these test words was the referent of the anaphor. At some point before or after the anaphor was presented on the CRT, its referent was presented for recognition and recognition times and errors were collected. The delay between the onset of the anaphor and the onset of the test word is called the stimulus onset asynchrony (SOA). The selected SOAs spanned the whole sentence, as can be seen in Table 1. The anaphor is acting as a prime, which should activate or reinstate the referent, that is, bring the referent in the cache with its fast retrieval time. The recognition time for the referent test word indicates whether the referent is in the cache (i.e.

fast) or in operating memory (i.e. slow). In addition, there were two types of primes, as shown in sentences 5a and 5b in Table 1. The prime could be either semantically related and referential (S+R+) as in 5a, or semantically related and non-referential (S+R-) as in 5b. The S+R- condition acts as a control condition where reinstatement of the referent is not expected. The S+R- condition is not referential, because of the use of an indefinite article (e.g. "a") and an adjective incompatible with the referent, but semantic priming is the same as in S+R+

Table 1

Example of an experimental text

Antecendent/test word: thermometer Anaphor: instrument

1-- The assistant was preparing solutions for a chemistry experiment.
2-- He measured the temperature of a solution using a *thermometer*.
3-- The experiment would take at least four hours.
4-- There would then be a ten hour wait for the reaction to finish.
5a- The thin *instrument* was not giving the expected reading.
5b- A broken *instrument* was not giving the expected reading.
SOA locations ^ ^ ^ ^ ^ ^ ^ ^ ^

A schema of the procedure is shown in Table 2. The words surrounded by stars are the test words.

Table 2

Schema of the procedure for three SOAs

SOAs Time	Before anaphor	350 msec after anaphor	1250 msec after anaphor
Ti	The	The	The
Ti+1	thin	thin	thin
Ti+2	*thermometer*	instrument	instrument
Ti+3	instrument	*thermometer*	was
Ti+4	was	was	not
Ti+5	not	not	giving
Ti+6	giving	giving	*thermometer*

Given that recognition latencies are much shorter for items in the cache than in the operating memory and given that an anaphor reinstates a referent not in focus by bringing the referent in the cache from operating memory, the following pattern of recognition latencies can be derived. A pattern supporting the end-of-sentence hypothesis would be a decrease of recognition latencies for the referent observed only at the end of the sentence.A pattern supporting the immediacy assumption would be a rapid decrease of recognition latencies for the referent as soon as the anaphor is encountered and reaching its peak before the next words are processed. A pattern supporting the cognitive lag position would be a decrease in recognition latencies starting soon after the anaphor is encountered and reaching its peak a few words later.

Method

Participants The participants were 36 undergraduate psychology students.

Materials There were 60 texts: 36 experimental, 18 filler, and 6 practice. The experimental texts contained as a referent an instance of a class (e.g. thermometer) to be used later as a test word, and as an anaphor the class name (e.g. instrument). An example of the material was

presented in Table 1. There were two priming conditions, S+R+ as in sentence 5a, and S+R- as in sentence 5b. During the presentation of each text, two or three test words were presented, one experimental and one or two fillers. The filler words were presented at semi-random locations in the text. There was an equal number of old and new test words in the entire experiment.

Procedure The experiment was computer controlled using real-time routines on the VAX/VMS 11/780 of the Computer Laboratory for Instruction in Psychological Research at the University of Colorado. The texts were presented using RSVP, with each word presented in the center of the screen for 300 msec. Two or three test words, surrounded by stars, were presented at different locations within each text. The participants were asked to recognize whether the test words were old or new, as fast as possible and without making mistakes. After each text, the participants were given feedback on the number of mistakes they had made.

Design There were 36 experimental texts and 18 experimental conditions, all within-subject. There were two priming conditions: 1) semantically related and referential (S+R+); 2) semantically related and non-referential (S+R-). The non-referentiality of the S+R- condition was achieved by the inclusion of an indefinite article (e.g. "a") followed by an adjective inconsistent with the referent. There were nine SOAs: 1) following the article at the beginning of the sentence (e.g. "the" or "a"); 2) following the adjective or, in other words, immediately before the prime; 3) 350 msec after the onset of the prime; 4) 650 msec after; 5) 1250 msec after; 6) following the third word from the end of the sentence; 7) following the second word from the end of the sentence; 8) following the last word of the sentence; 9) 600 msec after the last word of the sentence. The last four SOAs were used to test whether special integrative processes occur at the end of the sentence. The last SOA, 600 msec after the offset of the last word, was included to test whether integrative processes lag somewhat after processing the last word of the sentence (see Aaranson & Scarborough, 1976).

Results & Discussion

Separate analyses treating subjects and texts as random variables were performed. The recognition latencies for each priming conditions at each SOAs are presented in Figure 1.

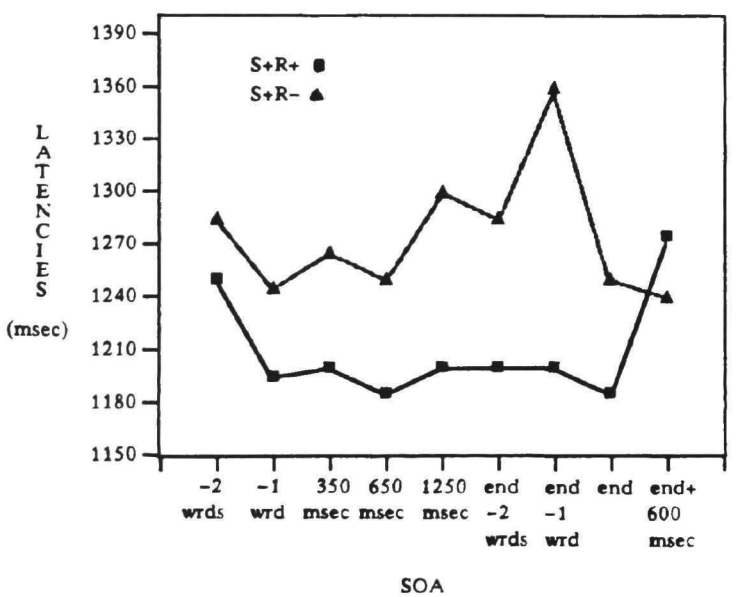


Figure 1. Recognition latencies for referent test words at each SOAs in each priming conditions.

As predicted, there was a strong effect of priming, with faster recognition latencies in the S+R+ condition than in the S+R- condition, $F(1,35)= 10.6$, $p= .003$, $MSe= 719102$ by subjects and

F(1,35)= 10.7, p= .002, MSe= 930316 by items. The referent was not reinstated in the cache from operating memory in the non-referential condition, S+R-. This indicates that in the S+R- condition the preceding syntactic cue (i.e. an indefinite article) and the preceding semantic cue (i.e. an adjective incompatible with the referent) were sufficient to quickly rule out activation of the referent. This is consistent with an on-line interactive model of discourse comprehension where syntatic, semantic, and pragmatic analyses run concurrently and synergistically.

While the interaction was not significant, the curves indicate that, in the S+R+ condition, the referent was activated early by the anaphor and stayed activated until the end of the sentence. The pattern of recognition latencies clearly does not support the end-of-sentence hypothesis and seems to support a particular version of the immediacy assumption: the referent is activated early after the anaphor is encountered and stays activated until the end of the sentence. Initial integrative processes at the discourse level - identification and search of the referent of an anaphor - are immediate. They do not await the completion of the syntactic and semantic analyses of the sentence. Again, the results support an on-line interactive system where syntax, semantics, and pragmatics cooperate concurrently. Moreover, these results support the hypothesis that cache management is applied at clause or sentence boundaries. Consequently, once an item is reinstated in the cache and is in focus, it stays activated until the end of the sentence or clause or until the next topic shift.

References

Aaranson, D. & Scarborough, H.S. Performance theories for sentence coding: Some quantitative evidence. *Journal of Experimental Psychology: Human Perception and Performance*, 1976, *2*, 56 - 70.

Bever, T.G. & Hurtig, R.R. Detection of a non-linguistic stimulus is poorest at the end of a clause. *Journal of Psycholinguistic Research*, 1975, *4*, 1 - 7.

Bobrow, D.G. & Webber, B.L. *Knowledge representation for syntactic/semantic processing.* Proceedings of AAAI, Stanford University, 1980.

Ehrlich, K.E. & Rayner, K. Pronoun assignment and semantic integration during reading: Eye movements and the immediacy of processing. *Journal of Verbal Learning and Verbal Behavior*, 1983, *22*, 75 - 87.

Guindon, R. *Anaphora resolution: Short-term memory and focusing.* Proceedings of the Association for Computational Linguistics Meeting, Chicago, 1985.

Just, M.A. & Carpenter, P.A. A theory of reading: From eye fixations to comprehension. *Psychological Review*, 1980, *87*, 329 - 354.

Kintsch, W. & van Dijk, T.A. Toward a model of text comprehension and production. *Psychological Review*, 1978, *85*, 363 - 394.

Miller, G.A. The magical number seven, plus or minus two: Some limits on our capacity for processing information. *Psychological Review*, 1956, *63*, 81 - 97.

Sidner, C. Focusing in the comprehension of definite anaphora. In M. Brady & R. C. Berwick (Eds.), *Computational Models of Discourse*, Cambridge: MIT Press, 1984.

Simon, H.A. How big is a chunk? *Science*, 1974, *183*, 482 - 488.

Using a Computational Model of Language Acquisition to Address Questions in Linguistic Inquiry

Jane Hill
Department of Computer Science
Smith College, Northampton, MA 01063

In this paper we are presenting a case study to illustrate the way in which our computational model of English language acquisition and language performance in very young children may be used to address questions in linguistic inquiry. Ours is a model which depends on the interaction of language and cognitive domains, and which seeks commonalities between the processes employed in language perception and sensory perception in other modalities, as well as between language production and response to stimuli in other modalities.

1. Computational Models and the Innatist-Empiricist Controversy

Since Noam Chomsky published Syntactic Structures in 1957 the dialogue between the innatists and the empiricists has fundamentally influenced the course of research in language acquisition. The debate between Chomsky and Piaget (Piatelli-Palmarini 1980) served to illustrate how firmly established are the opposing points of view, and how little movement toward reconciliation such debates as these seem to foster. Our model of language acquisition in the two-year-old (Hill 1982, 1983) has some bearing on the above controversy. The model represents a first attempt at building a vehicle which may be employed to examine and experiment with different hypotheses about language learning. Although it is obvious that computational models in and of themselves will not solve any controversies, nevertheless we believe that ultimately models such as ours will have a profound effect on the way in which questions will be posed and the answers that will be found. Even our simple model has already succeeded in stimulating its own healthy controversy.

2. Brief Overview of Model I

We will now describe our very simple model of the acquisition of English in the two-year-old and discuss some questions and some answers suggested by the use of the computational model in analyzing the Adam data which was gathered by Roger Brown, Ursula Belugi, Colin Fraser, and Courtney Cazden prior to 1973 (Brown 1973).

We can only give a very brief overview of the model and its assumptions in this paper. Readers interested in details of the model should consult Hill (1982, 1983) for full particulars of the model including examples of computer output together with corresponding linguistic data collected from a two-year-old child. The psychological validity of the model is defended in Hill and Arbib (1984) and in Hill (1984). The model is described as a member of the class of schema-theoretic models in Arbib, Conklin, and Hill (to appear). The model as described here is fully implemented. The development of the model, however, is an ongoing process. Figure 1 provides a diagram of the components of the model.

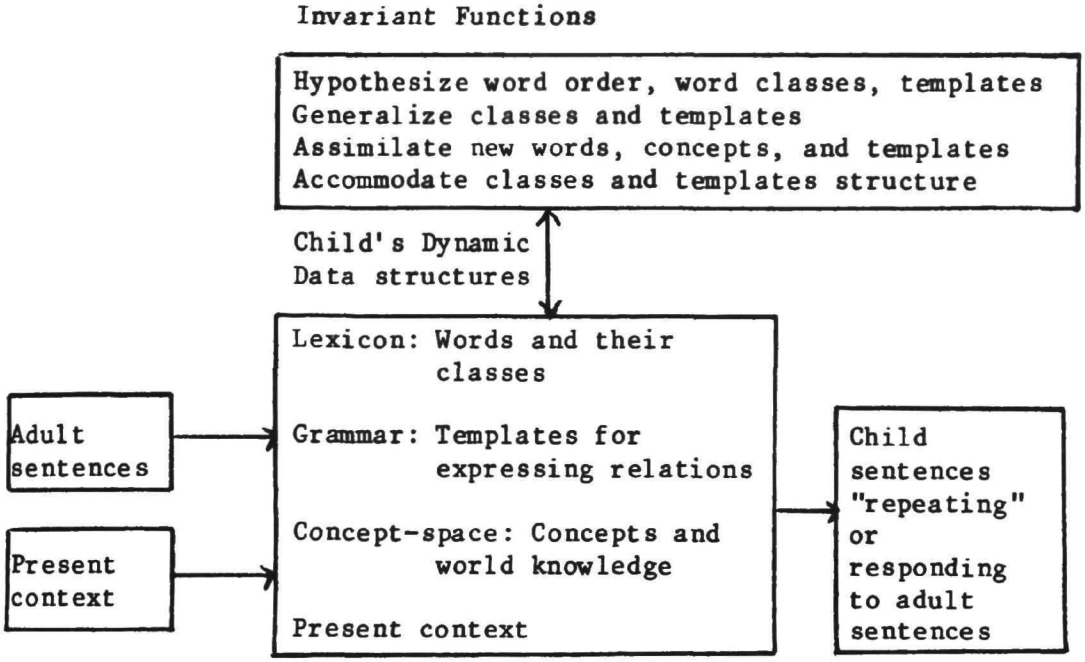


Figure 1. Components of the model

The model takes as its input adult sentences together with indications provided by the user, where relevant, of the physical context in which the sentences are uttered. Output from the model is a representation of child-like sentences repeating or responding to the adult input in accordance with the current state of the model's grammar. The child's knowledge is represented by dynamic data structures encoding the child's lexicon, the child's grammar, the conceptual knowledge of the child, and the physical context of the dialogue. No assumptions were made about the ultimate form of the adult grammar nor about what must be built-in to the model, but a precise account was kept of the knowledge and processes which were found to be necessary to be built-in to the model even for this elementary level of speech. The model notices and employs relations, word order, rules for concatenating relations and deleting words, and as the model grows word classes are formed. An important constraint on the model is that it does not depend on knowledge of adult word classes such as a division between actions and objects. The model requires schemas for word classification and template classification in order to grow, but the actual classes remain flexible and are inferred from the child's language behavior. The processes named above cause successive reorganizations of the grammar and the lexicon. The model suggests one way in which language based initially on cognitive knowledge can grow into a syntactic system which will be increasingly independent of its semantic and cognitive foundations. It is important to note that although the rules embodied in the model are simple, their interaction is complex enough to necessitate the use of a computer model.

The model in its original form (which we shall refer to as Model I) attained only the level of a two-year-old producing sentences of up to six words in length. The grammar of Model I (which we shall refer to as Grammar I) was entirely flat and was made up of rules for forming two-word utterances expressing relations and for combining those two-word relations into utterances which were two to six words in length. The grammar and lexicon

contained no articles and no conjunctions. The natural next question to be addressed by the model was how might the child progress from the flat grammar to a hierarchical one? As a means of examining this question we decided to explore the child's use of conjunctions.

3. Expanding Model I to Accommodate Coordinate Structure

We have examined the data collected by Roger Brown and his colleagues from the child Adam, from the age two years three months up to two years eleven months with particular attention to the use of the conjunction "and" by the adults in the transcribed sessions and by the child Adam. We used as input to our model the body of adult sentences which occurred in the transcribed sessions and which contained the conjunction "and." The output obtained from the model without any additional machinery quite readily matched the Adam data up to the time that Adam himself began to use conjunctions. What follows is a case study of the use of the model in exploring the child's understanding and formation of coordinate structure. The questions which linguists have addressed in the analysis of coordinate structure will be very briefly summarized. For a more complete discussion see Stockwell, Schacter, and Partee (1973).

1. One may employ conjunction reduction whereby redundant elements in a fully sentential coordinated sentence are deleted. For example

    John and Bill left

is derived from

    John left and Bill left

with the first occurrence of the word "left" deleted.

2. One may conjoin phrases and use the conjoined phrase as the subject of the verb "left."
3. One may employ some combination of these tactics.

In their article "The Development of Sentence Coordination", Tager-Flusberg, deVilliers & Hakuta (1982) argue that children's earliest phrasal conjunctions are not plausibly derived by conjunction reduction but are generated by directly combining like constituents by phrase structure rules. It will turn out that the model supports this argument but it should be noted that the claim is by no means universally accepted by linguists. (See for example Lust 1977, Lust and Mervis 1980.)

Our approach to building a model is to address the question of what must be added step by step to the model in order to parallel the developing language of the child. No one would imagine that the simple rules of Model I would suffice for any but the most elementary stage of language. The question we ask of the model is not if it will fail, but rather in what fashion it will fail, and most importantly can the model evolve in such a fashion that the more mature processes evolve from and grow out of the earlier processes. In the transcribed data, Adam at first frequently failed to respond at all to conjoined adult sentences. When Adam began to repeat conjoined utterances he repeated them by omitting "and." All these exchanges were developed by the model without any additional modification to the model's Grammar I or to the lexicon or concept-space of Model I. This was as it should have been.

At two years six months there occurs Adam's first use of a conjoined reply in response to an adult sentence.

Adult: Well, here, come here and cut your paper.

Adam: I come here a cut it.

Now, how can the model be modified to produce such an utterance? In Grammar I, "I" could be combined with "come" to produce "I come." "Come" could be followed by a slot filled by "here" producing "come here," and "cut" could be followed by "it" meaning "paper" to produce "cut paper" or "cut it." The first model could produce the response "I come here cut it" by forming the relations "I come," "come here," and "cut it." "I come come here cut it" was formed by concatenating these three relations and "I come here cut it" was formed by a deletion rule. There was no way in which "I come here a cut it" could be produced.

We tried to use Model I to conjoin the single words "come" and "cut" by adding "and" to the lexicon and adding the concept of coordination to the concept set. Grammar I rules yielded

I come come here come and cut cut it

which the deletion rules transformed into

I here come and cut it.

This was clearly wrong. No child would make such an error. It seemed that either we must devise different deletion rules for different situations by distinguishing between at least subject and object so that conjunction reduction could be employed, or alternatively that we must use the phrasal approach. Either way the data forced us to form a new template rule which conjoined relation pairs, not merely relation words or slot fillers. Thus this rule is the first hierarchical rule in a previously flat grammar. Moreover since this rule provided us with an alternative way to fill slots, the Grammar II is now more conveniently represented by a set of phrase structure rules than by the flat templates. We have given only one illustrative sentence here due to constraints of space but the arguments presented here are supported by many sentences in the data.

Since it is not a straightforward task to transform the template grammar to distinguish between subject and object we found that it was easier to form Model II by adding the ability to form conjoined phrases and use them as slot fillers. It appeared that this simple strategy sufficed for the Adam data. We cannot be certain of the validity of this strategy unless we painstakingly develop Model II by permitting it to evolve week by week from language samples of a single child as we did with Model I. It is an interesting fact about the model that, however, that the conjoined phrase strategy is much simpler to implement and is apparently adequate.

In summary, all the new coordinated responses could be accommodated by adding a new syntactic word class to the model and permitting hierarchical structure represented by the addition of a context-free phrase structure rule which took the Model I template

relation-word <u>slot-filler</u>

and called it simply relation, e.g.,

relation --> relation-word <u>slot-filler</u> and then added the rule

slot-filler --> relation "and" relation

The conjunction "and" was inserted as the first entry in the closed class in the Model II lexicon. Note that this additional phrase structure rule causes Grammar II to be recursive as well as hierarchical.

One interesting aspect of this case study of the addition of conjunctions to Model I which was flat, and which resulted in a hierarchical and recursive grammar for Model II, is that the machinery forced by the data

was contrary to that which we had anticipated when we first started to explore the addition of conjunctions. Since the model already had concatenation and deletion rules, before experimenting with the model we assumed that the conjunction reduction approach would be easier to implement. Experimenting with the model and comparing it to the data forced us to the realization that the current concatenation and deletion rules would not work for conjoined sentences and neither could they be easily modified.

In conclusion we would emphasize that we take the position that there is no need to assume that the hierarchical grammar evolves all at once. Neither is there any need to assume that hierarchy is triggered by conjunctions for all children. Mechanisms may appear in a different order in different children's grammars. The interesting observations offered by this case study are simply that experimenting with a computational model will often surprise the designer, will suggest new questions to explore, and will provide new insights which may complement those of traditional linguistic analysis. What can we conclude about the conjunction reduction hypothesis vs. the conjoined phrase structure hypothesis from this case study? Certainly the model cannot be regarded as offering conclusive evidence for one choice over the other. It is, however, interesting to note that in experimenting with this model it was far easier to produce sentences such as Adam's first conjoined sentences by employing the conjoined phrase analysis rather than the conjunction reduction approach, and, as anticipated, the addition of "and" to the grammar forced the development of a hierarchical grammar.

References

Arbib, M. A., Conklin, E. J., and Hill, J. C., to appear, From Schema Theory to Language, Oxford University Press.

Brown, R. A., 1973, A First Language: The Early Stages, Cambridge Mass., University Press.

Hill, J. C., 1984, "Combining Two Term Relations" Evidence in Support of Flat Structure," Journal of Child Language, 11.673-678.

_____, 1983. "A Computational Model of Language Acquisition in the Two-Year-Old," Cognition and Brain Theory, 6(3).287-317.

_____, 1982, "A Computational Model of Language Acquisition in the Two-year-old," Ph.D. Dissertation, University of Massachusetts at Amherst, reproduced by the Indiana University Linguistics Club, Bloomington Indiana, February 1983.

Hill, J. C., and Arbib, M. A., 1984, "Schemas, Computation, and Language Acquisition," Human development, 27.282-296.

Lust, B., 1977, "Conjunction Reduction in Child Language," Journal of Child Language, 4. 257-87.

Lust, B. and Mervis, C., 1980, "Development of Coordination in the Natural Speech of Young Children," Journal of Child Language, 7.279-304.

Piattelli-Palmarini, M. (ed.), 1980, Language and Learning: The Debate Between Jean Piaget and Noam Chomsky, Cambridge, Mass., Harvard University Press.

Stockwell, P., Schacter, P., and Partee, B., 1973, The Major Syntactic Structure of English, Holt, Rinehart and Winston, N. Y.

Tager-Flusberg, H., deVilliers, J., Hakuta, K., 1982, "The Development of Sentence Coordination, " in Language Development: Problems, Theories and Controversies, Volume I: Syntax and Semantics, edited by S. Kuczaj II, Lawrence Erlbaum Associates, Hillsdale, N. J.

A Model of Acquiring Problem Solving Expertise[1]

Dennis Kibler
Rogers P. Hall

Irvine Computational Intelligence Project
Department of Information and Computer Science
University of California, Irvine 92717

INTRODUCTION

Research on computational reasoning suggests that effectively organized knowledge, not search, is the key to expert performance. Paradoxically most work in the machine learning literature assumes that learning, a form of computational reasoning, occurs in a knowledge vacuum. Concepts are acquired through search in a space of candidate conceptualizations constrained by an implicit focus provided through representational "bias." For example, in early work on learning concepts from examples, a computational learner might have been implicitly constrained by having a restricted concept description language for assimilating carefully chosen examples. Thus learning was fundamentally disconnected from background knowledge. While early efforts at computational learning provide an illuminating starting point, our work is motivated by a recognition of the importance of multiple sources of background knowledge which human learners routinely bring to novel situations. In this paper, we describe a computational model of learning problem solving skills which proceeds by tentatively connecting new situations with existing knowledge through a process of analogical reasoning. This model is in early stages of implementation.

LEARNING PROBLEM SOLVING

The nature of problem solving expertise has also interested psychological researchers. By carefully contrasting the behavior of novice and expert problem solvers, psychological research reveals that competent problem solvers construct a form of intermediate semantic representation while understanding a problem text but before attempting to generate quantitative solutions (Chi, Feltovich and Glaser, 1981; Larkin, 1983). This intermediate representation appears to consist of abstract conceptual entities (e.g., force or momentum in physics) which are related to strategic methods through problem schemata reflecting problem categories (e.g., conservation of momentum).

As a goal for a computational model of acquiring problem solving skills, expertise in problem solving can be viewed as the possession of an abstract **conceptual vocabulary** tailored to the particular problem solving domain and an assortment of **problem schemata** reflecting problem categories. These schemata provide a mechanism for retrieving appropriate problem solving strategies when a problem from a particular category is encountered. Retrieval is based on conceptual cues which must be inferred from the problem statement. Acquisition of problem solving expertise in some domain then amounts to learning these problem schemata and the conceptual vocabulary out of which they are constructed.

Solving algebra story problems

The problem domain chosen for this work is that of learning to solve "story" problems typical of algebra instruction at the elementary and high school level. For example, consider the following "triangle" problem:

> Jerry walks 1 block east along a vacant lot and then 2 blocks north to a friend's house. Phil starts at the same point and walks diagonally through the vacant lot coming out at the same point as Jerry. If Jerry walked 217 feet east and 400 feet north, how far did Phil walk?

This is called a "triangle" problem since its solution rests on relating the problem to some basic facts about triangles. Mayer (1981), analyzing a large sample of problems appearing in secondary school texts, presents a clustering of problem types consisting of approximately 50 distinct problem templates organized as a simple classification hierarchy. Levels in the hierarchy correspond roughly to families of algebraic equations underlying sets of problems and the "story lines" on which problems are based. For example, in the "amount-per-time rate" family, a variety of story templates are evident including motion, current and work problems. Although the intent of Mayer's study is to document classification schemata which students reliably use to interpret and then solve algebra story problems, identification and description of these problem schemata provide a rare opportunity for choosing a domain in which there exists a substantial body of descriptive and experimental literature based on human performance (e.g., Hinsley, Hayes and Simon, 1977; Mayer, Larkin and Kadane, 1984; Kintsch and Greeno, 1985).

[1] This work was supported by a gift from the Hughes AI Research Laboratory, a division of Hughes Aircraft.

Multiple knowledge sources

We will assume that the story problem is presented to the learning system as a relatively flat set of propositions without an explicit representation of inter–connecting structural relations. For the triangle story mentioned above this representation might be:

event(e1, walking)	*agent(e1, Jerry)*	*distance(e1, 1 block)*
	direction(e1, east)	*trajectory(e1, along lot)*
event(e2, walking)	*agent(e2, Jerry)*	*distance(e2, 2 blocks)*
	direction(e2, north)	*destination(e2, friend's)*
event(e3, walking)	*agent(e3, Phil)*	*source(e3, ?x)*
	trajectory(e3, through lot)	*destination(e3, friend's)*
	distance(e1, 217 feet)	*distance(e2, 400 feet)*

with the goal of finding *distance(e3,Answer)*. This representation is relatively complete with respect to the literal problem statement, includes information which is extraneous to the desired solution (e.g., that Jerry walks three blocks), but also excludes information which must be inferred if a solution will be found (e.g., that a right triangle has been described). Representations of similar problems described in the psychological literature (e.g., Mayer, 1981) include inferences from the data and abstractions of important details without explicitly describing how such processes take place.

In addition to choosing a representation for problem statements at the point of input, we must also describe the types of knowledge which will constitute the "multiple knowledge sources" essential for problem understanding and solution. Fortunately, the task domain of solving algebra story problems provides a finite but sizable collection of requisite sources of knowledge. On the basis of an informal content analysis of problems occurring frequently in Mayer's (1981) classification taxonomy, we will assume that a novice problem solver must have knowledge of the objects which routinely appear in problem statements, knowledge of a factual variety regarding these objects, and limited inferential abilities regarding space and time. These knowledge sources are assumed to be relatively stable during the course of learning to solve algebra story problems.

Two sources of knowledge remain, and provide the basis for what is to be learned. First, the system is to acquire a practical **conceptual vocabulary** for primitive events which are central in story problem texts. Examples are simple motion or work events, which we treat as frame–like structures that serve to organize objects described above. Second, the system is to acquire a variety of solution methods organized as **problem solving schemata**. A schema consists of a cue expressed in terms of the entities and relations of the problem description and a set of quantitative or qualitative constraints. The cue serves to facilitate recognition of situations in which the constraints are applicable. For example, matching a schema cue with a problem description in which two simple motion events occur in an opposite direction might invoke a decomposition of the problem into additive components. Event frames and problem schemata correspond to the components of problem solving expertise described in the introduction of this paper.

A PROBLEM SOLVING ARCHITECTURE

Our model of solving algebra story problems depends upon the cooperation of multiple knowledge sources operating in an asynchronous fashion through a globally accessible problem description. As shown above, the initial problem description is a jumbled set of propositions with relatively little structure. From an abstract perspective, problem solving proceeds by using existing knowledge sources to manipulate this problem description until sufficient quantitative constraints are available to allow calculation of a solution. A solution path in this space of manipulated problem descriptions consists of increasingly coherent descriptions of the current problem generated by the actions of applicable knowledge sources.

Although the problem solver may utilize a large number of knowledge sources, we can divide these various knowledge sources into four functional types. While in the previous section we described background knowledge sources primarily in terms of their content (e.g., methods, events or time), we now describe knowledge sources in terms of their role in problem solving. These include: **augmenters, organizers, asserters,** and **decomposers**. Augmenters add propositions to the problem description by applying background knowledge. In the example problem presented earlier, information about triangles would be added by augmentation. Organizers add structure to the problem description by grouping disconnected propositions into frame structures. In the example, Jerry and Phil walking from source and destination points signal that simple motion-events have occurred. Expectations encoded in the frame structure for motion-events serve to organize the problem description. Asserters contribute quantitative constraint relations that involve a single organizing frame. For example, the equation $d = r * t$ would be asserted given the cueing attributes of a motion-event, known rate and time, and unknown distance. Decomposers transform a problem into quantitatively related subproblems, yielding a constraint equation involving more than one organizing frame. For example, in a motion problem involving several driving events, the problem might be decomposed into the sum of component distances.

The problem solver schedules activities of the various knowledge sources by a competitive process which attempts to expend the least effort while still progressing towards a goal state. Augmentation and organization are least costly, but do not generate any equations, and therefore may not necessarily lead

closer to a problem solution. Instead their contributions must be focused towards enabling and confirming other knowledge sources. Assertion and decomposition, although more complex, are the only operations that generate equations. They must be applied to achieve the goal. Since enabling conditions for any of the four types of knowledge sources may be only partially matched by a current problem description, we must often treat a knowledge source proposing a modification to the problem description as a hypothesis which is subject to confirmation. This is precisely the role of analogical reasoning in this model of problem solving.

Analogical Transformations in Problem Solving

In this section, we describe how recognition, elaboration and confirmation of analogies serve to connect a problem description with applicable problem schemata. The central goal of analogical reasoning is to alter the representation of the current problem so that it is partially equivalent to a previously experienced problem for which a solution method (either equations or a suitable decomposition) is known.

Analogical reasoning, then, is a process of viewing an unsolved problem *as if it were* a problem for which a solution strategy is already known. "Viewing as" in this context involves extending information from the solved problem into the description of the new problem, subject to some form of critical evaluation within the confines of the new problem. Given the importance of matching a developing problem description with known problem solving methods, analogical reasoning represents a point midway in a continuum between literal similarity and nonsensical (or anomalous) comparison. Under this assumption, reasoning and learning processes undertaken while viewing one problem as if it were another more familiar problem are comparable in kind to processes undertaken when a new problem is recognized as an instance of a known problem class.

Recognition of opportunities for transformation

Recognition of a potential analogy within our process model amounts to tentative acceptance of a problem schema on the basis of a partial match between the current problem description and a problem cue within the schema. In isolation from other processes, it is easy to imagine some form of search among existing problem schemata to find a subset of schemata which bear a promising resemblance to the target problem. The computational literature on analogical reasoning is sparse with respect to proposals for constraining such a search. Most promising is the notion of indexing potential analogs on the basis of abstract relational information (e.g., Carbonell, 1981 and Kolodner, 1983, 1984) so that recognition and retrieval will be based on higher level (and presumably more predictive) aspects of similarity. In the domain of algebra story problems, a higher level (or "systematic," according to Gentner, 1982) correspondence might be exemplified by a match between a simple motion and a simple work problem which involves some sort of event (e.g., a motion or work event), a rate and a single unknown. Such higher level descriptive aspects might be distinguished from surrounding information (e.g., the name of an agent or destination) *a priori* through supporting domain knowledge or by virtue of participating in the method associated with a problem cue in a related problem schema. Recognizing an applicable problem schema is a central activity in our model of problem solving, an activity to which processes of augmentation and organization are explicitly directed.

Elaboration and evaluation of prospective analogs

We hypothesize that recognition of analogies is identical to recognition of directly applicable problem schemata except with regard to the effort expended in confirming the match between problem descriptions. Confirmation is attained by incrementally elaborating the correspondence between aspects of a potentially applicable knowledge source and the current problem description. As aspects of the source description (e.g., organizing structural information) are extended to the problem description through elaboration, the validity of these extensions must be evaluated both in terms of the specific nature of the current problem and in terms of inferential capabilities over objects and relations in the domain of algebra story problems. In the same sense that augmentation and organization processes were described as bringing supporting knowledge sources to bear in understanding a new problem, so too can these processes serve in confirming tentative information extended from source to target problem descriptions.

For example, when confirming an analogy between a "work together" problem (two agents work on a single job together) and an "opposite direction" motion problem (agents travel in opposite directions from the same source point), knowledge of equal duration within the motion problem might be extended to the work problem by virtue of being integrally involved in the solution strategy of decomposition. The tentatively held assertion of equal duration in the work problem (extended from the motion problem) must be confirmed on the basis of knowledge which can be generated with augmentation processes (i.e., that time intervals may be additive). Similar sorts of confirmation could be achieved with organizing processes.

LEARNING MECHANISMS

The goals of learning, as described earlier, are the acquisition of problem schemata and the specialized conceptual vocabulary of which they are constructed. Broadly speaking, the learner's conceptual vocabulary (i.e., frame–like descriptions of events) gradually changes as the result of problem solving experience and instruction. Direct instruction will be involved in the extension or correction of other knowledge sources, while learning when to apply background knowledge (i.e., acquisition of problem schemata) is the learner's responsibility. Hence our computational model of learning to solve algebra story problems involves a variety of forms of learning including learning by being told, learning by taking advice and learning from examples.

The learning mechanisms we propose satisfy a number of constraints. First, they are all incremental. Thus, problem solving expertise is acquired gradually through active experience with a succession of problems and instruction presented by the teacher. Second, learning is tolerant of errors: correctness of acquired concepts or schemata is relative to problem solving experience rather than being defined in absolute. Concepts constantly change and evolve according to their problem solving utility. Third, newly acquired knowledge is connected to old knowledge by the recognition, elaboration and confirmation of analogies. Thus learning does not occur in isolation from existing knowledge of the task domain.

We now briefly consider some examples of learning processes, each example based on simple problems drawn from Mayer's (1981) taxonomy. These examples demonstrate direct interaction of instruction, inferences based on background knowledge, and problem solving experience. The first example shows how the learner refines initially naive concepts so that they are more appropriate for problem solving. The second example shows how analogy can drive the transformation of previously learned concepts and schemata into new problem solving knowledge.

Learning through instruction and experience

Before beginning, we need to describe the existing knowledge state of the learner. Suppose that the learner has no problem schemata and has a naive concept of a motion–event given by the following frame:

Motion–event: **agent:**
vehicle:
to:
from:

Assume the system is presented with the following problem:

> Example 1: *Bill Less drove from Boston to Cleveland, a distance of 624 miles, in the time of 12 hours. What was his driving speed?*

Without explicit knowledge of problem types and solutions strategies (i.e., problem schemata), the learner is unable to solve any problems. Hence we allow a teacher to directly instruct the learner to use the equation $distance = rate * time$. Assuming the learner can correctly solve this problem, a number of modifications are possible for existing knowledge sources. First, the naive concept of a motion-event changes to include additional slots for distance, time and rate, since these were used in the problem solution. In addition, distance, rate and time slots are distinguished as having played an essential role in the solution process. Other slots (e.g., agent) are noted as potentially inappropriate since they did not play a role in the solution. Thus the salience of slots in an event frame with respect to achieving problem solutions is recorded. Second, the learner forms a problem schema which associates with the cue, motion-event and goal to 'find rate', the method of using the equation $distance = rate * time$.

After this episode involving direct instruction and problem solving experience, the learner's concept of a motion-event would include:

Motion-event:	**agent: (n/a)**	**distance: (essential)**
	vehicle: (n/a)	**rate: (essential)**
	to: (n/a)	**time: (essential)**
	from: (n/a)	

Note that the learner has marked the *to* and *from* slots as being not applicable (n/a) even though these may prove important during later problem solving. In fact, even if the learner deleted these slots from the representation of a motion event altogether, recovery would be possible by adding the slots again on the basis of additional experience.

Assuming further problem solving experience with simple motion problems, the system's motion-event concept might evolve to look like the following where all slots are implicitly marked "essential":

Motion-event:	**distance:**	**start-position:**
	rate:	**end-position:**
	duration:	**start-time:**
	direction:	**end-time:**

The process of adding or reinforcing slots that are used and demoting slots that are not used allows the learner to progress without an absolute notion of concept correctness. Concepts are modified in a fashion which reflects their utility. Rather than mere occurrence, inclusion and survival of components of a concept (i.e., slots) depends on their participation in the problem solving process.

Learning while reasoning analogically

We now consider a second example which demonstrates underlying processes of analogical reasoning as a guide to learning. Assume for this example that the learner is capable of solving simple distance-rate-time problems and is presented with the following problem:

> Example 2: *A fisherman can catch a fish every 20 minutes. If he spends an 8-hour day fishing, how many fish will he bring home?*

We will also assume that the learner possesses a naive notion of a work-event represented as:

Work-event:	**agent:**	**pay:**
	job:	**instrument:**

The learner can recognize that this problem involves a work event, but has no problem schemata suggesting solution methods for this type of problem. In fact, it is not even possible to integrate all aspects of the problem description (e.g., the *rate* at which fish are caught) within the work event frame. However, since we are assuming that the learner has a relatively well–developed schema for simple distance–rate–time problems, it is possible to construct an analogy between this schema's cue and the current problem description. Specifically, recognition might occur on the basis of a partial match in which rates, durations and unknowns are placed in correspondence. Elaborating this analogy, work duration and rate could be associated with the naive work event frame, and a tentative correspondence between distance and work output could be formed. Thus, by viewing the simple work problem as if it were an instance of the relatively well–known class of distance–rate–time problems, the learner could attempt to solve the equation $output = rate * time$. Since this strategy yields a correct answer to the current problem, a new problem schema can be formed which associates unknown(work) and work-event with the successful equation.

Note that in this second example, the learner recognizes an analogy between problem types on the basis of a relatively remote partial matching between a schema cue and the current problem description. Although the initial kernel of correspondence is tentative, further elaboration serves to reorganize the problem description and a successful solution attempt confirms the initially tentative analogy. As a result, both the learner's conceptual vocabulary and repertoire of problem schemata are enlarged. In this example, however, recognition of the analogy between simple motion and work problems depends on a rather liberal allowance for entertaining partial matches. If the learner were allowed to attempt all such possible analogies, exhaustive search of weak partial matches over existing problem schemata cues could prove unreasonable. These problems await further study.

CONCLUSION

The preceding sections illustrate how a conceptual vocabulary and problem schemata can be learned through a combination of instruction, analogical reasoning between similar problem types, and problem solving experience. Acquisition of problem solving skills is incremental, pragmatic, and strongly connected to existing knowledge sources. Problem solving is modeled as a process of problem understanding. Through inferences supplied by processes of augmentation, organization, assertion and decomposition, problem descriptions become increasingly coherent and constrained. Analogical reasoning allows tentative application of known solution methods, subject to verification by surrounding knowledge sources.

REFERENCES

Carbonell J.G. (1981) Invariance hierarchies in metaphor interpretation. *Proceedings of the 3rd Annual Conference of the Cognitive Science Society*, 292–295.

Chi M.T.H., Feltovich P.J and Glaser R. (1981) Categorization and representation of physics problems by experts and novices. *Cognitive Science 5*, 121–152.

Gentner D. (1982) Are scientific analogies metaphors? In D. Miall (Ed.) *Metaphor: Problems and Perspectives*. New Jersey: Humanities Press.

Hinsley, Hayes and Simon (1977) From words to equations: meaning and representation in algebra word problems. In P.A. Just and M.A. Carpenter (Eds.) *Cognitive Processes in Comprehension*. Hillsdale, New Jersey: Lawrence Erlbaum and Associates, 89–106.

Kintsch W. and Greeno J.G. (1985) Understanding and solving word arithmetic problems. *Psychological Review 92*(1), 109–129.

Kolodner J.L. (1983) Maintaining organization in a dynamic long–term memory. *Cognitive Science 7*, 243–280.

Kolodner J.L. (1984) Towards an understanding of the role of experience in the evolution from novice to expert. GIT–ICS–84/07.

Larkin J.H. (1983) The role of problem representation in physics. In D. Gentner and A. Stephens (Eds.) *Mental Models*. LEA: Hillsdale, N.J., 75–98.

Mayer R.E. (1981) Frequency norms and structural analysis of algebra story problems into families, categories, and templates. *Instructional Science 10*, 135–175.

Mayer R.E., Larkin J.H. and Kadane J.B. (1984) A cognitive analysis of mathematical problem–solving ability. In R.J. Sternberg (ed.) *Advances in the Psychology of Human Intelligence*. LEA: Hillsdale, New Jersey, 231–273.

Creating and Comprehending Arguments

Stuart M. McGuigan

Yale University

John B. Black

Teachers College, Columbia University

Discussion and dispute are everyday activities for most of us. Impassioned or coldly rational, the process of argument is central to our relations with others and to our understanding of the world in general. The way we make and support conclusions reflects not only our capacity for reason but the way we structure any type of knowledge. Much of the argument we come across is in the form of small bits of text. Advertisements and editorial are explicit attempts to influence our choices in what we buy or how we vote. In other types of expository text, such as in newspaper or magazine articles, the argument is more subtle and often appears as description. Common to all is the statement of premises and the evidence used to support them.

Arguments are written to serve two goals: to communicate ideas and to persuade others that those ideas have value. The relative weight of these goals determines how the writer will structure the text. The *argument structure* is the organization of support and conclusions that represents the line of argument. If a writer's ideas are controversial or his or her audience is unreceptive, much attention will have to be given to the argument structure. Potentially weak assumptions must be buttressed and tentative premises substantiated. Undisputed points still require support but the amount of text devoted to plain description can be proportionately larger; the writer need not go far back in the chain of reasoning to support his conclusions. In arguing for an increase in military spending to the joint chiefs of staff, the president would only need point to the value of a strong defense. When arguing the same issue to congress, however, he may wish to provide evidence for accepting the premise "strong defense is a good thing". What was to one group an acceptable assumption was to the other a conclusion requiring support.

The Structure of Arguments

Upon examination of a large number of arguments, we found that support structures can be divided up into three basic types: argument by analogy, categorical argument and causal argument. The distinction between the first two is the difference in level of generality between the support and conclusion. In an argument by analogy, the conclusion and support are at the same level. Cats can be compared with dogs and one president can be compared to another. If I wanted to argue that Reagan's reform will have little lasting effect on the income tax process, I could point out that Eisenhower's attempts to simplify the tax structure had little effect. Eisenhower's case is analogous to Reagan's, both are examples of American presidents who tried to change the federal income tax system. Any other relevant similarities, such a both being Republican s, strengthens the effect of the analogy. With enough examples, an arguer can create a category that contains the features common to all the examples. Conclusions that come from generalization across several examples are the product of induction. The reverse process is deduction. The arguer has a category more general than the conclusion, attributes category membership to the object or idea, and by virtue of membership attributes other features of the category to the object. This form can be expressed as a syllogism. " President Reagan has a tax reform plan. All presidential tax reforms have little lasting effect. Therefore, Reagan's tax reform will have little lasting effect. "

Reagan's tax plan is a member of the category *presidents' tax reform* which has the feature *has little lasting effect*. The same point could have been made with both inductive and deductive support. The arguer could enumerate other examples of failed presidents' tax reforms, induced the category,

asserted membership for Reagan's plan, and then attributed the feature as the conclusion. The third type of support uses knowledge of cause or process to support the conclusion. Causal knowledge can exist at any level of generality, from what a specific president does at one moment to how people in positions of power are likely to behave at any time. Whereas categories or analogies are a slice of time or atemporal, cause and effect involve temporal sequence. We use the phrase "cause or process" to avoid the philosophical debate on the nature of *causality*. For purposes of argument, causes precede effects in time and allow either prediction of the effect given the cause or given the effect, explanation of its occurrence through the hypothesis of a cause. For example, Reagan's plan will fail because special interest groups will pressure congress to make myriad exceptions to any general simplification. The pressure must precede the change or the argument is not a causal one. These three support types are complimentary. Categories can be formed or modified from the examples used in analogy. Causal knowledge can be (and is ultimately always) derived from analogies and con be use to construct and alter categories.

The three support types do not exist merely for arguments sake but are part of the natural understanding process. People notice similarities are found between two objects or events. The first event or object provides the information for understanding the second. With only two examples, people cannot determine which features are important and which accidental, yet they can still generate expectations that will help guide further processing and later behaviors. When people have several examples of an object or event stored in memory, they can more efficiently retrieve the information necessary for processing a new instance if the features shared across instances is collected in a category. Though it seems obvious that any understanding system must make generalizations, the ubiquity and flexibility of the process of category formation are important for understanding arguments. The chain of potential hierarchies can be long and complex. Categories themselves can be grouped to form more general categories or can be split up to form subcategories that entail a more specific collection of features. What level of generality the arguer can begin with depends upon his audience. If the receivers of the argument are accepting, the arguer can begin close in level to his conclusion. If the receivers are not, however, the arguer will need to move further up or down the hierarchy. Moving up the hierarchy allows the arguer to derive her conclusions deductively. Moving down provide her with evidence to induce her conclusions. Causal or predictive knowledge concerns the ordering of events and is potentially the most useful. Knowing that earth tremors can predict volcanic eruption or that hitting the keys "control" and "C" stops the run of a loop can be to islander or cognitive scientist.

Counter Argument

People create counter-argument with the same processes used for constructing arguments. Perhaps the most common counter-argument is the counter-example. If you compare Reagan to Eisenhower, I could point out that a comparison with Nixon is more appropriate. Both are Republican s but Nixon's term of office is closer in time to Reagan's. You could use Ford as another analogy to counter my counter, but this leaves the possibility of another counter-analogy, using another criterion for similarity than temporal proximity. Instead, you can attack my counter-analogy and defend your initial analogy by examining the relevant features. Eisenhower and Reagan have more ideological similarities, both have shown concern for the growth of government and with the wage-earning tax-payer. A good analogy, therefore, is one which shares *relevant* features with the original instance. The number of distinguishing features can be compared to the number of shared features to produce an index of similarity. Relevance is determined by the feature under debate or the current needs of the understander. Arguers need to know what features discriminate between analogies to judge which comparisons are most apt.

For answering categorical arguments, two strategies are available. Initial category membership can be shown to be inappropriate by a feature conflict or another category can be shown to be more

appropriate due to greater relevant feature overlap. The argument " All Republican president's tax reforms are doomed. Ronald Reagan is a Republican president. Therefore Reagan's tax plan is doomed. " can be countered by showing that in fact there are two types of Republican s, conservative and liberal. Perhaps only the tax plans of liberal Republican s are doomed to failure and Reagan is a conservative Republican .

Causal arguments are vulnerable along any step of the causal chain beyond that which is known explicitly. The chain " Republican presidents create plans for tax reforms, give the plan to congress which alters the plan for special interest groups due to lobbying pressure. " is vulnerable at a number of points without contradicting any of the facts in the series. Special interests may not always conflict with good tax reform or the reforms are too watered down by the president before they even reach congress. Although knowledge that allows prediction to future events may be the most desirable, it is also the most fragile.

The Memory Representation of Arguments

We have developed a model of argument generation and comprehension (MAGAC) that simulates how people create and evaluate arguments. We use a frame architecture to represent the knowledge and procedures for filling slots and linking frames. We will now work through an example argument. Suppose the reader is confronted with the following text:

> Computers have affected all aspect of everyday life in the United States. The introduction of new technologies, over a short period of time, causes a disruption of the current, established culture. Therefore, the spread of computers will result in great unhappiness for many people.

The first sentence introduces the domain, the recent inroads of of computers into everyday life. This has been a fairly controversial topic and has received some play in the press. When the text is fully processed, the representation will be organized around the conclusion in a conclusion organization frame (COF). After the first sentence, the representation looks like COF1.

```
COF1: INTRODUCTION OF Computers
AKO: New Technology
TIME: Current
TIME COURSE: Short
RESULT: ?
```

The heading slot for COF1 is the "introduction of computers". The *A Kind Of* (AKO) slot tells of what more general class of things this specific knowledge is a member. The TIME slot indicates the time frame to which the knowledge or event being represented belongs. As the INTRODUCTION of an object is a process, it must take place over a period of time. The slot TIME COURSE is filled with information concerning how long an event took or will take in the future. In this case, "short" is a term relative to the introduction time of other technologies, such as the manufacturing of steel or the production of clocks, both of which took hundreds of years. The effects of the process represented or procedures for determining the effects are stored under the RESULT SLOT. Goals and plans of people or entities are stored under GOALS and PLANS SLOTs. The actual representation of the goals and plans would be stored elsewhere.

The second sentence of the argument gives the reader a general information which is represented in of SOF1. SOF1 is an instantiation of a "support organization frame."

```
SOF1: New Technology
AKO: Social Change
PROPERTY: origins in science
GOALS: Achieve new goals, previously blocked
       or
```

```
        Achieve old goals more efficiently

RESULT: (change in work force)
        (less resources needed to accomplish
         goal)
        ((If TIME course is short
           then disrupted culture)
         (If TIME course is long
           then no disruption))
```

Even if the receiver has not heard this argument before, it will not conflict with related knowledge. New objects or processes cause change and change can be disruptive. From this knowledge, SOF1 could be constructed for the purpose of understanding and storing the argument should it be encountered again. The goal knowledge is also very general. Technology is meant to achieve new goals, such as rapid communication by satellite, the efficiency of achieving old goals, such as generating power from nuclear reactions. Such improvements are accompanied by problems and dissent. The third sentence forms the conclusion. The reader already knows from experience that people often find change upsetting, especially swift and pervasive changes. SOF2 is constructed or instantiated to handle this information.

```
SOF2: Change
AKO: Cultural Process
PROPERTIES: disruptive
          creates unhappiness
```

Several general procedures are necessary for connecting the three frames. The first procedure is that which connects the topic to the evidence. For this example, inheritance from the AKO (a kind of) slot is used to fill empty slots. If we know that X is a kind of mammal but nothing else about it is explicitly known, We could assume default characteristics of a general "mammal" which would be stored in that higher level frame. In this case, "computers" are a kind of new technology. The result slot of "computers" is open and thus can be filled with the result slot of "new technology". In this causal argument, the process is applied and the result is calculated.

Procedure 1

If *SLOT* is empty then fill with corresponding RESULT *SLOT* from AKO *FRAME*.

Next, the connection between "new technology" and "social change" must be found. In this case, *procedure 1* again provides the link. New technologies are one kind of social change. It is important that it be a kind of social change (rather than personal) so that the generalization in the conclusion is warranted. The result is not given but requires calculation because it is dependent on the time course of events. Gradual social change is not disruptive. We believe it makes little sense to have this procedure stored outside the frame in a general list for two reasons. First, given a large number of special cases, the list would become unmanageable as more domain specific knowledge is added. Second, this information is only needed under "social change". Though other kinds of swift change (i.e geological) may be disruptive, it is not in the same way. The knowledge for calculating the result is stored where it is accessed automatically with the declarative knowledge.

The same conclusion could have been supported in a different way. Instead of referring to a general causal principle, the writer could have made an analogy to when something similar had happened in the past. In the absence of more definite knowledge, a similar instance can be used to understand a new one. This process of understanding is often used in argument. The analogy shows that X *can* have quality Y, that this is (at least) a possibility. In our example, the first and third sentences of the

paragraph remain the same but the body of the argument is changed to: " Automatic looms were also considered a great innovation in their day and resulted in a redistribution of the work force. " The comparison of computers and the automatic loom is based on their both being an example of a new technology that have radically affected the surrounding community. The implication is that use of computers will also cause loss of jobs. From there, it is a fairly straight forward inference that people will be unhappy. The representation for "Automatic Loom" looks like SOF3.

```
SOF3: INTRODUCTION OF Automatic Loom
AKO: New Technology
TIME: Past
TIME COURSE: Short
RESULT: Unhappiness
```

Both the AKO and TIME COURSE slots match those for COMPUTERS frame. If two frames have the same AKO SLOTs, then they are both members of the same general category and may be compared. No alteration of the category structure will occur nor is any feature from the category frame being used to make the argument. The more slots two frames share the stronger the comparison. If one frame has a slot filled that the other has blank, an IF-NEEDED connection is made from one slot to the other. In this case, the result slot from COMPUTERS can be filled with that from AUTO LOOM, if the information is desired. The slot filling is not a logical necessity and this is reflected in the IF-NEEDED connection. The result here, redistribution of the work force, is AKO social change. The conclusion derives from the frame SOCIAL CHANGE's RESULT slot. The connections in this argument are not as direct as in the previous one. The procedure for connecting the frames is:

Procedure 2

If AKO *SLOT* of X matchs AKO *SLOT* of Y then create pointer from X to Y.

The deductive or categorical argument has the traditional syllogistic structure underlying it. First category membership is asserted for the object or idea being debated. Then, by virtue of membership, other features from the category are attributed to it. With the same topic, a categorical argument would be

> New technologies are a feature of social change. Social change is alway disruptive for those living in the current established culture.

On the surface, this support is similar to the causal or analogical argument and a similar causal structure underlies this category. The representation and procedures of this argument, however, are different. Unlike analogy based arguments, the AKO slot is used to proceed to a higher level, in this case to the frame SOCIAL CHANGE. As a member of the category SOCIAL CHANGE, the frame COMPUTERS inherits any slots that do not contradict existings ones. The relationship is a necessary one, if something is a member of a category, then it must possess features *stored* at the category level. The slot is filled with procedure 3.

Procedure 3

Fill empty COF* slot with PROPERTY slot(s) from SOF* indicated in AKO slot.

The three procedures represent the way frames are connected for the three basic supports. These same processes are general enough for other uses and capture our intuitions of how knowledge is structured. The model handles counter-argument as the creation of new arguments or new links to existing structures. We will expand this model to include an evaluation of relevance to determine which features are most important in each case of calculated similarity. Argument provides a domain for exploring the creation and modification of knowledge structures and dynamic interplay of new information and past experience.

LEVELS OF GOAL DIRECTION AND THE CAUSES OF LEARNING

Dale M. McNulty

Irvine Computational Intelligence Project
and
Center for the Neurobiology of Learning and Memory
University of California
Irvine, CA 92717

ABSTRACT

A general purpose model and learning program are described which account for the phenomena of latent learning and irrelevant-incentive learning. The model is composed of three separate goal pursuit levels. At the lowest (latent) level are the constant, implicit goals associated with the system's memory management mechanisms. At a higher (overt) level are the dynamic, explicit behavioral goals which the implicit goals enable by manipulating memory representations to conform to the external surroundings. At the highest (meta-overt) level the latent level employs knowledge as meta-schemata. Functional specifics which enable the program to demonstrate the learning behavior in a maze environment are discussed.

1.0 Introduction and Relevant Historical Perspective

Prior to the 1930's psychologists associated learning with performance. For Pavlov, Thorndike, Hull, Spence, etc. learning was a reinforcement of stimulus-response associations. In 1924 Simmons observed a contrary phenomenon. Unrewarded rats allowed to explore a maze were just as capable of finding their way to food during subsequent rewarded trials as animals which were rewarded continuously. Blodgett (1929) termed this learning "latent" because the animals acquired maze knowledge in the absence of any demonstration of learning. Blodgett's conclusion contradicted the firmly entrenched Law of Effect, and Hull's theory of drive reduction, which require behavioral reinforcement to strengthen the S-R connections of learning. Early cognitivists (Tolman and Honzik, 1930) explained Blodgett's conclusion in terms of cognitive knowledge of goals and goal acquisition. The vast majority of modern theorists continue to consider learning to be goal directed.

The split from earlier S-R theory is a result of two important observations from the latent learning experiments (Thistlethwaite, 1951). First, reinforcement, such as food or water, is not necessary for learning to occur. Second, learning should be distinguished from performance. Knowledge need not be manifest for learning to occur. Information is often 'known' and not demonstrated until the proper motivation is presented. Bearing these distinctions in mind, it is possible to recognize two different paradigms which need to be explained by learning models. One type of experiment utilizes unrewarded trials, or exploratory behavior, to test the subject's ability to learn how to arrive at a location that has been associated with a reward. Experiments of this type are testing latent learning. A second type of experiment tests the subjects' ability to learn how to arrive at the location of an object that was irrelevant at the time of exposure. An example of an irrelevant object is food which has been rendered irrelevant by feeding subjects to the point of satiation. Experiments of this sort test irrelevant-incentive learning.

A program capable of both types of behavior will be described here. For the sake of simplicity, these two experimental paradigms will only be distinguished in this discussion where specific program operations distinguish the two.

2.0 Levels of Goals and the Causes of Learning

Since learning does occur in the absence of reinforcement and without associated performance, previous learning theories relying on reinforcement are not satisfactory. In formulating new models the obvious question to ask is: what are the causes of learning? Four possible explanations come to mind:

[1] Learning is not goal directed. This position is difficult to conclusively prove or disprove, however the nonrandom nature of learning suggests that learning is either goal directed, or, that animals attend to and store every experiential event. The later does not seem supportable, so it must be concluded that learning is, in some way, associated with goals.

This research was supported in part by the National Science Foundation under grant IST-81-20685 and the Naval Ocean Systems Center under contracts N00123-81-C-1078 and N66001-83-C-0255.

[2] Subjects in the latent learning and irrelevant-incentive learning paradigms receive reinforcement in ways other than feeding or drinking and, thus, the Hullian Model of stimulus-response adequately explains the phenomena. A number of authors have argued that the rats learn the maze because they are driven by such things as a need for handling or a desire to escape the maze. The experimental results are very complex and the possibility for many types of reinforcement can not be totally ruled out, but as Thistlethwaite shows, reinforcement does not adequately explain the entire latent learning effect. Many others, including Mackintosh (1974) and Bower and Hilgard (1981), have also effectively dismissed the reinforcement arguments, thus, the consensus is that latent learning is a real phenomenon which is not explained by the Law of Effect or reinforcement theory.
[3] A modified version of the Hullian Model of stimulus-response explains the latent learning phenomena. This position is not supported by the evidence. Again Thistlethwaite, Mackintosh, and others effectively argue against this. The S-R model of learning is generally not accepted today, so the argument will not be carried on here.
[4] Learning is goal based, but there are multiple levels of goal pursuit, some of which have not been previously recognized. (Goal is used here to mean objective, including the pursuit of an endeavor, such as Hull's drive reduction.) At one level the organism pursues goals such as eating and drinking, and at another level the pursuit of a different type of goal explains phenomena such as latent learning and irrelevant-incentive learning. At least one publication has speculated on the possibility of another, less obvious, level of goal pursuit. O'Keefe and Nadel (1978) argued, without detailing the mechanisms whereby it occurs, that rats are driven to create cognitive maps.

The most plausible explanation for learning seems to involve multiple levels of goal pursuit, but if multiple levels of goal direction do exist, then the earlier question can be rephrased to ask: what are the various goal levels associated with learning?

The CEL model (Granger, 1983a, 1983b) proposes three levels of goal pursuit. At the lowest level, innate, latent mechanisms semi-autonomously pursue constant goals which create and manipulate memory structures in such a way that knowledge and, thus, performance, eventually conform to the external environment. The goals at this level are functionally invariant, but the structures that the mechanisms operate on vary dynamically with the goals of the higher, overt level which change in response to the varying needs of the organism. This distinction between functional invariance with dynamic targets at the latent level and dynamic goals at the overt level can be likened to a packing machine in a soda factory. One day the machine may be employed to place orange soda in the cartons. The next day, depending on the (overt) needs of the plant, cherry soda may be packed into cartons with different labels. The machine continues from day to day performing the same function (packing 24 bottles at a time into a carton), but using the bottles and boxes targeted by the higher, overt level of the plant. The overt goals of the plant vary in the same way as the overt goals of the organism change to reflect a dynamic environment and the way the organism must respond to it. If the animal has been starved, the latent mechanisms assist the animal in finding food. If it has been deprived of water, the latent mechanisms still search and match data, but now, instead of matching memory structures that pertain to food, the mechanisms will be concerned with knowledge related to water.

The third level of goal pursuit accounts for still higher level goals such as reasoning, planning, and possibly learning set learning. At this level the organism manipulates low level knowledge with other low level knowledge. This level is called the meta-overt level because the low level knowledge serves a meta role in the acquisition of overt goals.

3.0 Examples of Latent Mechanisms

3.1 Introduction to the Program

CEL-0 is a program embodiment of the CEL (Components of Experiential Learning) model of learning and memory. The program, like the model, consists of twelve parallel and semi-independent primitives which build and manipulate hierarchical memories. These operators collectively perform five classes of data manipulation on memory representations termed episodic schemata. In brief, the twelve operators have the following functions:

Reception Operators
DETECT - sensory events from the various modalities (sight, touch, etc.).
SELECT - ively attend to sensory inputs on the basis of past experience.
NOTICE - input which matches the desirable and undesirable states and trigger COLLECT.
COLLECT - sensory events into packets (episodic schemata) for memory storage.
INDEX - schemata into the memory hierarchy for future reference.

Retrieval Operators
REMIND - the system of past experiences which are related to the present situation.
ACTIVATE - REMINDed schemata based on predictive values of schemata and behavior desired. ACTIVATE triggers reconstruction.

Reconstruction Operators
ENACT - the appropriate efferent actions in the active schema; tune SELECT to attend to predicted afferent events.
SYNTHESIZE - new schemata by matching inputs against predicted events; trigger refinement operators to modify schema based on matches or mismatches.

Refinement Operators
REINFORCE - strengthen successful schemata to reflect their predictive success.
BRANCH - within a schemata to indicate mismatches
DETOUR - within a schemata to indicate branches to be avoided.

In the maze environment these mechanisms combine to enable CEL-0 to demonstrate a latent learning ability enable CEL-0 to demonstrate to demonstrate a latent learning ability similar to rats. A run-time listing of CEL-0's maze performance is available in [McNulty and Granger, 1985].

To understand CEL-0's operation it is convenient to divide its activity into two functionally distinct classes: the acquisition of knowledge and the use of that learned knowledge (i.e. performance). This distinction, however, is completely artificial. CEL-0 is constantly learning, and thus, it is difficult to distinguish between CEL-0's learning and performance, but the distinction is useful for purposes of analysis. A detailed explanation of how CEL-0 performs in the maze is provided in [Granger and McNulty, 1984a, 1984b] and [McNulty and Granger, 1985]. This discussion will focus on the latent roles that some of the operators play and how they interact with the overt level.

3.2 The Role of Latent Mechanisms in the Acquisition of Knowledge

During the exploratory phase of the latent learning experiment CEL-0 is satiated and, thus, possesses no overt goal. When first placed in the maze CEL-0 'knows' nothing of the environment itself. It begins with four innate schemata (move forward, turn left, turn right, and turn around) and knowledge of when those schemata apply. CEL-0's knowledge of the maze is constructed from the sensory input it encounters as it negotiates the maze.

3.2.1 The NOTICE Operator

NOTICE has the innate goal of comparing afferent sensory events with events on the Desirable State List(DSL) and the Undesirable State List(USL). The DSL and USL are dynamic structures which change as CEL-0's overt goals change in response to the ambient environment and innate needs (e.g., hunger, thirst). Any match indicates that CEL-0 has experienced something of note and should store away for possible later use the experiences which lead up to that event.

3.2.2 The COLLECT and INDEX Operators

COLLECT and INDEX are the operators responsible for creating episodic schemata and placing them in memory for later use. COLLECT groups the most recent sensory events as a schema, places the schema in the memory hierarchy, and calls INDEX to create pointers to the new piece of knowledge so that it is recallable. Because the pointers are the only links between present and past, it is critical that the correct

attributes be indexed. INDEX uses the first and last sensory events of the episode as pointers. The former allows CEL-0 to find episodes related to current sensory events. The latter, called the goal index, enables CEL-0 to find schemata associated with overt goals. In the latent learning paradigm two circumstances cause the creation of new memories (see [Granger and McNulty, 1984] for a more complete discussion):

a)SYNTHESIZE detects a successful completion of an ACTIVATEd schema

b)a NOTICE of an event on the DSL or USL

INDEX is vested with an additional goal which plays a very important part in CEL-0's ability to improve its performance during rewarded trials. A COLLECTed schema is composed of seven sensory events representing the most recent CEL-0 experiences. Frequently, only a subset of that group is pertinent to the overt goal which caused the schema to be laid down in memory. The events not pertinent to the goal are extra baggage which CEL-0 may, but probably not, find useful during performance. INDEX helps the refinement of performance by creating many schemata from the one original. Each newly created schema is a sequential subset of the original events CEL-0 experienced. This function is called 'sensitivity analysis' since the net effect is to test the individual sensory events for relevance to the outcome of the schema.

3.3 The Role of Latent Mechanisms in Performance

During initial trials in the maze CEL-0 acquires knowledge of the maze as it 'explores' with no overt goal. Exploration is driven by CEL-0's latent goal level where the innate mechanisms autonomously go about satisfying their goals. In subsequent trials CEL-0 is made hungry and food is placed at one end of the T-maze. During the rewarded trials, CEL-0 must:

a)associate the need (satisfy hunger) with the reward

b)associate the location of the reward with stored schemata

c)ACTIVATE the appropriate schemata

d)Improve performance by refining learned schemata into more appropriate new schemata

3.3.1 The NOTICE Operator

On the first rewarded trial CEL-0 proceeds as on previous trials because it has no knowledge of the reward until it is encountered. Once detected, NOTICE initiates a COLLECT and INDEX which results in episodic schemata indexed by and terminating in 'the sight of food'. If allowed to eat, NOTICE will also COLLECT and INDEX schemata which result in 'consumption of food'. The distinction between these two schemata is important to CEL-0's performance in the irrelevant-incentive learning experiments because schemata resulting in the 'sighting of food' when satiated must be associated with schemata resulting in the 'consumption of food' when hungry.

3.3.2 The REMIND Operator

REMIND's goal is to match afferent sensory events and goals against previously stored indices. A match means that a current sensory event is related to a previous experience.

3.3.3 ACTIVATion of Schemata

REMINDed schemata might be predictions of future experiences. ACTIVATE's job is to choose the best possible predictor (schema) given CEL-0's current goals. ACTIVATE proceeds autonomously applying metrics of comparison and matching schemata results to goal relationships. The goals it matches against are the dynamic goals communicated from the overt goal level.

3.3.4 ENACTment of Efferent Actions

Episodic schemata are composed of two types of events. One type of event results from environmental sensory data which CEL-0 has encountered. The other type of event represents the sensory data CEL-0 has perceived of its own actions. CEL-0 responds to its environment, and as a result perceives new sensory data which represent a new situation in the environment. CEL-0 must now respond to that new situation. Thus, for a schema to be an accurate predictor of an outcome, CEL-0 must usually effect actions comparable to those contained in the stored schema. ENACT's goal is to search the active schema for these efferent acts and initiate them.

3.3.5 The SYNTHESIZE Operator

SYNTHESIZE has responsibility for constructing new schemata from previously stored episodes. It does this in concert with REMIND and ACTIVATE. Recall that new sensory inputs are constantly causing REMINDs of previous experiences and ACTIVATE is selecting schemata which might serve as tools for achieving overt goals. The ENACTed efferent events cause new sensory events and the REMIND/ACTIVATE process continues. As CEL-0's location in the maze changes, ACTIVATE may select a more appropriate schema over the one that is currently active. The new schema will have new efferent events which will cause new sensory stimulation. All of these sensory events, new and old, will be incorporated by SYNTHESIZE into new schemata which will be COLLECTed and INDEXed. In this way, new, refined schemata are constructed from previous experiences.. These new schemata can, in turn, be further refined by the same process during later trials in the maze. [McNulty and Granger, 1985] discusses this in more elaborate detail and shows how these mechanisms work with sensitivity analysis to produce a stepwise refinement of behavior.

3.3.6 REINFORCEment

ACTIVATEd schemata are predictions. Some predictions prove to be successful and some do not. A successful predictor is more useful than an unsuccessful predictor and, as such, deserves special status among CEL-0's stored experiences. When SYNTHESIZE detects the successful completion of an ACTIVATEd schema, REINFORCE is triggered to mark that schema as successful. ACTIVATE will use that strengthening as one of its metrics when selecting the best available predictor.

4.0 Conclusions

Experiments from the psychology literature suggest that some types of learning can not be explained by the unidimensional goal direction many previous learning theorists have employed. Two such types of learning are latent learning and irrelevant-incentive learning. A multidimensional system involving three levels of goal direction seems a more reasonable explanation of learning. The lowest level is the latent level where semi-autonomous mechanisms operate to map the external environment onto the internal data structures such that the organism's behavior conforms to the external surroundings. The goals at this level are implicit, innate and, thus, unchanging. At a higher level, and working in concert with the latent mechanisms, is the overt level where goals are dynamic, reflecting the changing desires and changing environment the organism finds itself in. At the highest level, existing knowledge structures (i.e. episodic schemata) are used as meta-schemata to pursue overt goals. Typical functions at the meta-overt level are: reasoning about prior experience (such as deduction and induction), planning, understanding, selection of schemata for activation, and 3-D rotation of spatial knowledge. These three levels working in concert enable simple structures to exhibit complex behavior.

REFERENCES

[1] Blodgett, H.C., "The Effect of the Introduction of Reward Upon the Maze Performance of Rats", Univ. Calif. Psychol., 4:113-34.

[2] Bower, G.H. and Hilgard E.R., Theories of Learning, Englewood Cliffs, Prentice-Hall, 1981.

[3] Granger, R.H., "Identification of Components of Episodic Learning: The CEL Process Model of Early Learning and Memory", Journal of Cognition and Brain Theory, 6,1, pp.5-38, February, 1983a.

[4] Granger, R.H., "An Artificial-Intelligence Model of Learning and Memory that Provides a Theoretical Framework for the Interpretation of Experimental Data in Psychology and Neurobiology", Department of Computer Science Technical Report 210, University of California, Irvine, September, 1983b.

[5] Granger, R.H. and McNulty, D.M., "The CEL-0 system: Experience with a computer model that learns to run a maze", Department of Computer Science Technical Report 220, University of California, Irvine, March, 1984a.

[6] Granger, R.H. and McNulty, D.M., "Learning and Memory in Machines and Animals: An AI Model That Accounts for Some Neurobiological Data", in Proceedings of the Sixth Annual Conference of the Cognitive Science Society, pp. 161, 1984b. (also in: Memory , Experience, and Reasoning, Kolodner, J.L. and Riesbeck, C.K. (eds.), Erlbaum and Associates, Hillsdale, NJ, to appear 1985).

[7] McNulty, D.M. and Granger, R.H., "Elements of Latent Learning in a Maze Environment", Information and Computer Science Technical Report #85-03, University of California, Irvine, 1985.

[8] Mackintosh, N.J., The Psychology of Animal Learning, Academic Press, London, New York, and San Francisco, 1974.

[9] O'Keefe, J. and Nadel, L., The Hippocampus as a Cognitive Map, Clarendon Press, Oxford, 1978.

[10] Simmons, R., "The Relative Effectiveness of Certain Incentives in Animal Learning", Comp. Psychol. Monogr., 1924, 2, 1-79.

[11] Tolman, E.C. and Honzik, C.H., "Introduction and Removal of Reward, and Maze Performance in Rats", Univ. Calif. Publ. Psychol., 4:257-75.

CONNECTIONISTIC LEARNING IN REAL TIME: SUTTON-BARTO ADAPTIVE ELEMENT AND CLASSICAL CONDITIONING OF THE NICTITATING MEMBRANE RESPONSE

J.W. MOORE, J.E. DESMOND, N.E. BERTHIER, D.E.J. BLAZIS, R.S. SUTTON, AND A.G. BARTO

Departments of Psychology and Computer and Information Science, University of Massachusetts-Amherst

Sutton and Barto's model of connectionistic learning by a neuron-like adaptive element predicts many aspects of classical conditioning, including acquisition, extinction, interstimulus interval effects in trace conditioning, overshadowing, Kamin blocking, and conditioned inhibition (Barto & Sutton, 1982; Sutton & Barto, 1981). The equations governing the learning process, i.e., the moment to moment changes in the "synaptic weights" associated with CSs, depend on two memory processes. The first is a local decaying trace representing the after-effect of input to the element from a CS. The second is the trace or memory of the element's output during the preceding computational epoch. The element's output or response is simply the weighted sum of its inputs, and the equation for modifying weights is basically equivalent to the Rescorla-Wagner model of associative learning. The output trace is interpreted as the element's prediction of its behavior during the current computational epoch.

This report outlines an extension of the Sutton-Barto model to the classically conditioned nictitating membrane (NM) response of the rabbit, a widely adopted "model system" for theoretical and neurobiological studies of learning and memory. The rabbit NM and related eye blink response offer an extensive experimental literature for assessing the performance of the model (Gormezano, Kehoe, & Marshall, 1983). Our approach might be applied to other instances of classical conditioning, but the mathematical details, such as constraints on functions, constants, and parameters, would no doubt vary from one case to the next.

Our studies of the properties of the Sutton-Barto model are concerned with the development of variants of the basic model capable of generating the form or topography of conditioned responses (CRs) as they unfold in real time within trials. Generating CR topography necessitates the imposition of constraints on the variables of the model. Given constraints that model CR topography for comparatively simple protocols, such as acquisition with a single CS in a forward delay paradigm, the question becomes whether the same constraints hold for more complex protocols, such as serial compounds conditioning and conditioned inhibition, in which two CSs are involved. In short, how valid are the model's predictions over a range of training paradigms?

To appreciate why efforts to model CR topography are interesting from the perspective of AI and neuroscience, consider the following rationale: One begins with the

hypothesis that a single adaptive element, which is basically a linear device with multiple input lines and a single graded output, is capable of modeling all of the characteristics of the system under consideration, in this case the NM CR. The next step is to select specific mathematical expressions relating variables and parameters of the model to each other and to time. This step has many earmarks of curve fitting but with the important difference that selected functions must adhere to the broader constraints of the model. The third step is to determine the extent to which the detailed model can describe the expermental literature surrounding the system under scrutiny and make novel predictions. Failures of the model guide subsequent theoretical development. In the domain of connectionistic learning of interest to the authors, these theoretical developments would entail networks of adaptive elements. We believe the problems associated with the development of a comprehensive theoretical rendering of the NM CR may point the way toward a better understanding of how connectionistic learning might be linked to problems of adaptive sensory-motor control encountered by living organisms and intelligent machines.

Space limitations preclude a description of the mathematical details of the Sutton-Barto model that produced the simulations presented in this paper. The constraints imposed on the model were designed to reflect the activity of single neurons of the brain stem with firing patterns related to the NM CR in a forward delay conditioning paradigm and with an interstimulus interval of 350 msec between the onset of the CS and the onset of the UCS. The following are some of the considerations that shaped the contraints imposed on the Sutton-Barto model:

1. CR related neurons in this paradigm rarely have baseline firing rates lower than 10-Hz. Their maximum firing rate rarely exceeds 130-Hz.

2. Following the onset of a CS, spikes are recruited at a slow rate. About 150 msec after CS onset, spike recruitment increases sharply and continues to increase in a negatively accelerated fashion throughout the remainder of the interstimulus interval.

3. Onset of the UCS results in a rapid recruitment of spikes to a rate near the maximum of 130-Hz. This high firing rate persists until UCS offset, after which firing initiated by the UCS declines geometrically toward baseline. The contribution of the UCS to the neuron's firing diminishes with CR acquisition because the conditioning process anticipates the UCS and therefore produces firing rates near the maximum. For this reason, post-UCS firing tends to decline with learning, an observation that might be related to "conditioned diminution of the UCR".

4. Other constraints concerned the changes in the topography of the NM CR over the course of acquisition and extinction trials: (a) CR latency decreases progressively during acquisition and increases during extinction. (b) CR amplitude increases progressively during acquisition and

decreases during extinction. Nevertheless, peak CR amplitude tends to coincide with the onset of the UCS.

5. The final constraint concerns the effect of interstimulus interval on the efficacy of conditioning. The model was required to yield progressively poorer conditioning under a delay conditioning paradigm with interstimulus intervals greater than the recognised optimal interval for the preparation of 250-350 msec. The unconstrained Sutton-Barto model yields an appropriate interstimulus interval function for trace conditioning but not for delay conditioning.

Modifications of the Sutton-Barto model that satisfy the above constraints were arrived at largely by trial and error. The fully implemented model successfully described CR topography and its changes with training. It also lends itself to simulation of firing patterns of single neurons with activity related to the CR.

Figure 1A shows simulated NM CR/UCRs (vertical axis) as a function of acquisition trials (oblique axis). Notice that the latency of the CR decreases over training. Peak CR amplitude increases over trials, yet remains just before the UCR. This process is reversed during simulated extinction trials (Fig. 1B).

Figure 2 summarizes a simulation of conditioned inhibition training. CS1 is reinforced with the UCS, and a compound consisting of CS1 and CS2 is not reinforced. The two trial types, designated CS+ and CS-, respectively, were alternated in simulated training. Figure 2A shows simulated CR topographies for the two trial types at the end of training. Figure 2B shows changes of "synaptic weights" over the course of training. Notice that the weight for CS2 becomes increasingly negative in value. Figures 2C and 2D show simulated cumulative peristimulus-time histograms of neuronal firing for the two trial types.

Simulation experiments also indicate that this variant of the Sutton-Barto model does a credible job of simulating serial-compound conditioning, Kamin blocking and overshadowing, and higher-order conditioning. In addition, simulations have revealed a variety of subtle and largely untested effects on conditioned responding associated with the within-trial timing of onsets and offsets of CSs in serial compound paradigms.

Whether the Sutton-Barto model can be constrained to model other varieties of behavioral conditioning with the same success that it has in the case of the NM CR remains an open question. Also unresolved is whether the structure of the model, and the particular constraints imposed for the NM CR, truly have implications for understanding physiological meachanisms of learning and memory. We believe this to be a distinct possibility.

ACKNOWLEDGEMENT

This research was supported by grants AFOSR 830215, NSF BNS 8317920, and USPHS 1 F31 MH08951

REFERENCES

Barto, A.G., & Sutton, R.S. (1982) Simulation of anticipatory reponses in classical conditioning by a neuron-like adaptive element. Behavioural Brain Research , 4, 221-235.

Gormezano, I., Kehoe, E.J., & Marshall, B.S. (1983) Twenty years of classical conditioning with the rabbit. Sprague, J.M. & Epstein, A.N. (Eds.) Progress in Psychobiology and Physiological Psychology , 10, 197-275.

Sutton, R.S., & Barto, A.G. (1981) Toward a modern theory of adaptive networks: Expectation and anticipation. Psychological Review , 88, 135-170.

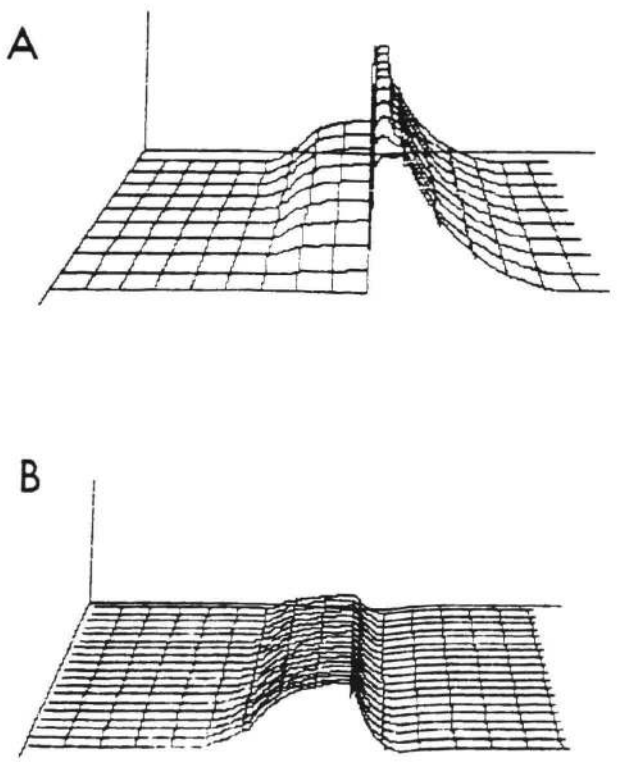


FIGURE 1

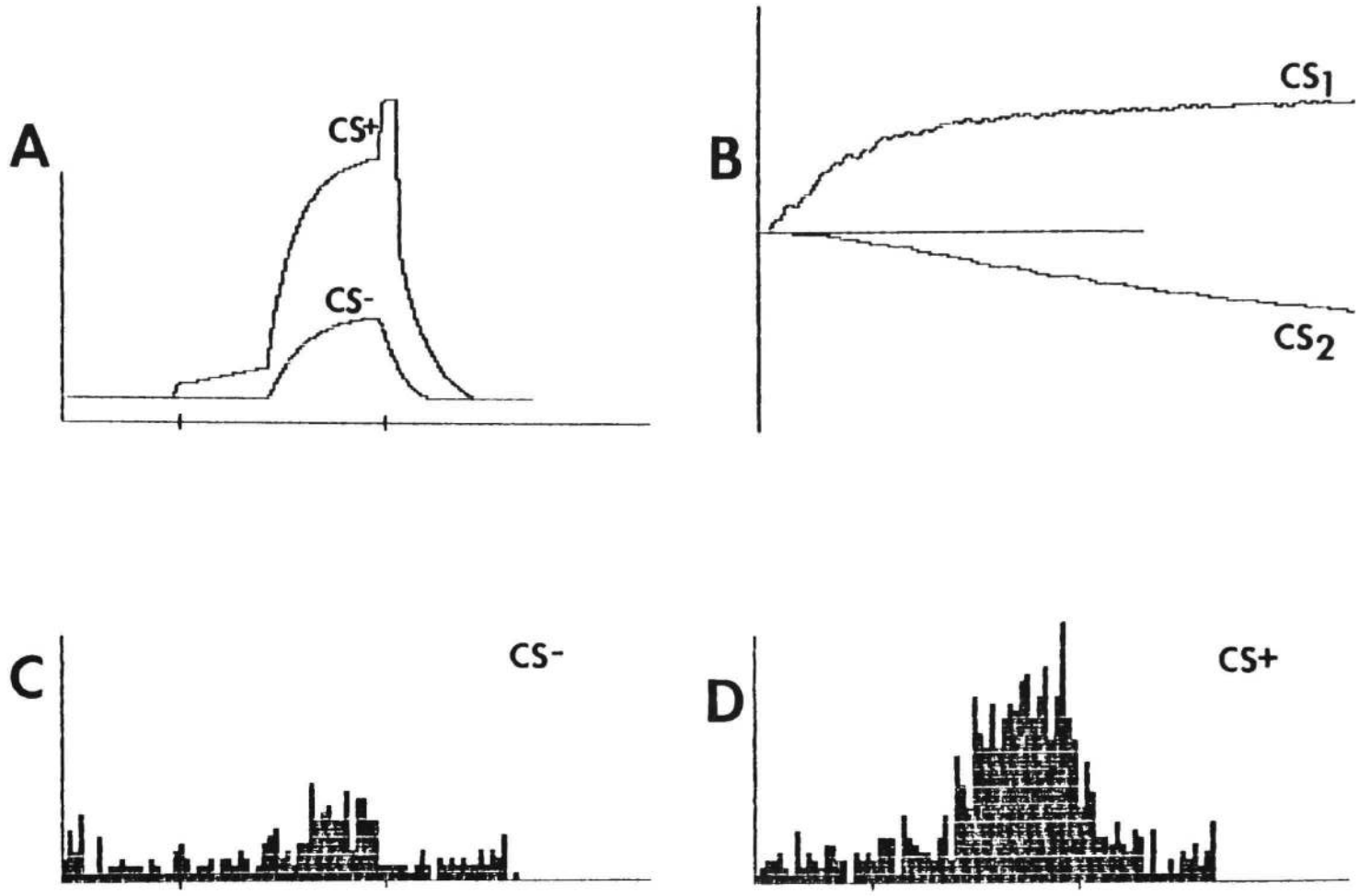


FIGURE 2

Explanation and Generalization Based Memory

Michael J. Pazzani
UCLA Artificial Intelligence Laboratory
3731 Boelter Hall
Los Angeles, CA 90024
and
The Aerospace Corporation
P.O. Box 92957
Los Angeles, CA 90009

Abstract

A model of memory and learning is presented which indexes a new event by those features which are relevant in explaining why the event occurred. As events are added to memory, generalizations are created which describe and explain similarities and differences between events. The memory is organized so that when an event is added, events with similar features are noticed. An explanation process attempts to explain the similar features. If an explanation is found, a generalized event is created to organize the similar events and the explanation is stored with the generalized event.

Introduction

The goal of this research is to identify the role of explanation in a generalization based memory. A computer program, OCCAM, has been implemented which learns about variations of kidnapping. The program starts out with general knowledge about coercion represented as a meta-MOP [7]. After some examples, it creates a MOP which describes a kind of kidnapping (along with the explanation that a family member of the victim's family pays the ransom to achieve the goal of preserving the victim's health). Further examples create a specialization of this MOP which represent an inherent flaw in kidnapping: that the victim can testify against the kidnapper, since the kidnapper can be seen by the victim. This specialization is stored as a sub-MOP of the kidnapping MOP and is indexed by the kidnapper's goal failure: going to jail. After some more examples, a similarity is noticed about the kidnapping of infants. This coincidence starts an explanation process which explains the choice of victim to avoid a possible goal failure, since infants cannot testify.

There are a couple of interesting features of this type of learning:

- The explanation process eliminates the problem of including unrelated coincidences in generalized events. For example, all of the infants kidnapped in the events presented to OCCAM have blond hair. This feature is not used in the explanation, so it is not included in the generalized event.

- There is causal and motivational information associated with generalized events. This information states why various features are included in the generalized event.

- The explanation process can make use of the generalized events in memory. Explanation consists of a rule based component similar to PAM [9] and a memory based explanation component. The rules state such things as that if someone says they are going to hurt a family member, this motivates a goal of preserving the health of the family member. There are no special rules about kidnapping. Therefore, it is not capable of explaining the kidnapping of infants until it has built a generalized event about the victim testifying against the kidnapper. The explanation process uses intentional links [2] to specify the relationships between goals, plans and events.

Related Work

Much early work on learning (e.g., [8], and [3]), centered on the acquisition of a concept from a number of examples. A characteristic description of a class of objects was built by inductive means by considering positive (and, in some instances, negative) examples. The work reported here differs from this work on concept acquisition in a number of ways. First, these programs were "told" what concepts to learn and examples were identified as positive or negative instances. In contrast, OCCAM is not told what to learn. Instead, OCCAM incrementally learns new concepts from examples as a natural consequence of organizing memory around similarities. Secondly, the generalized events built by OCCAM do not contain all features common to the examples. Its explanation process distinguishes between relevant and coincidental features.

DeJong presents a model of explanation based learning [1] which learns schemata from a single example. His program constructs an explanation of relationships between various components of an event by a knowledge-intensive understanding process similar to PAM. The explanation and the event are then generalized by retaining only those parts used in the explanation. Our work differs from DeJong's in a number of aspects. First, OCCAM learns incrementally. It is difficult to imagine a system learning the specialized motivation for kidnapping infants from the first example of a kidnapping, since the explanation process can find an explanation for kidnapping any person. In OCCAM, after the basic kidnapping schema (or MOP) is learned, later examples focus OCCAM on explaining coincidences about the age of the victims. Additionally, the explanation process makes use of other events or generalized events.

In IPP [5] and UNIMEM [6], Lebowitz is concerned with making "factual" generalizations. Unlike OCCAM, these programs make no attempt to perform a causal or explanatory analysis. Therefore, no distinction is made between relevant or coincidental features. After a number of diverse examples, IPP and UNIMEM can correct generalizations to remove coincidences which are contradicted.

CYRUS [4] is a program which organizes and searches a model of episodic memory. Like IPP, it does not produce an explanation of its generalizations. It avoids the problem of indexing on coincidentally similar features by an a priori set of relevant features.

Learning and Memory in OCCAM

OCCAM makes a distinction between two types of generalized events. *Explanatory generalized events* are MOPs created as a specialization of a more general MOP. Associated with each explanatory generalized event are new causal and goal relationships. For example, the kidnapping of infants is an explanatory generalized event which includes the special motivation for selecting the hostage. *Organizational generalized events* are also created as specialization of more general MOPs. However, they add no additional explanatory information. They correspond to the factual generalizations of IPP and serve mainly to organize the memory. An example organizational generalized event would be kidnappings where the hostages grandmother paid the ransom. (Unless, of course, some explanation could be found.)

There are two parts to the incremental learning algorithm used by OCCAM. The first step is to find the appropriate place in memory to index a new event. The memory is organized so that a new event will be added to memory in the same place as similar events. The second step is to attempt to create a generalization.

After the most specific applicable MOP is found, similar events are found by using the features of the new event as indices. Next, generalization is attempted by a number of *generalization rules* which postulate causal or intentional relationships. For example, one rule states *If an action always precedes a state, postulate the action causes the state.* Other rules which postulate goal relationships will be discussed in the next section. An explanation process is then used to verify the postulated causal or intentional relationships. This explanation process marks all features necessary for establishing the relationships. The explanation process used here is more focused than that used in DeJong's work. Rather than asking general questions such as "Why did this happen?", more specific questions are used such as "Was there an action which motivates a goal before this action which this action achieves." A

new MOP may be created depending on the result of the generalization:

- If the explanation is successful, than an explanatory generalized event is built and indexed under the most specialized MOP by the new features establishing the explanation. The new event and any similar events are organized under this new generalization, indexed by the features not used in the generalized event.

- If the explanation process is unsuccessful, and the most specific MOP is an explanatory generalized event, a default rule is used to attempt to form an organizational generalized event. This notes that there appears to be a coincidental relationship but does store any justification.

An Example: Learning about Kidnapping

The meta-MOP for coercion involves a PREParation, a THREAT, a DEMAND, and several RESULT scenes. Figure 1 illustrates an example of kidnapping which is a kind of coercion. In this Figure, the notation "the(FEATURE)" indicates the actual value of the feature is the same as the value of that feature. Coercion usually involves at least three roles: an ACTOR, who performs the PREParation, and says he will carry out the THREAT unless his DEMAND is met; an OBJECT which is the object of the PREParation and the THREAT (i.e., in kidnapping the hostage is the OBJECT); and the VICTIM which receives the THREAT, and usually performs one of the RESULTs. (The VICTIM in kidnapping is not the hostage but the person who pays the ransom.) The coercion meta-MOP is intended to be very general and account for many situations from kidnappings to playground arguments (e.g., "If you don't let me pitch, I'm gonna take my ball and go home").

```
-------------------------------------------------------------
K1: COERCION
ACTOR human  NAME Joe K.  HEIGHT tall AGE 30s HAIR brown
OBJECT human  NAME John V.  HEIGHT short AGE teens HAIR blond
              RELATION family TYPE son
                              OF the(VICTIM)
VICTIM human  NAME Dad V.  HEIGHT tall AGE 40s HAIR blond
              RELATION family TYPE father
                              OF the(OBJECT)
PREP atrans ACTOR the(ACTOR) TO the(ACTOR) OBJECT the(OBJECT)
DEMAND poss-by ACTOR the(ACTOR)
               OBJECT money AMOUNT 50000
THREAT health OF the(OBJECT) VAL -10
RESULT atrans  ACTOR the(ACTOR) FROM the(ACTOR)
               TO the(VICTIM) OBJECT the(OBJECT)
RESULT atrans ACTOR the(VICTIM)
              FROM the(VICTIM)
              TO the(ACTOR)
              OBJECT money AMOUNT 50000
RESULT $trial SENTENCE 15 VERDICT guilty
              WITNESS the(OBJECT) CRIMINAL the(ACTOR)
-------------------------------------------------------------
```

Figure 1: An Example of Coercion: A Kidnapping

The initial state of the memory of OCCAM contains only the coercion meta-MOP (mm-COERCE). K1, the example in Figure 1, is then added to memory. It is indexed under mm-COERCE by all of its features (i.e., its scenes and roles). The next example, K2, is similar to K1, except the AGE of the OBJECT is an infant, some minor difference in the features of the VICTIM and the ACTOR, and there is

```
Looking for similar events under mm-COERCE... found (K1).
Similarities:
COERCION
ACTOR human  HEIGHT tall AGE 30s HAIR brown
OBJECT human RELATION family TYPE son
                              OF the(VICTIM)
VICTIM human RELATION family TYPE father
                              OF the(OBJECT)
PREP atrans ACTOR the(ACTOR) TO the(ACTOR) OBJECT the(OBJECT)
DEMAND poss-by ACTOR the(ACTOR) OBJECT money
THREAT health OF the(OBJ) VAL -10
RESULT atrans ACTOR the(ACTOR) FROM the(ACTOR)
              TO the(VICTIM) OBJECT the(OBJECT)
RESULT atrans ACTOR the(VICTIM) FROM the(VICTIM)
              TO the(ACTOR) OBJECT money
-----------------------------------------------------------------
```

Figure 2: Noticing the Similarities between two kidnappings

no trial in which the ACTOR goes to jail. Figures 2 and 3 are an edited transcript of the creation of a MOP which describes the kidnapping of a family member for a monetary ransom. The similarities between K1 and K2 are noted (see Figure 2). Then, a rule, GENERALIZE-RESULTS, is used to postulate an explanation for this similarity. This rule states *Look for an action before the RESULT which motivates a goal which the RESULT achieves or an action before the RESULT which is part of plan which the RESULT realizes.* In this example, a goal of preserving the health of the OBJECT by the VICTIM is inferred and paying the ransom achieves this goal. Additionally, the ACTOR is performing the plan of keeping a bargain when he gives the OBJECT back. In general, a better explanation utilizes the goals rather than the plans. However, in this case, it's not possible to infer why the kidnapper releases the hostage. This new MOP (MOP.327) is indexed under mm-COERCE by the relevant features, and the inferred goal as shown in Figure 4. Notice that some features (e.g., the AGE, and HEIGHT of the kidnapper) are not included in the generalization even though they are common to both kidnapping examples because they are not used in the explanation.

```
-----------------------------------------------------------------
Running generalization rule GENERALIZE-RESULTS.
Inferring RESULT REALIZES PLAN (KEEP-BARGAIN)
Inferring RESULT ACHIEVES GOAL (P-HEALTH)
Making sub-mop MOP.327 {kidnap} of mm-COERCE from (K2 K1)
Used in explanation:
COERCION
ACTOR human
OBJECT human
VICTIM human RELATION family OF the(OBJECT)
DEMAND poss-by ACTOR the(ACTOR) OBJECT money
THREAT health OF the(OBJECT) VAL -10
RESULT atrans ACTOR the(ACTOR) FROM the(ACTOR)
              TO the(VICTIM) OBJECT the(OBJECT)
RESULT atrans ACTOR the(VICTIM) FROM the(VICTIM)
              TO the(ACTOR) OBJECT money
-----------------------------------------------------------------
```

Figure 3: Forming an Explanatory Generalized Event

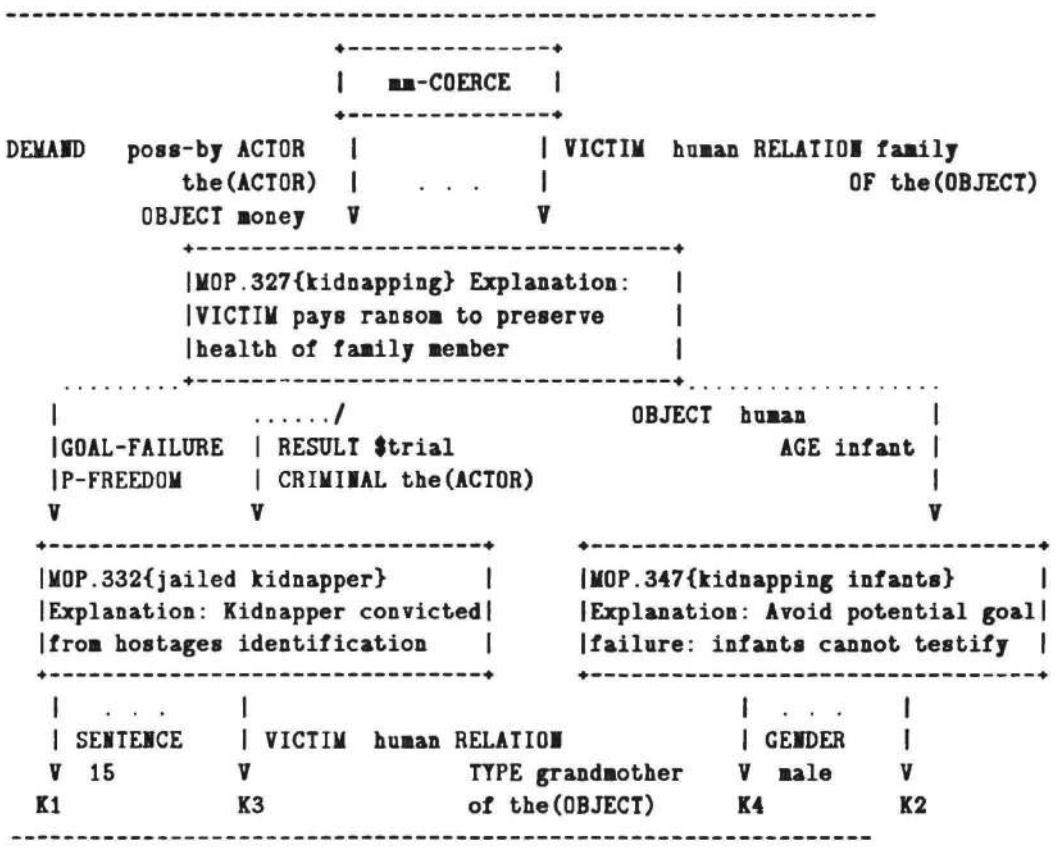


Figure 4: Memory after creating 3 specializations of mm-COERCE

The next event added to memory is K3, which is similar to K1 in that the kidnapper goes to jail after the hostage testifies. There are minor differences in the features of the participants. MOP.327, the kidnapping MOP, is the most specific MOP which is not contradicted by K3. It has an additional RESULT which is similar to a result of K1. A rule which states *If there is a RESULT which thwarts a goal, look for an action before the RESULT which enables the RESULT* finds an inherent flaw in kidnapping: the hostage sees the kidnapper when he is abducted and can testify against the kidnapper. A new MOP, MOP.332 (jailed-kidnapper) is created an indexed under MOP.327 (kidnap) by the indices of the RESULT, the goal failure, and the PREParation which enables the goal RESULT which thwarts the goal as shown in Figure 4. K1 and K3 are indexed under this new MOP, while K2 remains indexed under the kidnapping MOP.

K4, another kidnapping of a blond infant in which the kidnapper was not caught, is added to memory next. MOP.327 (kidnap) is found to be the most specific MOP which describes K4. A similarity is noticed between K4 and K2, the OBJECTs are both blond infants. An applicable generalization rule states *If the PREParation is performed on the an object, look for other MOPs which have a goal failure. Check if the PREParation avoids the goal failure, if it does postulate the ACTOR performed the PREParation to avoid the goal failure.* In this example, the goal of preserving freedom of the kidnapper cannot be thwarted by the infant testifying. A new MOP is created indexed by the AGE of the OBJECT (and not the hair color) as shown in Figure 4.

These examples illustrate the process of creating an explanatory generalized event. In a more realistic set of examples, several organizational generalized events would also be created and each MOP would index a greater number of events and sub-MOPs. The point of creating explanatory generalizations is to create specialized explanations for situations. With only mm-COERCE in memory, the explanation of kidnapping an infant would be "The ACTOR wants the VICTIM to do something" After MOP.327

(kidnap) is created, the explanation would be "The ACTOR wants a member of the OBJECT's family to give him money" After MOP.347 (kidnapping infants) is created the explanation would be "The ACTOR wants a member of the OBJECT's family to give him money and the ACTOR wants to avoid being convicted, so he's kidnapping an infant since infants can't testify"

Conclusion

OCCAM is a program which organizes memories of events and learns by creating explanatory generalized events. It addresses the issue of deciding which features are relevant in producing a generalization. It answers this question by proposing the relevant features are those which are essential in explaining why the event occurred (e.g., why a goal fails). The features which are not essential to arriving at an explanation are exactly those features which would be expected to vary in future events. The unessential features are not used as indices by OCCAM since they are not useful in understanding future events. OCCAM can learn more quickly and accurately than many previous systems since it relies on an explanation process to eliminate unessential features rather than correlation over a large number of examples. Indexing by relevant features has some implications for expert systems which operate by recalling similar experiences. Should a medical expert system index a case by the patient's weight, height, clothing or jewelry? The answer proposed here is to use these as indices in explanatory generalized events only if they are essential in establishing a pathological explanation. Organizational generalized events describe those situations where a coincidence is noted but there is no explanation. These coincidences might initiate and focus the search for new pathological knowledge.

Acknowledgements

Discussions with Mike Dyer and Margot Flowers helped in the evolution of the ideas in this paper. Communications with Michael Lebowitz were also fruitful. This research was supported in part by a grant from the Keck Foundation.

References

1. DeJong, G. Acquiring Schemata Through Understanding and Generalizing Plans. Proceedings of the Eighth International Joint Conference on Artificial Intelligence, Karlsruhe, West Germany, 1983.

2. Dyer, M.. *In Depth Understanding.* MIT Press, 1983.

3. Hayes-Roth, F. and McDermott, J. Knowledge Acquisition from Structural Descriptions. Proceedings of the Fifth International Joint Conference on Artificial Intelligence, Cambridge, Mass., 1977.

4. Kolodner, J. *Retrieval and Organizational Strategies in Conceptual Memory: A Computer Model.* Lawrence Erlbaum Associates, Hillsdale, NJ., 1984.

5. Lebowitz, M. Generalization and Memory in an Integrated Understanding System. Computer Science Research Report 186, Yale University, 1980.

6. Lebowitz, M. "Correcting Erroneous Generalizations". *Cognition and Brain Theory 5*, 4 (1982).

7. Schank, R. *Dynamic Memory: A Theory of Reminding and Learning in Computers and People.* Cambridge University Press, 1982.

8. Vere, S. Induction of Concepts in the Predicate Calculus. Proceedings of the Fourth International Joint Conference on Artificial Intelligence, Tbilisi, USSR, 1975.

9. Wilensky, R. Understanding Goal Based Stories. Computer Science Research Report 140, Yale University, 1978.

BAYESIAN NETWORKS: A MODEL OF SELF-ACTIVATED MEMORY FOR EVIDENTIAL REASONING*

Judea Pearl
Cognitive Systems Laboratory, Computer Science Department, UCLA

ABSTRACT

The paper reports recent results from the theory of Bayesian networks, which offer a viable formalism for realizing the computational objectives of connectionist models of knowledge. In particular, we show that the Bayesian network formalism is supportive of self-activated, multidirectional propagation of evidence that converges rapidly to a globally-consistent equilibrium.

1. INTRODUCTION

This study was motivated by attempts to devise a computational model for humans' inferential reasoning, namely, the mechanism by which people integrate data from various sources and generate a coherent interpretation of that data. Since the knowledge from which inferences are drawn is mostly judgmental--namely, subjective, uncertain, and incomplete--a natural place to start would be to cast the reasoning process in the framework of probability theory. Probability theory is also useful because it is the simplist calculus which permits inferences to flow two ways: from hypothesis to evidence (predictive), as well as from evidence to hypothesis (diagnostic). Unfortunately, traditional probability theory has erected cultural barriers against its usage in modelling human cognition. Scholarly textbooks on probability theory try hard to create the impression that to construct an adequate representation of probabilistic knowledge we must first define a ***joint distribution function*** on all propositions and their combinations, and that this function should serve as the basis for all inferred judgements-- a rather distorted picture of human reasoning.

Human judgments regarding a small number of propositions (such as the likelihood that a patient suffering from a given disease will develop a certain type of complication) are issued swiftly and reliably, while judging the likelihood of a conjunction of many propositions is done with great degree of difficulty and hesitancy. This suggests that the elementary building blocks which make up human knowledge are not entries of a giant joint-distribution table, but rather low-order probabilistic relations between small clusters of semantically-related propositions.

Additionally, a person reluctant to giving a numerical estimate for the conditional probability $P(A|B)$, will normally show no hesitation to state whether propositions A and B are dependent or independent,given C, namely, whether knowing the truth of B will or will not alter the belief in A, assuming that C is true. Evidently, the notion of conditional dependence is more basic than the numerical values attached to probability judgments, contrary to the picture painted in most textbooks on probability theory, where the latter is presumed to provide the criterion for testing the former. This suggests that the fundamental structure of human judgmental knowledge can be represented by ***dependency graphs*** and that mental tracing of links in these graphs are responsible for the basic steps in querying and updating that knowledge. Bayesian networks offer an effective formalism for these graph operations.

*This work was supported in part by the National Science Foundation, Grant #DSR 83-13875

2. BAYESIAN NETWORKS

Bayes Networks are directed acyclic graphs in which the nodes represent propositions (or variables), the arcs signify the existence of direct causal influences between the linked propositions, and the strengths of these influences are quantified by conditional probabilities (Figure 1).

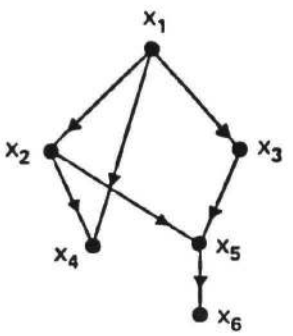


Figure 1

Thus, if the graph contains the variables $x_1, \ldots, x_n$, and S_i is the set of parents for variable x_i, then a complete and consistent quantification can be attained by specifying, for each node x_i, an assessment $P'(x_i \mid S_i)$ of $P(x_i \mid S_i)$. The product of all these assessments,

$$P(x_1, \ldots, x_n) = \prod_i P'(x_i|S_i) \tag{1}$$

constitutes a joint-probability model which supports the assessed quantities. That is, if we compute the conditional probabilities $P(x_i \mid S_i)$ dictated by $P(x_1, \ldots, x_n)$, the original assessments are recovered. Thus, for example, the distribution corresponding to the graph of Figure 1 can be written by inspection:

$$P(x_1,x_2,x_3,x_4,x_5,x_6) = P(x_6|x_5)\,P(x_5|x_2,x_3)\,P(x_4|x_1,x_2)\,P(x_3|x_1)\,P(x_2|x_1)\,P(x_1).$$

An important feature of Bayes network is that it provides a clear graphical representation for many independence relationships embedded in the underlying probabilistic model. The criterion for detecting these independencies is based on ***graph separation***: namely, if all paths between x_i and x_j are "blocked" by a a subset S of variables, then x_i is independent of x_j given the values of the variables in S. Thus, each variable x_i is independent of both its siblings and its grandparents, given the values of the variables in its parent set S_i. For this "blocking" criterion to hold in general, we must provide a special interpretation of separation for nodes that share common children. We say that the pathway along arrows meeting head-to-head at node x_k is normally "blocked", unless x_k or any of its descendants is in S. In Figure 1, for example, x_2 and x_3 are independent given $S_1 = \{x_1\}$ or $S_2 = \{x_1,x_4\}$, because the two paths between x_2 and x_3 are blocked by either one of these sets. However, x_2 and x_3 may not be independent given $S_3 = \{x_1,x_6\}$, because x_6, as a descendant of x_5 , "unblocks" the head-to-head connection at x_5, thus opening a pathway between x_2 and x_3.

3. AUTONOMOUS PROPAGATION AS A COMPUTATIONAL PARADIGM

Once Bayesian network is constructed, it can be used to represent the generic causal knowledge of a given domain, and can be consulted to reason about the interpretation of specific input data. The interpretation process involves instantiating a set of variables corresponding to the input data and calculating its impact on the probabilities of a set of variables designated as hypotheses. In principle, this process can be executed by an external interpreter who may have access to all parts of the network, may use its own computational facilities, and may schedule its computational steps so as to take full advantage of the network topology with respect to the incoming data. However, the use of such an interpreter seems foreign to the reasoning process normally exhibited by humans [Shastri and Feldman, 1984]. Our limited short-term memory and narrow focus of attention, combined with our inflexibility of shifting rapidly between alternative lines of reasoning seem to suggest that our reasoning process is fairly local, progressing incrementally along prescribed pathways. Moreover, the speed and ease with which we perform some of the low level interpretive functions, such as recognizing scenes, comprehending text, and even understanding stories, strongly suggest that these processes involve a significant amount of parallelism, and that most of the processing is done at the ***knowledge level itself***, not external to it.

A paradigm for modeling such active knowledge base would be to view a Bayesian network not merely as a passive parsimonious code for storing factual knowledge but also as a computational architecture for reasoning about that knowledge. That means that the links in the network should be treated as the only pathways and activation centers that direct and propel the flow of data in the process of querying and updating beliefs. Accordingly, we assume that each node in the network is designated a separate processor which both maintains the parameters of belief for the host variable and manages the communication links to and from the set of neighboring, logically related, variables. The communication lines are assumed to be open at all times, i.e., each processor may at any time interrogate the belief parameters associated with its neighbors and compare them to its own parameters. If the compared quantities satisfy some local constraints, no activity takes place. However, if any of these constraints is violated, the responsible node is activated to revise its violating parameter and set it straight. This, of course, will activate similar revisions at the neighboring nodes and will set up a multidirectional propagation process, until equilibrium is reached.

While constraint-propagation mechanisms have found several applications in AI, such as vision [Rosenfeld, Hummel and Zucker, 1976; Waltz, 1972] and truth maintenance [McAllester, 1980], their use in evidential reasoning has been limited to non-Bayesian formalisms [e.g. Lowrance, 1982, Shastri and Feldman, 1984]. The reason has been several-fold.

First,the conditional probabilities characterizing the links in the network do not seem to impose definitive constraints on the probabilities that can be assigned to the nodes. The quantifier $P(A|B)$ only restricts the belief accorded to A in a very special set of circumstances: namely, when B is known to be true with absolute certainty, and when no other evidential data is available. Under normal circumstances, all internal nodes in the network will be subject to some uncertainty and, more seriously, after observing evidence e the conditional belief in A is no longer governed by $P(A|B)$ but by $P(A|B, e)$, which may be totally different. The result is that any assignment of beliefs, $P(A)$ and $P(B)$, to propositions A and B can be consistent with the value of $P(A|B)$ initially assigned to the link connecting them; therefore, no violation of constraint can be detected locally.

Next, the difference between $P(A|B, e)$ and $P(A|B)$ seems to suggest that the weights on the links should not remain fixed but should undergo constant adjustment as new evidence ar-

rives. This, in turn, would require an enormous computational work and would wipe out the advantages normally associated with propagation through fixed constraints.

Finally, the fact that evidential reasoning involves both top-down (predictive) and bottom-up (diagnostic) inferences has caused apprehensions that, once we allow the propagation process to run its course unsupervised, pathological cases of instability, deadlock, and circular reasoning will develop [Lowrance, 1982]. Indeed, if a stronger belief in a given hypothesis means a greater expectation for the occurrence of its various manifestations and if, in turn, a greater certainty in the occurrence of these manifestations adds further credence to the hypothesis, how can one avoid infinite updating loops when the processors responsible for these propositions begin to communicate with one another?

This paper reports that coherent and stable probabilistic reasoning ***can*** be accomplished by local propagation mechanisms while keeping the weights on the links constant throughout the process. This is made possible by characterizing the belief in each proposition by a ***vector*** of several parameters, each representing the degree of support that the host proposition obtains from one of its neighbors. Maintaining such a breakdown record of the sources of belief is also postulated as the mechanism which permits people to trace back reasoned assumptions for the purposes of modifying the model and generating explanatory arguments.

4. PROPAGATION IN SINGLY-CONNECTED NETWORKS

The problems associated with asynchronous propagation of beliefs, can be solved completely if the network is singly connected, namely, if there is one underlying path between any pair of nodes. These include trees, where each node has a single parent, as well as graphs with multi-parent nodes, representing events with several causal factors. The analysis of trees is carried out in Pearl [1982], and the extension to general singly connected graphs is reported in Kim and Pearl [1983]. In both cases, the belief-updating scheme possesses the following properties:

1. New information diffuses through the network in a single pass, i.e., equilibrium is reached in time proportional to the diameter of the network.

2. The primitive processors are simple, repetitive, and they require no working memory except that used in matrix multiplication.

3. The local computations and the final belief distribution are entirely independent of the control mechanism that activates the individual operations. They can be activated by either data-driven or goal-driven (e.g., requests for evidence) control strategies, by a clock, or at random.

Thus, this architecture lends itself naturally to hardware implementation, capable of real-time interpretation of rapidly changing data. It also provides a reasonable model of neural nets involved in cognitive tasks such as visual recognition, reading comprehension [Rumelhart, 1976], and associative retrieval [Anderson, 1983], where unsupervised parallelism is an uncontested mechanism.

5. MANAGING LOOPS AND THE DEVELOPMENT OF CAUSAL MODELS

The efficacy of singly-connected networks in supporting autonomous propagation raises the question of whether similar propagation mechanisms can operate in less restrictive networks (like the one in Figure 1), where multiple parents of common children also possess common ancestors, thus forming loops in the underlying network. If we ignore the existence of loops and permit the nodes to continue communicating with each other as if the network was singly-connected, it will set up messages circulating indefinitely around the loops and the process most probably will not converge to a coherent equilibrium.

A straightforward way of handling the network of Figure 1 would be to appoint a local interpreter for the loop x_1, x_2, x_3, x_5 that will account for the interactions between x_2 and x_3. This amounts basically to collapsing nodes x_2 and x_3 into a single node, representing the compound variable (x_1, x_2). This method works well on small loops, but as soon as the number of variables exceeds 3 or 4, collapsing requires handling huge matrices and washes away the natural conceptual structure embedded in the original network.

A second method of propagation is based on "stochastic relaxation" [Hinton, Sejnowski and Ackley, 1984]. Each processor interrogates the states of the variables within its influencing neighborhood, computes a belief distribution for the values of its host variable, then randomly selects one of these values with probability given by the computed distribution. The value chosen will subsequently be interrogated by the neighbors upon computing their beliefs, and so on. This scheme is guaranteed convergence, but usually requires very long relaxation times to reach a steady state.

A third method called ***conditioning*** is based on the ability to change the connectivity of a network and render it singly connected by instantiating a selected group of variables. In Figure 1, for example, instantiating x_1 to some value would block the pathway x_2, x_1, x_3 and would render the rest of the network singly connected, where the propagation techniques of the preceding section are applicable. Thus, if we wish to propagate the impact of an observed data, say at x_6, to the entire network, we first assume $x_1 = 0$, propagate the impact of x_6 to the variables $x_2, \ldots, x_5$, repeat the propagation under the assumption $x_1 = 1$ and, finally, linearly combine the two results weighed by the prior probability $P(x_1)$. It can also be executed in parallel by letting each node receive, compute, and transmit several sets of parameters, one for each value of the conditioning variable. This mode of propagation is not foreign to human reasoning. The terms "hypothetical" or "assumption-based" reasoning, "reasoning by cases," and "envisioning" all refer to the same basic mechanism of selecting a key variable, binding it to some of its values, deriving the consequences of each binding separately, and integrating those consequences together.

Finally, an approach is described in Pearl [1984] which introduces auxiliary variables and permanently turns the network into a tree. To understand the basis of this method, consider an arbitrary tree-structured network. The leaves in this network are tightly coupled in the sense that no two of them can be separated by the others, and therefore, if we were to construct a Bayes network with these variables ***alone***, a complete graph would ensue. Yet, together with the intermediate variables, the interactions among the leaf variables are tree structured, thus demonstrating that some networks can be broken up into trees by introducing dummy variables. This scheme enjoys the advantage of uniformity: the processors representing the dummy variables can be identical to those representing the real variables, in full compliance with our architectural objectives. Moreover, there are strong reasons to believe that the process of reorganizing data structures by adding fictitious variables mimics an important component of conceptual development in human beings, the evolution of causal models.

People often invent hypothetical unobservable entities such as "ego", "elementary parti-

cles", and "supreme beings" to make theories fit the mold of causal schema. When we try to explain the actions of another person, for example, we invariably invoke abstract notions of mental states, social attitudes, beliefs, goals, plans, and intentions. Medical knowledge, likewise, is organized into causal hierarchies of invading organisms, physical disorders, complications, clinical states, and only finally, the visible symptoms. Computationally speaking, we can interpret these mental constructs as names given to memory locations that encode a summary of the interaction between the visible variables and, once calculated, permit us to treat the visible variables as if they were mutually independent. Thus, the restructuring of Bayes networks into trees by introducing auxiliary variables shares many computational features with the development of causal models in people. It is suggestive, therefore, to identify the auxiliary variables with the mental constructs of "hidden causes", and to conjecture that humans' relentless search for causal models is motivated by their desire to achieve computational features similar to those offered by tree-structured Bayesian networks.

REFERENCES

Anderson, John R., (1983), "The Architecture of Cognition", Harvard University Press, Cambridge, MA.

Hinton, G.E., Sejnowski, T.J., and Ackley, D.H., (1984), "Boltzman Machines: Constraint Satisfaction Networks that Learn", Technical Report CMU-CS-84-119, Department of Computer Science, Carnegie-Mellon University.

Kim, J. and Pearl, J., (1983), "A Computational Model for Combined Causal and Diagnostic Reasoning in Inference Systems", ***Proceedings of IJCAI-83***, 190-193.

Pearl, J., (1982), "Reverend Bayes on Inference Engines: A Distributed Hierarchical Approach", ***Proc. AAAI Nat'l. Conf. on AI,*** Pittsburgh, PA, pp. 133-136, August.

Pearl, J., (1984), "Learning Hidden Causes from Empirical Data", Technical Report R-38, UCLA Computer Science Dept. To be published in the ***Proceedings of IJCAI-85***.

Shastri, L. and Feldman, J.A., (1984), "Semantic Networks and Neural Nets", TR-131, Computer Science Dept., The University of Rochester, Rochester, NY, June.

Lowrance, J. D., (1982), "Dependency-Graph Models of Evidential Support", COINS Technical Report 82-26, University of Massachusetts at Amherst.

McAllester, D., (1980), "An Outlook on Truth Maintenance", Artificial Intelligence Laboratory, AIM-551, Cambridge: MIT.

Rosenfeld, A., Hummel, A., and Zucker, S., (1976), "Scene Labeling by Relaxation Operations", ***IEEE Trans. on Computers,*** pp. 562-569.

Rumelhart, D. E., (1976), "Toward an Interactive Model of Reading", ***Center for Human Info. Proc. CHIP-56,*** UC San Diego, La Jolla, CA.

Waltz, D. G., (1972), "Generating Semantic Descriptions from Drawings of Scenes with Shadows", AI TR-271, AI Laboratory, Massachusetts Institute of Technology, Cambridge, MA.

EXPERT VARIANCE: DIFFERENCES IN SOLVING A DYNAMIC ENGINEERING PROBLEM[1]

Michael Prietula
Computer and Information Science Program

Frank Marchak
Department of Psychology

Dartmouth College
Nathan Smith Building
Hanover, NH 03755

In this research we examine how expert engineers differ in solving a specific engineering problem. This effort arose out of observations and experiences with a project originating at the Center for Advanced Engineering Study at the Massachusetts Institute of Technology. In this previous project, one of us developed a simulation as part of a National Science Foundation effort to study new approaches to continuing education for practicing engineers and scientists in industry.[2] During the development of the simulation, it became apparent that this simulation presented a quite interesting environment to study how people reason about complex, physical systems. As a consequence, we are using a version of this simulation to investigate a variety of phenomena in engineering problem solving. This paper reports on some of our initial findings. Specifically, we examine the concept of "expert variance" with respect to one aspect of problem solving: the strategies evidenced by the sequence of steps used to solve the problem.

METHOD

Subjects. Four engineering professors from the Thayer School of Engineering at Dartmouth College participated in the study. Each held a Ph.D. degree or the equivalent and had a minimum of 15 years experience in teaching and/or industry.

Materials. The steam system was reimplemented on an Apple Macintosh© computer. The simulated system (see Figure 1) depicted both components (via icons) and flows of steam and condensate (the numeric values). Two boilers burn fuel to heat the feedwater and produce steam to be fed at high pressures into the high-pressure header (HPH) for distribution. Steam output from the HPH feeds into a pump, a turbogenerator (for the production of electricity) and, if too much high-pressure steam is being produced, into a pressure reducing valve (which reduces the pressure and shunts it to the next header). Intermediate-pressure steam is produced by the third boiler and fed into the intermediate-pressure header (IPH). The IPH distributes steam to the process, a crusher, a second turbogenerator, and a second pressure reducing valve. The low pressure header is fed by both turbogenerators, the crusher, the pump, and a pressure reducing valve. The low pressure header (LPH) provides a second intput to the process, feeds the deaerator, and a vent (excess steam is shunted to the atmosphere). Steam condensate is fed back to the deaerator from a turbogenerator and the process. Loss of water from the system is handled through a makeup water source. The deaerator prepares the returned condensate and excess steam for reintroduction to the boilers. Subjects used a mouse to manipulate a pointer on the screen, enabling them to

select components. For each component, the options of changing its current operating value or finding its current cost was available. In addition, the current and prior total system costs were continuously displayed as was the balance of the system. When a value was changed by a subject, the input value was first evaluated to insure it met the constraints associated with that component. Next the system was updated, which involved calculating the new cost for both the device and the entire system, updating these values, and searching for and indicating on the screen the exact sources of imbalance (if any). Further, the new value, the device involved and the current time were recorded to provide a trace of the subject's manipulations.

Procedure. The subjects were faced with the following problem:

> *The system, in its initial configuration, is operating in a very inefficient manner. Attempt to minimize the cost of running the system by redirecting and modifying the flows of steam and condensate subject to the constraints of the system indicated in the written materials supplied.*

Given a description of the system and related equations, the subjects interacted with the simulation, verbally describing their actions and reasoning. The optimal value was not initially presented. After approximately thirty minutes or when the subject indicated that a solution had been reached, the optimal value for the system configuration was presented. Provided this goal had not already been met, the subject was asked what other changes could be made to his configuration to attempt to reach this goal. The subject then made further modifications to the system or described what strategies would be employed to meet this criterion. An entire session lasted approximately one hour.

SUMMARY OF RESULTS

The analysis of the data consisted of the examination of the modification trace, experimenter observations, and recorded verbal protocol in order to determine (1) the nature of the components of reasoning and (2) how expert reasoning differed based on the identified components.

Components of Reasoning. Analysis of the components of reasoning indicated that experts similarly relied on three primary types of knowledge brought to bear in solving this problem: knowledge of **devices**, knowledge of **systems** of devices, and knowledge of **strategies** of energy conservation regarding systems and devices.[3]

Knowledge of **devices** (i.e., components) embodied the types of physical objects which may be found in a typical processing plant relying on steam use (e.g., turbogenerators, pressure reducing valves, vents). We found that three types of device "roles" were evidenced in solving the problems. First, there is the *structural* role where the device is simply described in terms of its particular properties. This is similar to Kuiper's[4] interpretation, but in this context applicable to the device level. For example, a "boiler" device may be described as a steam producing object which ingests fuel and feedwater, operates at some level of efficiency, and provides steam at some rate and pressure. A particular boiler device has properties instantiated to specific values (e.g., burns oil, is 80% efficient, produces steam at 110,000 lbs/hr), but also would have a specific purpose and function in the context of the behavior of the entire system. This is similar to Kuiper's *functional* description. However, we found that devices take on different functional descriptions based on their role in a particular strategy. For example, a boiler is a steam producing device -- its purpose is to generate steam to be propogated throughout the

system. Subjects occasionally used boilers "to balance" rather than purely "to produce." This is a subtle difference, but one reflecting a very particular role assigned to a boiler usually reserved for vents or pressure reducing valves. In this case, the additional role taken on by the boiler was determined by constraints inherent in a particular strategy and permitted by the understanding of the functioning of the device.

Knowledge of **systems** is knowledge of configurations. It refers to knowledge of how collections of devices operate together as a system. There are two characteristics which seem to distinguish knowledge of systems from knowledge of devices: interaction and collective purpose. *Interaction* refers to the propogation of effects, feedback, and interdependencies of several devices. Essentially, this reflects the appreciation of the dynamics and interconnectivity of the problem. *Collective purpose* refers to the attribution of a purpose to a collection of devices in the context of the system. For example, using a high level of abstraction, a simple description of the overall system behavior can be made by defining the plant as a set of interconnected systems which (1) produce steam, (2) distribute steam, (3) consume steam, (4) produce electricity, and (5) consume electricity. This representation reflects the devices, their role in the system, and the topology of the system indicated by connectivity flows.

The subjects also demonstrated knowledge of general **strategies** of energy conservation to be applied in reducing energy waste and monetary loss. Two dominant strategies were identified: (1) decrease FUEL Cost -- *minimize the cost of purchasing fuel for boilers*, and (2) generate ELECTRICITY -- *generate as much electricity inhouse as possible*. These two strategies correctly address the major sources of cost savings in the simulation. In addition, two additional strategies were identified, but served more of an ancillary or tuning role than the prior two: (3) attend to PRVs -- *pressure reducing valves (PRVs) should be used mainly as an intermediate control method to maintain steam balance* and (4) attend to VENTs -- *venting steam should be avoided and used only as a temporary mechanism to handle steam fluctuations.*

In summary, the experts relied on similar types of knowledge: of devices, systems, and strategies to achieve low cost solutions.

Source of Expert Variance. Three basic differences between experts were found. First, they differed in their ability to generate a minimum cost solution in the absence of an explicit goal (recall that the system was presented to the subjects without the minimum goal cost initially available). Second, the experts differed in the dominant strategy selected (of the first two previously mentioned). Third, they differed in the way the particular strategy was implemented.

Cost Reduction. Two subjects (S1 and S2) did not reach the minimum and, consequently, were then shown the goal and permitted to make further adjustments. In both cases, the subjects quickly achieved the minimum cost after the goal was presented. On the other hand, the other two subjects (S3 and S4) did successfully achieve the minimum cost without the goal present.

Overall Strategies. S1 and S2 differed in their initial attack of the problem. S1 incorporated the FUEL-oriented strategy by attempting to base-load (i.e., increase the output of the boiler to the maximum) the cheaper-fueled boilers and reduce the load on the expensive boiler. When S1 was shown the goal cost (indicating

that additional cost reductions could occur), attention was immediately paid to the turbogenerators and the ELECTRICITY strategy unfolded to permit the proper adjutments. S1 basically approached the task with the idea that lowering the cost of the boilers and redirecting steam were the most important things to do. This caused S1 to achieve a "local" minimum but not a global one. Only after the explicit goal was presented did the explicit "power vs. oil" tradeoff correctly occur. On the other hand, S2 immediately began by invoking the ELECTRICITY strategy and attempted to generate as much electricity as possible while balancing the system via the PRVs and the vent. Again, S2 did not achieve the minimum cost and was shown the goal. Almost in converse fashion from S1, S2 then focused on inhouse electrical production and continued to manipulate the flow parameters of the turbogenerator outputs in order to maximize the amount of power purchased within the constraints of the balanced system and level of steam production.

S3 and S4 both achieved the minimum cost solution without requiring the goal to be presented; however, they accomplished this quite differently. S3 was very methodical and essentially incorporated a strategy which was a combination of the ELECTRICITY and FUEL approaches. S3 attempted to produce as much electricity as possible while at the same time minimizing the use of the expensively-fueled boiler and balancing the adjustment with the vent. S4, however, was unique. With a minimum number of moves, S4 sequentially invoked the FUEL and ELECTRICITY strategies. S4 definitely was the quickest and most efficient of the subjects.

Implementation. An additional observation can now be made concerning how the strategies were implemented. The strategies identified involve adjustments to several devices at a time. For example, to increase the electricity both turbogenerators are involved along with one or more boilers -- thus defining the *device set* of the strategy. We found that the subjects took essentially two approaches to implementing a strategy and adjusting the device set. In one method, a subject would select a strategy, such as ELECTRICITY, and implement it "component- by-component" in a cautious manner where one element of the device set would be changed (e.g., a turbogenerator), the results tested (i.e., is it in balance?), and, if necessary, adjustments made to balance the system (to either members or nonmembers of the device set). The subject would then move on to the next device modification in the strategy (e.g., the other turbogenerator). This reflects a **locally-guided** approach sensitive to imbalances as the strategy unfolds. Subject S3 demonstrated this perfectly be attempting to explicitly balance the system on four separate occasions. On the other hand, the second type of implementation tolerated intermediate imbalances as each component of the device set was addressed. In this **globally-guided** approach, the subjects would first make adjustments to all members of the device set prior to any attempt to balance the results by adjusting nonmember components. The extreme case is illustrated by S4 who essentially was not concerned with balancing the system until the final adjustments were made. In one sense, a locally-guided search represents a cautious tactic by testing the effects of the device set adjustments. The globally-guided tactic seemed to reflect both a more optimistic confidence in the presumed behavior of the system and a commitment to a specific series of modifications.

CONCLUSION

Expertise is more than a large accumulation of facts; it concerns basic qualitative differences in the representation of knowledge and the control of the reasoning processes. The analysis of expert reasoning, as we have indicated, uncovers a picture of comparisons which is more complicated that simple expert-novice dichotomies often suggest. As expected, experts as group distinguish themselves (or are distinquished) from the flock via performance. They do what they do more proficiently and efficiently than the others. They have mastered the body of knowledge upon which proficiency is built. Experts, however, do not only know more -- they know differently. Humans are limited in their ability to deal with large amounts of information; consequently, they must incorporate ways to reduce the demands of the task during performance. It is precisely *because* of such cognitive limitations that experts have adapted efficient ways to solve problems in their domains. Expert problem solvers have augmented and modified their knowledge through years of experience. Lack of this experience often makes problem solvers "knowledge rich but strategy poor." More relevant to this discussion, it also provides a source for individual variation among experts. Experts in the same field simply may not demonstrate the same reasoning behaviors.[5] Experience, then, is both a source of expert performance and expert variance.

We have reported some of our initial findings regarding the nature of expert variance in problem solving. There are, of course, many additional avenues to pursue. For example, we are examining expert-novice differences, explicating more detailed information on the nature of the qualitative and causal judgements underlying the selection and execution of the strategies, and have implemented an OPS5 model of the strategies. We shall see...

FOOTNOTES

[1] This project was supported by a grant from the Control Data Corporation, Education and Training Research Group. We express our appreciation to Frank Roberts of CDC, the Schools of Engineering at Dartmouth and MIT for their cooperation and support as well as Professor Michael Mohr of MIT.

[2] Project PROCEED (Program for Continuing Engineering Education) is a National Science Foundation effort in conjunction with the Center for Advanced Engineering Study at MIT. The purpose of Project PROCEED is to develop and demonstrate a nation-wide system for continuing education in industry by developing educational materials based on detailed case studies of actual engineering problems.

[3] It is also certain that the experts commonly relied on basic thermodynamic **concepts** such as energy and material balance, enthalpy, heat rate, and power generation; however, this evidence was generally always implicit in their interpretation of system behavior and function and rarely expressed. In fact, it is this knowledge which underlies the nature of causality and explanation on a device level. The next step in the research is to examine this level of reasoning.

[4] **Kuipers, B.** Commonsense reasoning about causality: Deriving behavior from structure, *Artificial Intelligence, 24,* 1984.

[5] **Feltovich, P.** *Knowledge-based components of expertise in medical diagnosis.* Ph.D. dissertation, University of Minnesota, 1981 (Published as University of Pittsburgh, LRDC Report PDS-2, September 1981) and **Prietula, M.** *An investigation of reasoning methods used in physical database design problem solving,* Unpublished Ph.D. dissertation, University of Minnesota, April 1985.

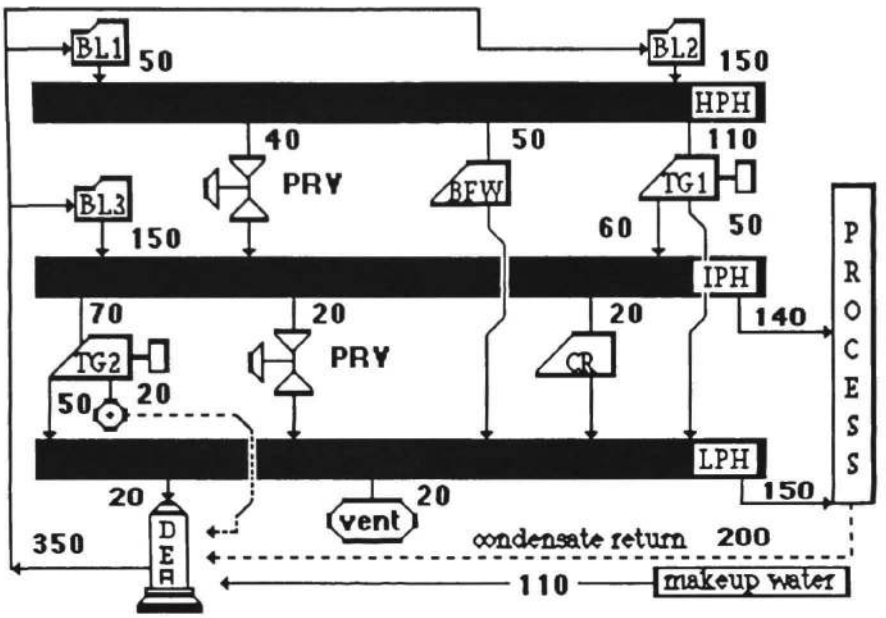

BL1
50
BL2
150
HPH
40
50
110
PRV
BFW
TG1
BL3
150
60
50
IPH
70
20
20
140
TG2
PRV
CR
50
20
LPH
20
20
150
vent
350
DEA
condensate return 200
110
makeup water
PROCESS

Figure 1

MACHINE UNDERSTANDING AND DATA ABSTRACTION IN SEARLE'S CHINESE ROOM

William J. Rapaport

Department of Computer Science and Graduate Group in Cognitive Science
University at Buffalo, State University of New York, Buffalo, NY 14260

1. INTRODUCTION.

In John Searle's Chinese-Room thought experiment, Searle, who knows neither written nor spoken Chinese, is locked in a room supplied with instructions in English that provide an algorithm allegedly for understanding written Chinese. Native Chinese speakers outside the room pass questions written in Chinese characters into the room; Searle uses these symbols, otherwise meaningless to him, as input and, following only the algorithm, produces, as output, answers written in Chinese characters, passing them back to the native speakers. The "answers ... are absolutely indistinguishable from those of native Chinese speakers" (Searle 1980: 418). The experiment is used to support the following argument:

> [I] still don't understand a word of Chinese and neither does any other digital computer because all the computer has is what I have: a formal program that attaches no meaning, interpretation, or content to any of the symbols. [Therefore,] ... no formal program by itself is sufficient for understanding .... (Searle 1982: 5.)

Many have disagreed over what Searle's argument is designed to show. The version I have just cited is clearly invalid: At most, the thought experiment might show that that particular program is insufficient for understanding, but not that a program that did attach meaning, interpretation, or content to the symbols could not understand. Such an attachment would also take the form of an algorithm (cf. Rapaport, forthcoming). But Searle denies this stronger claim, too:

> I see no reason in principle why we couldn't give a machine the capacity to understand English or Chinese, since in an important sense our bodies with our brains are precisely such machines. But ... we could not give such a thing to a machine ... [whose] operation ... is defined solely in terms of computational processes over formally defined elements. (Searle 1980: 422; cf., also, the "robot reply", p. 420.)

And this is so because "only something having the same causal powers as brains can have intentionality" (Searle 1980: 423). These causal powers are due to the (human) brain's "biological (i.e. chemical and physical) structure" (Searle 1980: 422). The *biological* stance taken by Searle is essential: For even a simulated human brain "made entirely of old beer cans ... rigged up to levers and powered by windmills" would not really exhibit intentionality (Searle 1982: 4), even though it appeared to. Searle, however, does not specify precisely what these causal powers are, and this is the biggest gap in his argument.

However, in his book, *Intentionality*, Searle tells us that "mental states are both *caused by* the operations of the brain and *realized in* the structure of the brain" (Searle 1983: 265), so we might hope to find an explication of these causal powers here. Indeed, a careful analysis of these two notions reveals (1) what the requisite causal powers are, (2) what is wrong with Searle's claim about mental states, and (3) what is wrong with his overall argument.

Moreover, it is consistent with my analysis that *some* intentional phenomena, e.g., pain and other qualia or "feels", *need not* be functionally describable, but, rather, might be the results of physical or chemical properties of the entity that experiences them. However, though these phenomena *need* not be functionally describable, they very well might be.

My theory, in rough outline, is this: Consider Searle's beer-can-and-windmill simulation of a human brain, programmed to simul thirst. Searle says that it is *not* thirsty. We might reply that perhaps it feels *simulated* thirst; and we might then go on to wonder if simulated thirst *is* thirst. Even better, we should say that it *simulatedly* feels simulated thirst. Similarly, the Chinese computer system simulatedly understands simulated Chinese. But, so goes my theory, the simulated feeling of simulated thirst *is* thirst, and simulated understanding *is* understanding. The differences between such simulations and the "real" thing—or, more to the point, the *human* thing—lie in their physical make-up. The first implemented in the beer-can computer may not "feel" the way that thirst implemented in a human feels (what, after all, is it like to be a thirsty beer-can computer?), but they are both *thirst*. And so for understanding.

2. QUESTIONS ABOUT CAUSATION AND REALIZATION.

We need to clarify what Searle means by "causation" and "realization". In the passage cited above, he says that it is brain *operations* that *cause* mental states, whereas it is brain *structure* that *realizes* them. This difference proves important. Yet, earlier in *Intentionality*, Searle answers the "ontological" question, "What is the mode of existence of ... Intentional states?" in two ways: by saying that they "are both caused by and realized in [a] the *structure* of the brain" and [b] "the *neurophysiology* of the brain" (Searle 1983: 15, my italics).

But which is it? Operations and structure are, arguably, things that can be shared by brains and beer-can contraptions. Since Searle clearly does not want to commit himself to that, the answer must be neurophysiology. Very well, then. What does Searle mean when he says that intentionality is *caused by* the neurophysiology of the brain? He means that "Intentional states stand in *causal* relations to the neurophysiological" (Searle 1983: 15). But what does *that* mean?

And what does Searle mean when he says that intentionality is *realized in* the neurophysiology of the brain? I shall argue that he means that it is "implemented" in the brain, using a somewhat technical sense belonging to the computational theory of abstract data types; but that theory is more complex than Searle realizes.

3. DATA ABSTRACTION AND IMPLEMENTATION.

Computer programs describe *actions* to be performed on *objects* (cf. Lecarme 1981: 60-61). The actions are expressed in terms of the operations of the particular programming language used, and the objects are represented by *data structures*, each of which must be constructed out of the data structures available in the particular programming language. Data structures can be classified into different data *types*. An *abstract data type* is a formal (i.e., mathematical or abstract) data structure, together with various characteristic operations that can be performed on it (cf. Aho *et al.* 1983: 10-14). An *implementation* of an abstract data type is (usually) an actual data structure in a program, that plays the role of the abstract data type. This characterization is admittedly rough, but will serve my present purposes.

An example should help. A *stack* is an abstract data type consisting of potentially infinitely many items of information ("data") arranged ("structured") in such a way that new items are added only to the "top" of the stack and an item can be retrieved only if it is on the top. The usual image is that of a stack of cafeteria trays: the last item put on the stack is the first to come off. The programming language Pascal does not have stacks as a built-in data type, but they can be implemented in Pascal by arrays, which *are* built in. Unlike a stack, an item can be added to or removed from any cell of an array: But if one steadfastly refuses to do that and steadfastly treats an array in a last-in/first-out manner, then it is, for all practical purposes, a stack. Indeed, real stacks of cafeteria trays are more like arrays than stacks.

The relation between an abstract data type and an implementation of it is reminiscent of that between an Aristotelian genus or species and an individual of that genus or species, and implementation is reminiscent of instantiation—but not exactly: An Aristotelian individual has essential properties, namely, those had by the species to which it belongs. Its accidental properties are those that differentiate it from other individuals of its species.

But, whereas a stack has a top essentially, an array has one only accidentally. And two implementations of a stack can differ in more than merely accidental ways: Arrays are essentially different from linked lists, yet both can implement stacks. And stacks are essentially different from queues, yet arrays can implement them both.

These differences arise from the fact that not all properties of an abstract data type need be "inherited" by an implementation, nor does the abstract data type have to have all the essential properties of its implementation. For instance, stacks are infinite; arrays in Pascal are finite and of fixed size. Arrays that implement stacks *can* be accessed in the middle (even if they shouldn't be); stacks cannot. Finally, one abstract data type can implement another. Thus, e.g., the abstract data type *sequence* can be implemented by the abstract data type *linked list*, which, in turn, can be implemented by *symbolic expressions* in LISP. These can be thought of either as a "real" implementation, or as yet another abstract data type ultimately to be implemented by electronic signals in a computer.

Since these notions play an important role in my theory, a few applications of them to other areas may prove helpful. In doing this, I shall be likening certain things to abstract data types; I do not intend to argue that all of these things *are* abstract *data* types (although some are). Consequently, I shall use terms such as *Abstraction* to refer to the general category of such things.

When you "listen to music" on a record, are you *really* listening to music or merely to a recording—a simulation—of music? Clearly, both, because recordings of music *are* music. A musical score is an Abstraction that is implemented by, say, an orchestra. The relations between an abstract data type and an implementation of it are precisely those between a score and a performance of it. Even the "implementation" of an abstract data type by the mathematical notation used to describe it is paralleled by the relation between the abstract score and its printed copy.

Here is a mathematical example: Peano's axioms describe the abstract data type *natural numbers*. Any sequence of elements satisfying those axioms is an implementation of "the" natural numbers and, thus, *is* a sequence of natural numbers. Similarly, rational numbers can be implemented as equivalence classes of ordered pairs of (any implementation of) natural numbers.

And an example from science: Different collections of water molecules and collections of alcohol molecules are implementations of the Abstraction *liquid*.

And, more to the point, perhaps mental phenomena, like abstract data types, are Abstractions that can be implemented in different media, say, human brains as well as electronic computers.

4. REALIZATION AS IMPLEMENTATION.

We can now turn to what Searle means when he says that intentionality is "realized in" the brain. As a first approximation, let us say that:

(T1) *A is realized in B* means: *A* is an Abstraction implemented in *B*.

Since Searle claims that intentionality is also "caused by" the brain, we should inquire into the relationship between *being caused by* and our new understanding of *being realized in*.

Suppose, first, that *A* is caused by *B*. Is *A* realized in *B*, in the sense of (T1)? The motion of billiard ball #1 may be caused by the motion of billiard ball #2, yet #1's motion is not realized in #2's. Kennedy's death was caused by, but not realized in, Oswald. The *Mona Lisa* was (efficiently) caused by, but surely not realized in, Leonardo. *But*: The *Mona Lisa* was (materially) caused by Leonardo's canvas, and it surely is thereby realized therein. An American flag might be caused by Fourth-of-July fireworks, and thereby realized therein. And the simulation of a stack can be caused by the execution of a program containing arrays; the stack is thereby realized by the execution of the program. So, *being caused by* does not imply *being realized in*, but it is not inconsistent with it. It is possible for *A* to be caused by, but not realized in, *B*.

Suppose, next, that *A* is realized in *B*, in the sense of (T1). Is *A* caused by *B*? Consider a stack realized in an array. Is the stack caused by the array? This is a strange way to express it, but not a bad way: A *real* stack *was* caused to be by a real array. *But* it is an *abstract* stack that is realized in the array. And similarly for the *Mona Lisa* and the American flag. Thus, let us add to (T1),

(T2) If an Abstraction *A* is realized in *B*, then a real *A* is caused by *B*.

Moreover,

(T3) If an Abstraction *A* is realized in *B*, then real *A is B "under a description"* (or, "in a certain guise").

Is this being fair to Searle? Is his notion of realization the same as the notion of implementation, as captured by (T1)-(T3)? There is evidence to suggest that it is. Here, I shall merely note that (T1) is consistent with Searle's claim that "mental phenomena cannot be reduced to something else or eliminated by . . . re-definition" (Searle 1983: 262), as long as by 'mental phenomena' we understand something abstract. For to implement an Abstraction is *not* to reduce it to something (because of the only partial overlap of properties) *nor* is it to eliminate it (for the same reason, as well as because of the possibility of distinct implementations). I would suggest that "implementationism" might prove to be a more fruitful, less constraining viewpoint in the philosophy of science than reductionism has been.

Before pushing on, let me remind you of my purpose. Searle argues that a computer running the Chinese program does not understand Chinese, because only human brains can thus understand. Understanding, even if describable in a computer program, is biological; hence, an electronic computer cannot understand. On the other hand, I am arguing that there can be an abstract notion of understanding—a functional notion, in fact, a computational one—that can be implemented in computers as well as in human brains; hence, both can understand. I gain my leverage over Searle by making a distinction that is implicit in his theory, but that he fails to make. Just as an array that implements a stack does not do so in the way that a linked list does, so a computer that implements the understanding of Chinese need not do so in precisely the way that a human does. Nevertheless, they both understand.

The distinction that Searle does not make explicit can be discerned in the following passage:

> [M]ental states are as real as any *other* biological phenomena, as real as lactation, photosynthesis, mitosis, or digestion. Like these other phenomena, mental states are caused by biological phenomena and in turn cause other biological phenomena. (Searle 1983: 264; my italics.)

The use of 'other', here, perhaps begs the question. But it also suggests that by 'mental states' Searle means *implementations of abstract mental states*. I shall refer to these as *implemented mental states*.

We are now in a position to see where Searle is led astray. It is simply false to say, as Searle does, that one kind of thing, "mental states[,] are both *caused by* operations of the brain and *realized in* the structure of the brain" (Searle 1983: 265). Rather, it is one thing—an implemented mental state—that is caused by the operations of the brain, and it is something else altogether—an abstract mental state—that is realized in the structure of the brain.

For example, a "liquid", considered as an Abstraction, *can* be realized (implemented) in a collection of molecules. But what that collection *causes* is *actual* (or implemented) liquidity. Similarly, the Abstraction, "liquid-properties-of-water", can be realized (implemented) in *different* collections of water molecules, but the *actual* liquid properties of (some particular sample of) water are caused by *different* actual behaviors. Moreover, the Abstraction, "liquid" (*simpliciter*) can certainly be realized (implemented) in different collections of molecules (water molecules, alcohol molecules, etc.), and the actual liquidity of these molecules is caused by *different* actual behaviors; yet *all* are liquids, *not* because of the *causal* relationship, but because of the *realizability* relationship.

So, too, the Abstraction, "understanding" can be realized (implemented) in both humans and computers; *actual* understanding can be caused by both humans and computers; but, because of the realizability relationship, *both are* understanding.

5. THE RELATIONSHIPS BETWEEN CAUSATION AND REALIZATION.

In an analogy, Searle (1983: 269) offers an analysis of a combustion engine (see Figure 1), consisting of high-level phenomena causing other high level phenomena, each of which is both caused by and realized in corresponding low-level phenomena, one causing the other, respectively, as in Figure 2.

But Searle's diagram (Fig. 1) is not detailed enough; the complexity hinted at in (T3) needs to be made explicit. Searle's diagram should be augmented by a mid level analysis, as in Figure 3. More distinctions appear in Figure 3 than may actually be needed; we can, however, distinguish the following:

- a certain kind of relationship (causal, according to Searle)—call it "causation-1"—between the low-level phenomena and the mid level phenomena;
- a certain (arguably distinct) kind of causal relationship—call it "causation-2"—between low level phenomena;
- a certain kind of causal relationship—call it "causation-3"—between mid-level phenomena, paralleling (and possibly distinct from) causation 2;
- a certain kind of relationship (arguably causal)—call it "causation 4"—between high level phenomena, paralleling (and possibly distinct from) causation 1 and causation 2;
- a certain kind of relationship—call it "R"—between mid and high-level phenomena; and

— the relationship of realization (or implementation) between low- and high-level phenomena.

How are causation-1, causation-2, causation-3, and causation-4 related, and what is R? Searle, we have seen, conflates the several causations. Instead, as the second stage of my theory, I propose the following:

— R is the relation of being an instance of.

— There are several possibilities for causation-1. The simplest, perhaps, is to define it in terms of realization and R-inverse. Another is to take it as what Jaegwon Kim has called "Cambridge dependency"—the sort of "non-causal causation" involved when a sibling's marriage "causes" you to become an in-law. I favor a third interpretation: Castañeda's relation of *consubstantiation* (cf. Castañeda 1972: 13ff, 1975: 145f)—a relation, somewhat like co-referentiality, holding between intensional entities. In any case, causation-1 is *not* a kind of causation at all.

— causation-2 is ordinary, physical causation; and

— causation-3 is definable in terms of causation-1 and causation-2.

Thus, I propose that when a low-level device realizes (implements) a high-level Abstraction, it produces a mid-level phenomenon. A general theory of the relationships of causation (Searlean and otherwise), realization (or implementation), and low-, mid-, and high-level phenomena is presented in Figure 4.

We can now apply these distinctions to the Chinese-Room thought experiment. Machine understanding is, indeed, understanding, just as human understanding is: They are both instances of the more abstract (functional or computational) characterization of understanding. Machine understanding is caused-1 by a computer (or computer program) in which abstract understanding is realized; that is, machine understanding is an instance of abstract understanding, which is realized in a computer.

6. ON THE BIOLOGICAL CHARACTER OF INTENTIONALITY.

This way of looking at the issues clarifies the "biological naturalism" (Searle 1983: 264) that underlies Searle's claim that non-biological computers cannot exhibit intentionality on the grounds that intentionality is biological.

But why must intentionality be biological? Because, according to Searle, only biological systems have *the requisite causal properties* to produce intentionality. What are these causal properties? Those that are "causally capable of producing perception, action, understanding, learning, *and other intentional phenomena* (Searle 1980: 422; my italics). This is nothing if not circular. Moreover, to implement an abstract data type, it is only necessary to have the (physically) realizable properties of the abstract data type. (E.g., for an array to implement a stack, it is sufficient that it can store data and have a "top"; it need not be infinite.) And Searle has offered us no reason to think that intentionality, abstractly conceived, *could* not be implemented in a beer can and windmill device.

But most importantly, the theory presented here shows that "the requisite causal powers" means causation-1, which is not real causation at all. The requisite causal powers, then, are simply the ability to realize a species of an Abstraction (or to be consubstantiated with an instance of an Abstraction). The present theory makes it clear that "*causality*" *is a red herring*. Only realizability (and perhaps consubstantiation) counts.

7. CONCLUSION.

The relationship between an Abstraction and its implementations underlies machine understanding of natural language (and of AI in general). It is unlike the more familiar relationship between species and individuals (or universals and particulars). Nor is it the case that implementations are "reductions" of Abstractions. Thus, there is no need to advocate a *reduction* of the mental to the biological, eliminative or otherwise. Rather, the mental is *implementable* in the biological, as well as in the digital-electronic. Implementability is a notion worth further study.

REFERENCES

Aho, Alfred V., John E. Hopcroft, and Jeffrey D. Ullman, *Data Structures and Algorithms* (Reading, MA: Addison-Wesley, 1983).

Castañeda, Hector-Neri, "Thinking and the Structure of the World," *Philosophia* 4(1974)3-40. Originally written in 1972. Reprinted in 1975 in *Critica* 6(1972)43-86.

_____, "Identity and Sameness," *Philosophia* 5(1975)121-50.

Kim, Jaegwon, "Noncausal Connections," *Nous* 8(1974)41-52.

Lecarme, Olivier, "Pascal," *Abacus*, Vol. 1, No. 4 (Summer 1984): 58-66.

Rapaport, William J., "Searle's Experiments with Thought," *Philosophy of Science* (forthcoming).

Searle, John R., "Minds, Brains, and Programs," *Behavioral and Brain Sciences* 3(1980)417-57.

_____, "The Myth of the Computer," *New York Review of Books* (29 April 1982): 3-6; cf. correspondence, same journal (24 June 1982): 56-57.

_____, *Intentionality: An Essay in the Philosophy of Mind* (Cambridge: Cambridge University Press, 1983).

Shoemaker, Sydney, "Functionalism and Qualia," in N. Block (ed.), *Readings in Philosophy of Psychology*, Vol. 1 (Cambridge: Harvard University Press, 1980): 251-67.

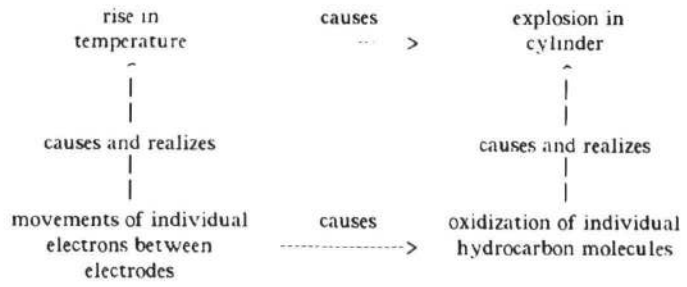


Figure 1

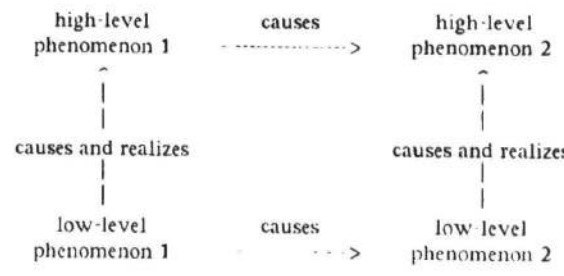


Figure 2

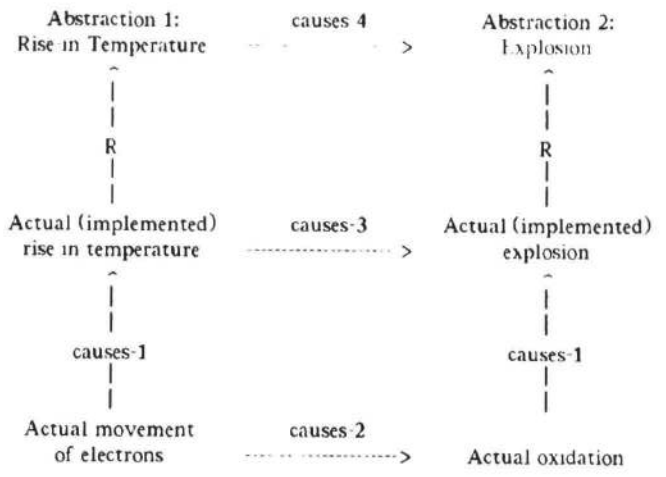


Figure 3

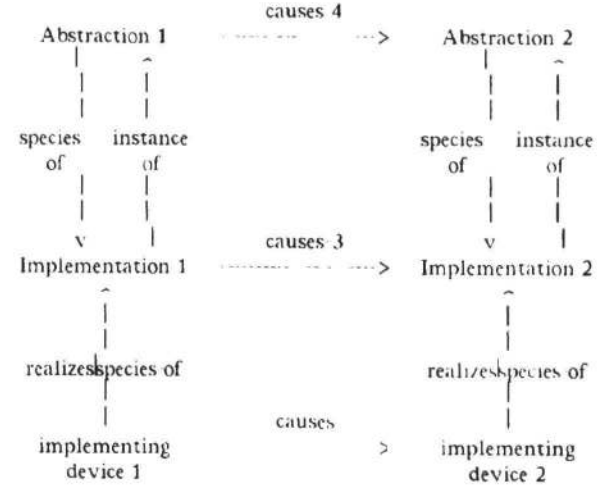


Figure 4

Toward a Unified Model of Deception

Donald D. Rose

Irvine Computational Intelligence Project
Department of Information and Computer Science
University of California, Irvine, 92717

Abstract

We will first argue that ignoring possible deception in multi-agent scenarios can lead to planning failures; specifically, we show how standard deduction may be able to solve the Wise Man Problem, but not a variant where some agents are deceptive (i.e., the Wise-Yet-Deceitful Man Problem, or W-Y-D). Second, we will show how to avoid planning failures in scenarios such as W-Y-D, by developing models of both (1) the deceptive tendencies of other agents, and (2) how these other agents *themselves* reason about deception; the concepts of *best-case* and *worst-case* deceptive agents will be introduced as examples. Third, we will show how to represent *deception axioms* within Konolige's deduction model of belief, and the fourth section will more closely analyze how one solves W-Y-D. Finally, we will suggest how the new model developed for this problem can be generalized into a more unified model of deception.

Introduction

This paper deals with increasing the sophistication with which agents reason about *other* agents' beliefs, as well as their own. The particular enhancement of interest here is allowing agents to plan not only for cases where an agent is truthful, but also for cases where he is deceitful (to others, to himself, or to both). Thus, we will relax the assumption that what an agent *says* he believes always equals what he is *actually* believing (i.e., if other agents are fooled by agent *d*, then *d*'s beliefs will not be the same as the others' beliefs *about d*'s beliefs). We begin by illustrating the Wise Man Problem, considered "a good test of the *competence* of any model of belief" [KONOLIGE, 84]. Using W, B and U to represent the possible responses (i.e., black, white, or don't know), here is:

THE WISE MAN PROBLEM. *A king, wishing to know which of his three advisors is the wisest, paints a white dot on each of their foreheads, tells them there is at least one white dot, and asks them to tell the color of their own spots. After a while the first replies U; the second, on hearing this, also replies U. The third then responds W.*

The problem is whether No. 3 can ascertain his color, based solely on seeing the others' dots, and *reasoning about their beliefs.* His reasoning proceeds as follows: "Suppose my spot were black. Then No. 2 would know that his own spot was white (since, if it were black, the first of us would have seen two black spots and thus would have known his own spot's color). Since both answered U, my spot must be white" [KONOLIGE, 84]. Note, however, that No. 3 did not consider the possibility that No. 1 or No. 2 (or both) might give a response that they *did not believe themselves.* Nor does No. 3 address a more subtle question: whether or not *No. 2 plans for deception* as well. Thus, the standard Wise Man Problem led to a correct answer by No. 3. But suppose the following scenario ensued:

THE WISE-YET-DECEITFUL MAN PROBLEM. *The King, after No. 3's success, decides to test No. 3's abilities further. He replaces the other wise men with two evil men and repaints all the dots. The first man, experienced in deceit, gets white; his protege, a naive deceiver, gets black. The third again gets white. The same rules apply: there's at least one white dot, and each must tell the King his color. (The King warns the third of the others' deceptive abilities; the third dismisses this since no one has ever lied before the King.) The first replies W; his protege says B. Then the third man incorrectly responds B. The King banishes him from his court.*

This sad ending could have been avoided if the warning had been properly reasoned with. If No. 3 had codified the other men's deceptive abilities into rules, then added these to his existing rules about how he and the others draw inferences, this planning failure would not have happened. We now look at why No. 3 gave an incorrect answer, and how his reasoning should proceed in order to solve this new problem.

Solving the Wise-Yet-Deceitful Man Problem

Let us use Pi to represent the proposition "*i* has a white dot" (and ¬Pi for black). Now, the way No. 3 reasoned in trying to solve our new problem was the same as for the standard Wise Man Problem. Recalling initial configuration (P1 ∧ ¬P2 ∧ P3), No. 3 reasons: "Suppose I was white. Then No. 1's response would have been U, because he could not have known his color unless No. 2 and I were both black. Since he said W, I must be black." Note that No. 2's response seems to further confirm No. 3's reasoning. Upon seeing No. 3 is black, and hearing No. 1 say W, No. 2 would immediately conclude he was black (again, since seeing two black dots should make No. 1 say W). Since No. 2 *did* say B, No. 3 might feel secure in the conclusion that he's black.

The dilemma here is that agents No. 1 and No. 2 are modelled as completely truthful. However, in real-world environments, agents often interract with other agents that not only might possibly lie, but can often hide such deception. The types of behavior we need to account for here are (1) whether or not an agent is being deceitful, and (2) whether or not that agent himself considers that others may try to deceive *him*. So how does No. 3 solve the Wise-Yet-Deceitful Man Problem? First, we must take into account the King's warning about No. 1 and No. 2. Thus No. 3 will model No. 1 as a *best-case* deceptive agent - i.e., No. 1 may or may not lie, and he believes others may or may not do the same. No. 2 will be modelled as a *worst-case* deceptive agent - i.e., No. 2 *always* lies, yet believes other agents always tell the truth. Finally, everyone knows these traits of both No. 1 and No. 2. Thus, one can see why No. 2's is "worst-case"; in a deceptive world, he's infinitely gullible, yet his infinite deceit is always recognized. No. 1, however, trusts no one (and, since everyone knows it, others think twice before trying to deceive him); in addition, everyone knows his responses are unpredictable (i.e., he may or may not be lying). Thus No. 1 is in the "best" position to capitalise in a deceptive world.

Here is how No. 3 should solve the Wise-Yet-Deceitful Man Problem (W-Y-D). Remembering the initial dot configuration (P1 $\wedge$ $\neg$P2 $\wedge$ P3), a more wary No. 3 reasons: "There are only two possible assumptions: either *No. 1 is truthful and I'm black*, or *No. 1 is lying and I'm white* (see informal proof section for why this is so). Suppose the first assumption holds. Then No. 2, who always believes everyone is truthful, would believe No. 1 when he said W; hence *No. 2 believes he's black* (since seeing two black dots would, in No. 2's view, lead to No. 1's W response). Thus I know that No. 2 would've said anything *except* B at this point, since everyone knows No. 2 always lies. However, No. 2 *did* say B; hence *No. 2 must not believe he's black*. Thus, my first assumption is false, and so the second must be true. Hence, I tell the King I'm W, I expose the fact that No. 1 lied before his majesty – and I keep my job." (Note that telling the King about No. 2's lie is no revelation, since everyone already knows No. 2 always lies.)

Representing Axioms of Deception

The next step is to formally represent axioms that model *all* forms of reasoning an agent may go through in a possibly deceptive environment (i.e., the model should account for any response, and any belief about the truthfulness of the agent responding). In addition to these axioms will be axioms for modelling *specific* agents (e.g. No. 2). All axioms are based on Konolige's deduction model of belief [KONOLIGE, 84]. First, some notation: the *belief operator* $[S_i]$ is used to indicate whether agent i believes a certain proposition. For example, [S3]P3 says that agent No. 3 believes the proposition P3 ("No. 3's dot is white"); [S3]$\neg$P3 says he believes $\neg$P3 ("No. 3's dot is black"). In short, $[S_i]x$ is true if x is in i's belief set, for any proposition or belief x [KONOLIGE, 84].

Now, the final W-Y-D conclusion was that No. 3 believed not only that his dot was white, but also that No. 1 was lying; if L_i is the proposition "agent i lied", then No. 3's final conclusion would be stated as [S3](P3 $\wedge$ L1). In short, the simplest kind of proposition has no belief operator; more complex propositions start with a belief operator (indicating who holds the proposition), and may or may not have belief operators after it, depending on the degree of nesting being represented. (A final note: [S0] means that whatever follows it is a "common belief", and would be held by any agent). Thus, the first four axioms we desire should capture the most basic elements of our W-Y-D scenario: (1) No. 2's dot is black, the others are white; (2) it's a common belief that there's at least one white dot; (3) when an agent has a white dot, all others know it's white (and this rule itself is a common belief); and (4) is the same as (3), but for black dots. Using the formal language of the deduction model of belief, we have:

W1 P1 $\wedge$ $\neg$P2 $\wedge$ P3

W2 [S0](P1 $\vee$ P2 $\vee$ P3)

W3 $(P_i \supset [S_j]P_i) \wedge [S0](P_i \supset [S_j]P_i)$ $\quad i \neq j,\ j \neq 0$

W4 $(\neg P_i \supset [S_j]\neg P_i) \wedge [S0](\neg P_i \supset [S_j]\neg P_i)$ $\quad i \neq j,\ j \neq 0.$

In a world where deception is *not* modelled, the next actions taken would occur after each agent's response, when axioms are asserted stating that (1) the agent believes what he said, and (2) all other agents believed him as well. For example, if No. 1 had responded U ("I don't know") in our problem (where his dot happens to be W - i.e., where P1 is true), the non-deceptive model creates:

W5 $\neg$[S1]P1 $\wedge$ [S0]$\neg$[S1]P1.

This reads: "it is *not* the case that No. 1 believes P1, and it is a common belief that this is so." This is exactly the axiom created when the deduction model is used to solve Konolige's variation of the Wise Man Problem (the Not-So-Wise Man problem, mentioned later; see [KONOLIGE, 84], p. 48). However, this is exactly the type of axiom we must *avoid* if we start planning for the possibility of dishonest responses; *it should not be a universal given that what an agent believes is also believed by everyone else.* Thus, instead of asserting this **W5** axiom (and a similar **W6** after No. 2 responds), we add to the initial axioms six *deception axioms.* Each has the same theme concerning a possibly-deceptive agent i: depending on what i said (and whether or not agent j believes i is lying), j will make some deduction about *what agent i believes about his own dot's color.* For example, if j believes i is truthful when he says W, then j believes that i believes he's white (**W5'** below). In **W10**, however, j believes i is lying when he says he doesn't know his color (i.e., U); thus, j believes i actually *either believes he's white, or believes he's black.* The first three axioms are of interest when j believes i is *not* lying; the last three for when j feels i *is* lying. Remembering that Wi being true means that "i said W", etc., the new axioms are:

W5' $([S_j](\neg L_i \wedge W_i) \supset [S_j][S_i]P_i) \wedge [S0](...)$

W6' $([S_j](\neg L_i \wedge B_i) \supset [S_j][S_i]\neg P_i) \wedge [S0](...)$

W7 $([S_j](\neg L_i \wedge U_i) \supset [S_j](\neg[S_i]P_i \wedge \neg[S_i]\neg P_i)) \wedge [S0](...)$

W8 $([S_j](L_i \wedge W_i) \supset [S_j]\neg[S_i]P_i) \wedge [S0](...)$

W9 $([S_j](L_i \wedge B_i) \supset [S_j]\neg[S_i]\neg P_i) \wedge [S0](...)$

W10 $([S_j](L_i \wedge U_i) \supset [S_j]([S_i]P_i \vee [S_i]\neg P_i)) \wedge [S0](...)$.

([S0](...) abbreviates the fact that the left side of the conjunction is a common belief.) Now, since we assume all six axioms are known to all agents, how can we model an incomplete (less-than-ideal) agent - one who is not cognizant of possible deception - if deception abilities are common beliefs? There are actually two ways in which to model an agent who is naive about deception. First, the agent's rule set may be incomplete because he has left certain deception axioms out of his set of "relevant problem-solving information". Thus, the heart of the axiom, as well as the fact it is common knowledge, would *never be used in the agent's reasoning.* This behavior is called *circumscriptive ignorance* [KONOLIGE, 82]. In short, one way an agent might not plan for possible deception is to exhibit *relevance incompleteness* - by ignoring axioms that are essential to solving a problem, an agent may become ignorant of some of the logical consequences of his beliefs. However, we model naive deceptive agents a second way; we allow all agents to have, and use, the six deception axioms, but define two specific axioms which model No. 2's specific behavior. The first axiom captures No. 2's belief that no agent ever lies; the second states that everyone knows No. 2 always lies:

W11 $([S2]\neg L_i) \wedge [S0]([S2]\neg L_i) \quad i \neq 2$

W12 [S0]L2.

This approach is useful for agents who must reason about No. 2's beliefs: if No. 3 desires, he can always perform voluntary circumscription and *ignore* **W11** (and speculate: "perhaps No. 2 isn't completely gullible"), or can ignore **W12** (i.e., "perhaps No. 2 doesn't always lie"). In the former case, No. 2 would (in No. 3's view) start "recognizing" rules **W8-W10**, because (to No. 2) the possibility of Li (i.e., "agent i lied") now exists. Such circumscriptive ignorance by No. 3 would most likely be done if an answer was not found by other means first; in the W-Y-D scenario, such a step was not necessary.

One final step is needed to model No. 3's reasoning in the W-Y-D scenario: representing the axioms constructed after No. 1, then No. 2, give their responses. After No. 1 looks and sees No. 2 is black and No. 3 is white, he does not know his color (white); however, he lies. Although both these facts are known only to No. 1 (**W13**), what No. 1 *says* is a common belief (**W14**):

W13 $\neg[S1]P1 \wedge L1$

W14 $W1 \wedge [S0]W1$.

Thus, all agents except No. 1 will not have **W13** in their reasoning system. (Remember that our aim was to distinguish between what agents *say* they believe, and what their actual beliefs are; these two axioms do just that.) Now, No. 2, who blindly believes that No. 1 knows he's white, cannot get any information from No. 1's response. Thus, upon seeing two white dots, No. 2 concludes he doesn't know his color (which is black). However, he always lies, so he cannot say U; thus he says B (although a W response would still fit his deceptive behavior pattern):

W15 $\neg[S2]\neg P2 \wedge L2$

W16 $B2 \wedge [S0]B2$.

An Informal Proof of the W-Y-D Solution

Let us further illuminate No. 3's reasoning. The diagram shows how No. 3 uses his axioms, his sense (and "given") beliefs, and his beliefs about what *others* believe in order to solve the W-Y-D problem. Each of No. 3's beliefs are generated from one of No. 3's 16 axioms, which completely model what he sees and thinks about other agents (e.g. No. 3 knows No. 2 is lying ($[S3]L2$) from **W12** (which states "it is a common belief that No. 2 always lies".) Note first how No. 3 deduces there are only two possible assumptions. After No. 1's response of W, No. 3 reasons about the four combinations between No. 1 possibly lying, and No. 3's possible colors. ($L1 \wedge P3$): possible (if No. 1 lied, No. 3 would have to be white, assuming no other incompleteness on No. 1's part); ($\neg L1 \wedge \neg P3$): possible (if honest, No. 1 *would* say W upon seeing two black dots); ($L1 \wedge \neg P3$): impossible (saying W upon seeing two Bs isn't lying); and ($\neg L1 \wedge P3$): impossible (if No. 3 is white, there's no way No. 1 can deduce he himself is white; thus, if he said W, No. 1 would be lying). Since only the first two options are possible, we prove the first if the second leads to a contradiction:

"First I assume my dot is black, and that No. 1 told the truth ($[S3](\neg L1 \wedge \neg P3)$). Since No. 1 said he's W ($[S3]W1$), it follows No. 2 heard this ($[S3][S2]W1$). Since No. 2 believes everyone is honest, he believes that *No. 1* actually believes he is white ($[S3][S2][S1]P1$). Now, if my dot is black ($[S3]\neg P3$) then No. 2 saw I am black ($[S3][S2]\neg P3$). Also, No. 2 and I see No. 1 is white ($[S3]P1$ and $[S3][S2]P1$). These last three beliefs about No. 2's beliefs (i.e., $[S3][S2]...$) lead to the conclusion that No. 2 believes he is black ($[S3][S2]\neg P2$). This makes sense: if No. 2 believes that I'm black, that No. 1's white, and that No. 1 actually believes he's white - then No. 2 must now believe he's black (see bottom of "contradiction"). Continuing: since I believe that No. 2 always lies ($[S3]L2$), and I heard his response of B ($[S3]B2$), deception axiom **W9** leads me to conclude that No. 2 *cannot* be holding the belief that he's black. In other words, rules **W8-W10** tell me that when someone lies, I must deduce that he actually believes the *negation* of what he *said* he believed. In this case, I thus believe the negation of $[S2]\neg P2$ ($[S3]\neg[S2]\neg P2$) (see top of contradiction). Since a contradiction has been found, my initial assumption must be false, so I deduce the "opposite" of $(\neg L1 \wedge \neg P3)$" - i.e., $[S3](L1 \wedge P3)$.



The Four Classes of Deceptive Behavior

Up to this point, our discussion has focused on generalizing the deduction model of belief to allow for the case where one or more agents exhibit deceptive behavior (making other agents believe what is not true). The behavior discussed so far we call *intentional deception* (or "misleading" behavior), because one deliberately attempts to mislead others. We propose generalizing this notion of deception into a larger framework, adding three other basic types of deception: *unintentional deception* ("misinterpreted" behavior); *unintentional self-deception* ("naive" behavior); and *intentional self-deception* ("irrational" behavior). Our first example illustrates intentional deception:

I went out with Agnes because I wanted Linda to get jealous and thus bake me a cake. Linda thought I was losing interest, so when I came home there was a big cake on the table. It worked.

Here, a deceiver *deliberately tries* to make Linda think he used a certain line of reasoning (e.g. "I'm fed up with Linda so I'm going to start dating other girls"), when in fact such reasoning was not used at all by the deceiver. Yet a slight variation results in unintentional behavior:

I went out with Agnes, a cousin I had not seen in years. Linda thought I was losing interest, so when I came home there was a big cake on the table. I was totally surprised.

Here, Linda assumed the "deceiver" was using the same line of reasoning laid out above, and so took the same action. The difference is that there is *no intent* to deceive here; Linda misinterpreted her boyfriend's behavior as loss of interest. Thus, the most basic forms of deception are (1) intentional: one tries to fool another and succeeds; and (2) unintentional: one does *not* try to fool another, yet succeeds in doing so anyway. The last two forms involve the notion of *deceiving oneself.* In unintentional self-deception, one doesn't try to fool oneself, yet (unfortunately) succeeds:

I did not know the bear's growling was relevant to my petting him, so I petted him. He bit off my hand.

Thus the classification "naive behavior". Another explanation is that one holds a belief, even though one's belief system would deduce the opposite. The key is that you are not cognizant of this latter fact, and hence *not aware that your behavior was illogical* - i.e., you don't know where your reasoning was naive. However, even if one *is* aware of certain deduced beliefs, one might still continue to believe the opposite:

I knew *the bear's growling meant he was angry, but I petted him anyway. He bit off my hand.*

Why one would try to *deliberately deceive oneself* seems difficult to understand. One explanation: *emotion* may interfere on the logical reasoning process; someone wanting to pet a bear may love bears so much that rational danger signs are acknowledged, but disobeyed. Fear can also interfere with rational behavior; one may *know* that the odds of accident are less in planes than in cars, yet still refuse to take a plane for no valid, logical reason. In short, anyone who has performed an activity even though "logic dictated otherwise" has exhibited intentional self-deception. Agents do not always believe deductions that they make and are aware of, and often accept the opposite without the need for justification. People can go even further and "rationalize" their irrational behavior - acknowledging *reasons* for non-justified beliefs *that do not actually exist* in their belief system. These four proposed classes merit closer analysis than this paper provides; we claim all four can be modelled by slightly modifying the basic deception axioms.

Conclusion

This paper has four main conclusions. First, we found that being able to model possible deceptive behavior is an essential ability for problem-solving agents, and constructed an example where reasoning about deception was the only way to solve the problem. Specifically, in our variant of the Wise Man Problem (the Wise-Yet-Deceitful Man Problem) we showed how reasoning about the abilities of "worst-case" and "best-case" deceptive agents allowed agent No. 3 to deduce not only that his color was white, but that agent No. 1 *must* have been lying. Second, we showed how fundamental deception axioms could be represented in Konolige's deduction model of belief, and how reasoning with these axioms can model different types of deceptive behavior. Third, we suggested a taxonomy of deceptive behavior, where standard intentional deception is shown to be related to three other types: unintentional deception, unintentional self-deception ("naivete"), and intentional self-deception ("irrationality"). The final conclusion is that Konolige's notion of incompleteness, and our modelling of different facets of deceptive behavior, are actually quite closely related. We've seen that when an agent does not plan for possible deception by other agents, he is exhibiting a form of *relevance incompleteness* (i.e., ignoring deception axioms can lead to incorrect deductions about other agents). Yet, reversing our point of view, we see that when an agent exhibits non-deceptive incompleteness, another agent (e.g. No. 3) *viewing* this behavior is often deceived if such incompleteness is not explicitly revealed (and, in real-world situations, it often is not). Thus, being fooled because another agent exhibits incompleteness is an instance of unintentional deception (because another agent did not use lines of reasoning *you assumed he was capable of using*). This paper, on the other hand, has focused on how to model both intentional and unintentional deception with explicit axioms. Thus, we feel that adding such explicit deception axioms to Konolige's paradigm, plus using his model's ability to reason with incomplete agents, will result in a more unified model of belief in general.

References

Konolige, K. (1982) Circumscriptive Ignorance. *Proceedings of the Second National Conference on Artificial Intelligence*, Carnegie-Mellon University, Pittsburgh, PA.

Konolige, K. (1984) Belief and Incompleteness. *Center for the Study of Language and Information Report No. CSLI-84-4*, Stanford University, Stanford CA.

BUILDING A COMPUTER MODEL OF LEARNING CLASSICAL MECHANICS *

Jude W. Shavlik
Gerald F. DeJong

Artificial Intelligence Research Group
Coordinated Science Laboratory
University of Illinois at Urbana-Champaign
Urbana, IL 61801

ABSTRACT

A computational model of learning in a complex domain is described and its implementation is discussed. The model supports knowledge-based acquisition of problem-solving concepts from observed examples, in the domain of physics problem solving. The system currently learns about momentum conservation, in a psychologically plausible fashion, from a background knowledge of Newton's laws and the calculus. The background knowledge is consistent with the mathematical abilities of a college student who has been exposed to calculus. In its contribution to machine learning, this research is important for artificial intelligence. From a psychological perspective it demonstrates the computational consistency of a mechanism that may underlie human learning in a complex domain. This work also has implications for computer-aided instruction, in that it advances a learning model for a complicated domain involving both symbolic and numerical reasoning.

INTRODUCTION

In complex domains like physics, people seem to understand general rules best if they are accompanied by illustrative examples. A large part of most physics texts is taken up by examples and exercises. Indeed, there is psychological evidence that a person who discovers a rule from examples learns it better than one who is taught the rule directly [Egan74] and that illustrative examples provide an important reinforcement to general rules [Bransford76, Gick83].

Solving physics problems requires complicated and diverse reasoning. Simply knowing all of the formulae is insufficient. A student must understand the physics behind each formula - knowing how, when, and why it applies. Furthermore, a skilled physicist is able to make quick judgements concerning which aspects of a situation are irrelevant or have negligible effect. We are investigating the process by which a mathematically-sophisticated student acquires these skills.

Physics problem-solving was selected as a domain for several reasons. First, it is concerned with adult learning and so does not suffer the confounding influence of maturation. Second, it forces us to address the very important but neglected problem of combining symbolic and numeric reasoning. Finally, one of us (Shavlik) has had extensive experience as a teaching assistant observing human students struggling through their first physics class.

In its initial state the implementation of our model is capable of performing many of the mathematical manipulations expected of a college freshman who has encountered the calculus. Through tutoring with examples, the system acquires concepts taught in a first semester college physics course; hence the name of the system, **Physics 101**. Newton's laws - which are provided to the system - suffice to solve all problems in classical mechanics (see page 10-1 of [Feynman63]), but the general principles that are consequences of Newton's laws are interesting for their elegance as well as their ability to greatly simplify the solution process. Since the system's physical knowledge rests on the strong foundation of Newton's laws, only its mathematical abilities will limit the physical concepts it can acquire.

Explanation-based learning [DeJong81, DeJong83], is a computer-based knowledge acquisition method that utilizes sophisticated domain representations. In this type of learning a computer generalizes a problem solution into a form that can be later used to solve conceptually similar problems. The generalization process is driven by the *explanation* of why the solution worked. It is the deep knowledge about the domain that allows the explanation to be developed and then extended. We are applying this paradigm to the learning of classical physics. This approach to learning has much in common with [Mitchell85, Silver83, Winston83]. See [DeJong85] for a full discussion.

* This research was partially supported by the Air Force Office of Scientific Research under grant F49620-82-K-0009 and partially by the National Science Foundation under grant NSF-IST-83-17889.

DESCRIPTION OF THE MODEL

Our learning model is inspired by our intuitions concerning the importance of concrete experiences when acquiring abstract concepts. We are implementing this psychologically plausible model as a experiment to test if it is computationally consistent.

Figure 1 contains the model. After a physical situation is described and a problem is posed, the student attempts to solve the problem. We are interested in the process of learning during a successful solution; our attention is currently focussed on learning from a teacher's example. When the student cannot solve a problem, he requests a solution from his instructor. The solution provided must then be verified; additional details are requested when steps in the teacher's solution cannot be understood. We divide the process by which the student understands an example into two phases. First, using his current knowledge about mathematics and physics, the student verifies that the solution is valid. At the end of this phase the student knows that his instructor's solution solves the current problem, but he does not have any understanding of *why* his teacher chose these steps to solve the problem. During the second phase of understanding, the student determines a reason for the structure of each expression in the teacher's solution. Especially important is understanding new formulae encountered in the solution. After this phase the student has a firm understanding of how and why this solution solved the problem at hand. At this point he is able to profitably generalize any new principles that were used in the solution process, thereby increasing his knowledge of classical physics.

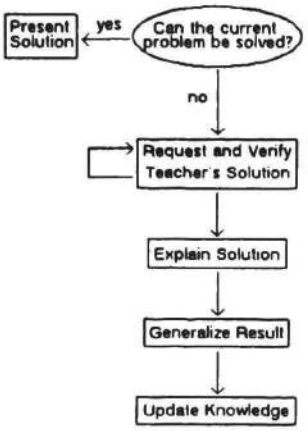


Figure 1. A Model of Learning Classical Mechanics

In our model, physical situations are represented as *worlds*, which comprise a number of *objects*. Figure 2 presents the representation of the generic world, **worldN**, which is used to instantiate specific physical situations. Objects have four measurable attributes; mass, position, velocity, and acceleration. The relationships among these attributes, with respect to symbolic differentiation and integration, are known, although they are not shown in this figure. Only the algebraic relationships are shown.

Newton's second and third laws also appear in figure 2. (Newton's first law is a special case of his second law.) The second law says that the net force on an object equals its mass times its acceleration. The net force is decomposed into two components; the external forces and the internal forces between objects in the world. An external force results from any external fields that act upon an object. Object i's internal force is the sum of the forces the other objects exert on object i. These *inter-object* forces are constrained by Newton's third law, which says that every action has an equal and opposite reaction.

The current implementation of the model learns the physical concept of momentum conservation by analyzing its teacher's solution to a simple collision problem. A fuller description of the learning process can be found in [Shavlik85]. The sample problem is shown in figure 3. In this one-dimensional problem there are two objects moving in free space, without the influence of any external forces. (Nothing is known about the forces between the two objects. Besides their

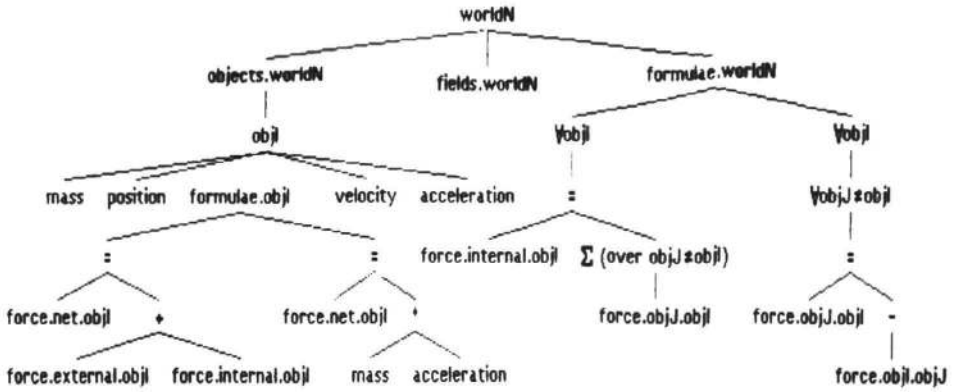


Figure 2. The Generic Representation of a Physical Situation

mutual gravitational attraction, there could be, for example, a long-range electrical interaction and a very complicated interaction during the collision.) In the initial state (**state A**) the first object is moving toward the second, which is stationary. Some time later (**state B**) the first object is recoiling from the resulting collision. The task is to determine the velocity of the second object after the collision.

First, **Physics 101** unsuccessfully attempts to solve the problem using its initial physical knowledge. The system cannot solve this problem, though, as the force exerted on object two by object one is not known. At this point the system requests a solution from its teacher. The solution provided can be seen in figure 4. Without explicitly stating it, the instructor takes advantage of the principle of conservation of momentum, as the momentum (*mass* × *velocity*) of the world at two different times is equated. After that, various algebraic manipulations lead to the answer. In order to accept the answer, **Physics 101** has to verify each of the steps in this solution.

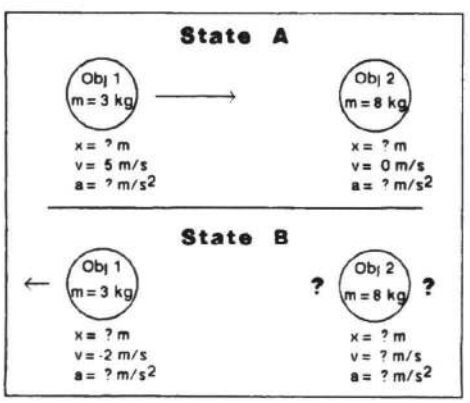


Figure 3. A Two-Body Collision Problem

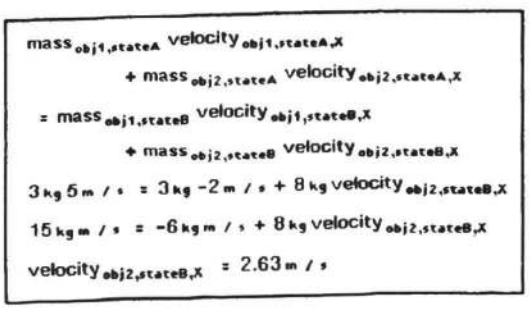


Figure 4. The Teacher's Solution

Four possible classifications of a teacher's solution steps have been identified. Besides being mathematically correct, the instructor's calculations must be physically consistent.

(1) A known formula could have been used: *force* = *mass* × *acceleration* is of this type.

(2) New variables can be defined in order to shorten later expressions.
A formula such as *momentum* = *mass* × *velocity* would fall in this category.

(3) Equations can be algebraic variants of previous steps. The replacement of variables by their values also falls into this category.

(4) The teacher can specify an equation that states a relationship among known variables, yet the system knows of no algebraically equivalent formula. These steps require full justification, which the system does by using its abilities to symbolically reason with calculus. Only the equations falling in this category are candidates for generalization.

The last three steps in figure 4 are easily verified in our model, as they are simple algebraic manipulations. The hard part is determining a physical justification for the first equation in the teacher's solution. Since the two sides of this initial equation only differ as to the state in which they are evaluated, an attempt is made to determine a time-dependent expression describing the general form of one side of the equation. Using its physical and mathematical knowledge, **Physics 101** determines that

$$mass_1\ velocity_{1,X}(t) + mass_2\ velocity_{2,X}(t) = constant_1 \qquad (1)$$

This result validates the first equation in the teacher's solution, as the left-hand side of this equation can be equated for any two times.

At this point the system has ascertained that the teacher's solution does indeed solve the collision problem. In the next step, it tries to understand why the newly-experienced formula is structured the way it is. This formula has been validated - that is, **Physics 101** knows it is mathematically and physically correct but the system must determine *why* the instructor used this equation.

In the initial equation of figure 4, the teacher used four variables to determine the value of object two's velocity. The system analyzes its teacher's solution and detects that summing the two objects' momenta eliminates from the calculation the force each object exerts upon the other, regardless of the details of these forces. (This is a consequence of Newton's third law.) Since each object in a physical situation potentially exerts a force on every other object, in the general case cancelling the net inter-object force upon an object requires summing the momenta of *all* the objects.

Equation 2 presents the result **Physics 101** obtains by extending its instructor's solution technique to a world with an arbitrary number of objects.

$$\sum_{i=1}^{N} mass_i\ velocity_i = \int \sum_{i=1}^{N} force_{external,i}\ dt \qquad (2)$$

This formula says: *The total momentum of a collection of objects is determined by the integral of the sum of the external forces on those objects.* A second problem, which involves three bodies under the influence of an external force, has been solved by **Physics 101** using this generalized result.

Much research on learning involves relaxing constraints on the entities in a situation, rather than generalizing the number of entities themselves. Nonetheless, many important concepts require generalizing number. Explanation-based learning provides a solution to a major problem, namely, how do you know when it is valid and proper to generalize the number of entities? For example, compare the concept of *tripod* with *bicycle wheel*. Both concepts contain a number of repeated components. Suppose a three-legged tripod and a 25-spoked wheel are observed. An explanation-based system can build a general concept for each. The general tripod concept will contain precisely three legs, as any other number of legs is unstable. The general wheel concept, however, will allow a variable number of spokes. The explanation of a component's functionality dictates when it is valid and proper to generalize its number.

CONCLUSION

By analyzing a worked example, the current implementation of **Physics 101** is able to derive a formula describing the temporal evolution of the momentum of any arbitrary collection of objects. This formula can be used to solve a collection of complicated collision problems. Other physical concepts to be learned by the system include work, friction, conservation of energy, simple harmonic motion, and conservation of angular momentum. As these additional concepts are learned, previously-learned concepts will have to be refined. This work will also investigate how the system can learn to estimate which features of a problem can be *ignored* when solving the problem, an important trait possessed by experts.

We have developed a model for learning in a complex domain requiring both symbolic and numeric reasoning. Our approach is knowledge-based: the system requires and applies detailed knowledge about the calculus and Newton's laws. Once a new concept is learned, it is added to the system's knowledge base and is thereby available to help solve future problems and as a stepping stone toward acquiring more difficult concepts. This research contributions to machine learning and psychological modeling. It also has implications for intelligent computer-aided instruction in rich domains where sophisticated learning models are necessary.

REFERENCES

[Bransford76] J. D. Bransford and J. J. Franks, "Toward a Framework for Understanding Learning," in *The Psychology of Learning and Motivation, Volume 10*, G. H. Bower (ed.), Academic Press, New York, NY, 1976, pp. 93-127.

[DeJong81] G. F. DeJong, "Generalizations Based on Explanations," *Proceedings of the Seventh International Joint Conference on Artificial Intelligence*, Vancouver, B.C., Canada, August 1981, pp. 67-70.

[DeJong83] G. F. DeJong, "An Approach to Learning from Observation," *Proceedings of the 1983 International Machine Learning Workshop*, Urbana, IL, June 1983. (Also appears as Working Paper 45, AI Research Group, Coordinated Science Laboratory, University of Illinois at Urbana-Champaign.)

[DeJong85] G. F. DeJong, R. J. Mooney, S. A. Rajamoney, A. M. Segre and J. W. Shavlik, "A Review of Explanation Based Learning," Technical Report, AI Research Group, Coordinated Science Laboratory, University of Illinois, Urbana, IL, 1985.

[Egan74] D. E. Egan and J. G. Greeno, "Theory of Rule Induction: Knowledge Acquisition in Concept Learning, Serial Pattern Learning and Problem Solving," in *Knowledge and Cognition*, L. W. Gregg (ed.), Lawrence Erlbaum and Associates, Hillsdale, NJ, 1974.

[Feynman63] R. P. Feynman, R. B. Leighton and M. Sands, *The Feynman Lectures on Physics, Volume 1*, Addison-Wesley, Reading, MA, 1963.

[Gick83] M. L. Gick and K. L. Holyoak, "Schema Induction and Analogical Transfer," *Cognitive Psychology 15*, (1983), pp. 1-38.

[Mitchell85] T. Mitchell, S. Mahadevan and L. Steinberg, "LEAP: A Learning Apprentice for VSLI Design," Technical Report LCSR-TR-64, Rutgers University, New Brunswick, NJ , January 1985.

[Shavlik85] J. W. Shavlik, "Learning About Momentum Conservation," *Proceedings of the Ninth International Joint Conference on Artificial Intelligence*, Los Angeles, CA, August 1985.

[Silver83] B. Silver, "Learning Equation Solving Methods from Worked Examples," *Proceedings of the 1983 International Machine Learning Workshop*, Urbana, IL, June 1983, pp. 99-104.

[Winston83] P. H. Winston, T. O. Binford and M. Lowry, "Learning Physical Descriptions from Functional Definitions, Examples, and Precedents," Memo 679, MIT Artificial Intelligence Lab, Cambridge, MA, January 1983.

PERSUASIVE ARGUMENTATION IN RESOLUTION OF COLLECTIVE BARGAINING IMPASSES*

Katia Sycara-Cyranski

School of Information and Computer Science
Georgia Institute of Technology

ABSTRACT

In this paper we present a process model that uses past experience in generating arguments of persuasion. We view persuasive argumentation as an instance of problem solving. As such, we employ knowledge organization ideas and problem solving techniques that have been advocated in an analogical view of problem solving. To illustrate our ideas, we use the domain of mediation of labor disputes. Our model is implemented in the PERSUADER, a computer program that gives advice in collective bargaining mediation.

1. INTRODUCTION

Persuasion has been and will continue to be a chief instrument in the conduct of human affairs. Arguments are the means by which persuasion is effected. During persuasive argumentation, an agent, **the persuader** attempts to change the beliefs of another agent, **the persuadee**. In this paper, we present a process model of persuasive argumentation that uses past experience to create new arguments. Our model is influenced by the work of Kolodner and Simpson (1984) on case-based reasoning in problem solving. We use the domain of labor management disputes to illustrate our points.

Traditionally, the psychological literature has treated persuasion as a process of communication (Brembeck and Howell, 1976). In our model, persuasive argumentation is viewed as an instance of **problem solving**. The goal of the persuader as problem solver is to convince the persuadee to accept a particular proposition.** In labor mediation, the mediator is the persuader and the union or company the persuadee. When an impasse is reached in contract negotiations, a mediator is usually called in. The goal of the mediator is to convince the parties to reach a mutually acceptable contract without a strike.

This goal is achieved incrementally through many rounds of persuasive argumentation. In each round, the mediator tries to narrow the disputants' differences with respect to a contract issue, by convincing them to move towards a common value. In such cases, mediators traditionally use well-known persuasive arguments. An example of such an argument is that the adoption of seniority reduces labor turnover. These arguments and the appropriate ways to use them are identified in books on collective bargaining (Herman and Kuhn, 1981, Randle, 1951).

We view these arguments as **plans** that the mediator uses to achieve the goal of changing a party's position with respect to a contract issue. For a plan to be applicable, its preconditions have to be satisfied. The main factor determining the effectiveness of arguments of persuasion is the attitudes and beliefs of the persuadee (Abelson, 1959). The persuader has such a model of the persuadee in mind, to which he is addressing the persuasive arguments. We consider the **persuadee model** as part of the argument plan's preconditions. Another part of the preconditions is the **economic context** of the dispute. Argument plans are known by the mediator and are instantiated when the present case matches their preconditions. The task of the persuader is to decide the applicability of these plans to the situation at hand. To motivate our exposition, we present the following example:

> The Yellow-Jackets textile company involved in a collective bargaining case refuses to grant the workers plantwide seniority for promotions and layoffs. The mediator suggests that seniority improves worker morale, resulting in more efficient plant operation and, consequently, decrease of production cost. The company points out that quite a number of key employees are junior and, during a layoff, they would be the first to go. This would impede the operation of the plant. The mediator, having this additional information, recalls a similar situation where the following solution was found: an exception in seniority for a number of key employees was accepted by the union in exchange for superseniority for union officers and stewards. The mediator proposes this compromise to the company, which agrees.

In this example, the mediator proposes an argument plan that she thinks is suitable to the particular situation. To generate the initial argument, the mediator recalls relevant economic factors, important goals of similar persuadees, and experiences with the same contract issue. Since these three forms of information might come from different mediation experiences, the mediator needs to combine information from the individual available schemata.

---

* This research has been supported in part by NSF Grant No. IST-8317711 and in part by ARO Grant No. DAAG 29-85-K-0023. I would like to thank Janet Kolodner and Bob Simpson for helpful discussions and comments.

**In adversarial argumentation the arguer does not attempt to change the beliefs of the interlocutor (Flowers, et. al., 1982).

constructing the most appropriate combination for the present situation. We call this schema the **argumentation precedent**. In this case the precedent includes the information that efficient plant operation is an important company goal, that seniority improves worker morale leading to worker efficiency, that the economic conditions are recession, and that the majority of textile industry contracts have seniority provisions. The precedent is used as a set of preconditions, against which arguments are tested for applicability.

The next two figures show the conceptual content of the initial argument plan and argumentation precedent for the above example. Space limitations prohibit a full explanation.

THE PERSUADER'S INITIAL ARGUMENT PLAN

```
persuadee: Yellow-Jackets company
issue    : *seniority*
preconditions: argumentation precedent (below)
claim: Increased plant efficiency comes from granting seniority
persuader-goal: Change weight of issue  ;see section on strategies
argument-type: Self-interest            ;see section on convincing power
strength: .7                            ;see section on convincing power
```

Figure O-1

ARGUMENTATION PRECEDENT

```
persuadee-model: goals of the Yelow-Jackets, including their relative
                 importance
economic-context: recession, unemployment in the textile industry,...
```

Figure O-2

---

2. THE OVERALL MODEL

We present the overall process model for persuasive argumetation in Figure 3.

PROCESS MODEL OF PERSUASIVE ARGUMENTATION

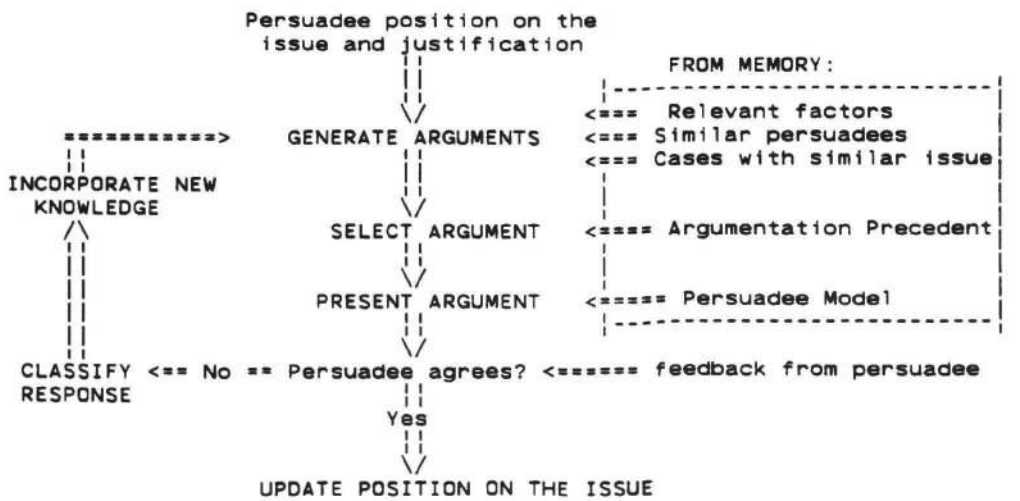


Figure O-3

The input to the argumentation process is the persuadee's position on an issue and the position he needs to be convinced of. In mediation, these correspond to the value of the contract issue that the party has rejected and the mediator's proposal. The first stege in persuasive argumentation is to **generate** potentially applicable arguments using the contract issue as a probe. The most appropriate argument is then **selected** from the retrieved ones. This is

done by using the argumentation precedent, as a set of preconditions against which the potential effectiveness of retrieved arguments is tested. Consider, for example, the argument that **the adoption of seniority for promotions reduces grievances**. The rationale is that seniority is a criterion well-understood by the workers and thus will eliminate potential complaints of unfairness. The strength of this argument for the company depends directly on the importance of reducing grievances as a company goal. Relative importance of goals is included in the argumentation precedent.

Next, the persuader **presents** the selected argument. If the persuadee agrees, the appropriate **update** of the settlement is made, namely that there is agreement on this issue. If the persuadee disagrees, the reasons for the disagreement are analyzed for new information that could alter subsequent argumentation, such as new information about the persuadee's concerns (e.g., the company's concern about key employees), new information about economic factors (e.g., the strength of foreign competition), and corrected inferences about the relative importance of the persuadee's goals. The mediator **incorporate the new knowledge** into the argumentation precedent. In this way, the argument preconditions are dynamically learned as a result of comparing successful and failed applications of the argument. The process of generating potentially applicable arguments is then repeated, testing argument effectiveness against the updated argumentation precedent. A new, more convincing argument is selected for presentation.

3. THE PERSUADEE MODEL

The persuadee model, used during argument generation, selection, and presentation, contains the attitudes and beliefs of the persuadee. These are represented in terms of his collection of goals and their relative importance. Goals of a union or company negotiator are of two types: personal career goals and the goals of the union or company he represents. We represent these goals in **goal trees** (Carbonell, 1979). In the subsequent figure we depict the partial goal tree of a company.

COMPANY GOAL-TREE

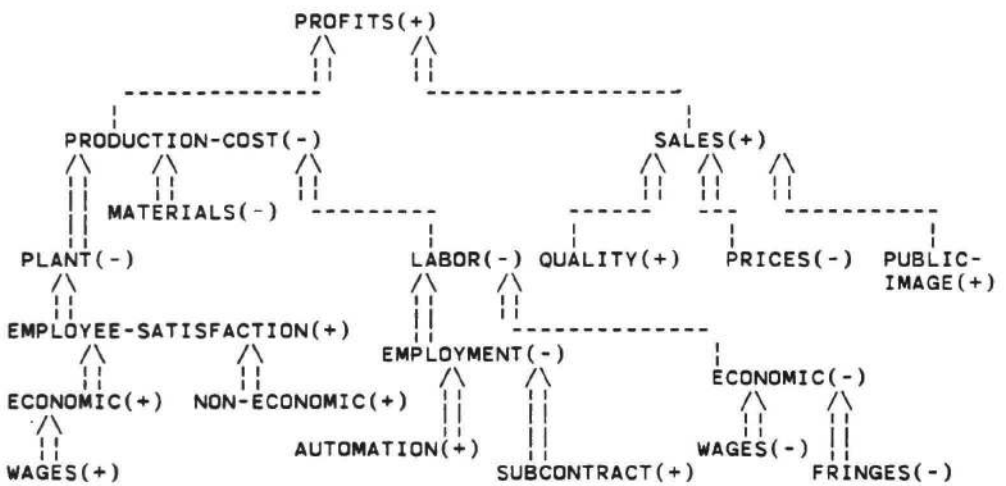


Figure 0-4

The notation for the relationships among goals in the tree is adopted from Spohrer and Riesbeck, (1984). A (+) sign corresponds to the goal of increasing the particular quantity to which it refers, while a (-) sign corresponds to decreasing the quantity. For example, increasing profits, PROFITS(+), which occupies the root of the goal tree, represents the company's highest level goal. The children of a node, connected to it through **support** links, denote the subgoals through which the supergoal is satisfied. For example, profits can be raised, PROFITS(+), by decreasing production costs, PRODUCTION-COST(-), or by increasing sales, SALES(+).

Also included in a goal tree is the relative importance of the party's goals, though for simplicity, this is not shown in the figure. The figure depicts a "prototype" instance (Rosch, 1977) of a company's goals. Goal trees vary with particular negotiators and companies (unions), and the best one possible is needed to construct effective arguments. When a persuader is faced with an unknown persuadee, he can use a prototype goal tree, or a persuadee model by transferring characteristics from the goal tree of a previously encountered and similar persuadee.

4. EFFECTIVE PERSUASION

There are two central issues in selecting the most effective argument plan: first, the persuader's goal, namely in what way does he want to change the persuadee's beliefs; and second, how to do it most convincingly. The first issue involves strategies of persuasion and the second, criteria for the persuasive power of arguments.

4.1 Strategies for argument plan selection

One measure of successful persuasion is the acceptance of the proposed solution by the parties. In mediation, this means the willingness of a party to accept a suggestion regarding a particular contract issue. This willingness depends on the party's assessment of the monetary value of that issue and the issue's importance. Hence, a mediator has two possible goals in convincing a party to accept a previously rejected issue:

1) changing the importance that the party attaches to the issue, or
2) changing the party's assessment of the issue's proposed monetary value

The argumentation strategies used to accomplish these goals determine how the argument plan selection is done. For example, if the persuader's goal is to change the importance accorded an issue by the persuadee, and he chooses to use the first strategy, then a threatening argument plan has to be used. Three argumentation strategies can be used to accomplish the first goal:

(a) indicate possible unpleasant consequences of the present demand
(b) propose alternatives
(c) produce evidence showing that the particular proposal promotes an important goal of the persuadee

To illustrate the first strategy, suppose a union rejects a wage settlement. The mediator tells the union that if the company is forced to grant higher wages, it will become non-competitive and therefore will be forced to lay off workers. If an important union goal is preservation of employment for its members, then the union will abandon its goal of higher wages in order to satisfy its employment goal. Two strategies can be used to accomplish the second goal:

(d) recall a "counterexample" from the persuadee's record of contracts
(e) recall examples of similar unions (companies) having settled for the proposed value or less (more)

To illustrate the last strategy, consider a union's rejection of an increase of 10 cents per worker per hour in health benefits as unacceptably low. The mediator presents contracts signed by the same or other unions which incorporate an equal or lower increase. This argument is effective because perception of "low" or "high" values is determined by **prevailing practice**, namely what settlements similar disputants have agreed to.

4.2 The convincing power of arguments

For persuasion to be effective, the appropriate type of argument has to be presented in each situation. Examining a great number of arguments used in labor mediation, we have identified six categories of argument plan types. They have general applicability, although we will use examples from the mediation domain to clarify their use. We present them in a default ordering of persuasive power (from weakest to strongest):

1) Appeal to universal principle

In using a universal principle, the persuader appeals to some core belief of the persuadee as support for the argument. An example is the argument that a particular wage value does not afford the workers a "decent living standard". Arguments of this type are generally weak, since they appeal to moral principles rather than to the economic realities. However, if "public image" is an important company goal, arguments of this type take on added power.

2) Appeal to "minor standard"

"Minor standards" provide exceptions as a basis for refutation of arguments based on prevailing practice. In mediation, "minor standards" are used as justifications to propose settlements to the employees of one company that differ from settlements within the industry in general. Examples of minor standards include steadiness of employment and hazardous work (Elkouri and Elkouri, 1973).

3) Appeal to "prevailing practice" standard

People's attitudes and goals are strongly influenced by the groups to which they belong. They use the achievements of their peers as a standard with which to compare their situation and expectations. In mediation, this corresponds to the **prevailing practice standard**. Prevailing practice is the most frequently used argument in labor mediation. Its credibility derives from economic reality. A company cannot underpay its employees for fear of loosing them to competitors; a union cannot insist on concessions much above what is given in the industry, for fear of lay-offs.

4) Appeal to precedents as counterexamples

Use of precedents as counterexamples provides a strategy to convince a persuadee that his claim is not as tenable as he would like to think. The power of counterexamples lies in their ability to point out contradictions between the claimed and the actual behavior of the persuadee. Psychological consistency theories (Heider, 1958; Festinger, 1957) give evidence for the persuasive power of counterexamples.

5) Appeal to self-interest

The persuasive power of these arguments depends on the importance of the goal that is claimed to be promoted by the adoption of the persuader's proposal. People will substitute the satisfaction of a lesser goal for a more important one. An example of such an argument is the acceptance by a company of seniority, because it reduces labor turnover, despite the resulting curtailment in management rights.

6) Threats

People want to satisfy their goals, so threatening an important goal of a persuadee is the most effective of arguments. In labor-management disputes, the threat of a strike is the most frequently used and clearly the most powerful argument. However, there are other threats that can be very persuasive, as when a food-processing company's employees threaten to "leak" news of health violations at the plant. The mediator's role here is to convince the company that the employees will carry out their threat and that similar tactics have damaged recalcitrant companies in the past.

5. SUMMARY AND FUTURE WORK

We have presented a portion of the reasoning and domain knowledge necessary in a process model of persuasive argumentation, and given examples from the domain of labor mediation. In this paper, we have concentrated mainly on the task of argument selection. Important factors in this selection are the persuadee model, the argumentation strategies and the convincing power of arguments. Many issues have not been addressed. For example, what is the exact algorithm to construct the argumentation precedent, what is the role of feedback, what is the most appropriate memory organization.

REFERENCES

Abelson, H. (1959). Persuasion.... New York, NY: Springer Publishing Company, Inc.

Brembeck, W. and Howell, W. (1976). Persuasion: A means of social influence. Englewood Cliffs, NJ: Prentice-Hall, Inc.

Carbonell, J. G. (1979). Subjective Understanding: Computer Models of Belief Systems. Doctoral dissertation. Yale University Research Report #150.

Elkouri, F. and Elkouri, E. (1973). How Arbitration Works. Washington, DC: The Bureau of National Affairs, Inc.

Festinger, L. (1957). A theory of cognitive dissonance. Stanford, Calif: Stanford University Press.

Flowers, M., McGuire, R. and Birnbaum, L. (1982). Adversary arguments and the logic of personal attacks. In W. G. Lehnert and M. H. Ringle (Eds). Strategies for Natural Language Processing , Hillsdale, NJ: Lawrence Erlbaum Associates.

Heider, F. (1958). The Psychology of Interpersonal Relations. New York, NY: Wiley.

Herman, E. and Kuhn, A. (1981). Collective Bargaining and Labor Relations. Englewood Cliffs, NJ: Prentice-Hall, Inc.

Kolodner, J. and Simpson, R. (1984). Experience and Problem-Solving: A Framework. In Proceedings of the Sixth Annual Conference of the Cognitive Science Society. June 28-30, Boulder, Colorado.

Randle, W. (1951). Collective Bargaining: Principles and Practices. Cambridge, Mass.: The Riverside Press.

Rosch, E. (1977). Classification of real-world objects: origins and representations in cognition. In Johnson-Laird, P. N. and Wason, P. C. (Eds.) Thinking, Readings in Cognitive Science. Cambridge: Cambridge University Press.

Spohrer, J. C. and Riesbeck, K. (1984). Reasoning-driven Memory Modification in the Economics Domain. Yale University Research Report #308.

The interaction of lexical expectation and pragmatics in parsing filler-gap constructions [1]

Michael K. Tanenhaus and Laurie A. Stowe
University of Rochester
and
Greg Carlson
Wayne State University

Sentences with embedded questions in which there is a long-distance dependency between a noun phrase or filler and an empty category or gap require the parser to hold onto the filler in memory until the position of the gap can be located. Gap location is complicated by local ambiguity. For example, the filler "horse" in the fragment in (1a) could be the direct object of the verb "raced" as in (1b) or it could be the object of the preposition "toward" as in (1c). In the former case the gap follows the verb, in the latter case it follows the preposition.

(1) a. The sheriff wasn't sure which horse the cowboy raced.....
b. The sheriff wasn't sure which horse the cowboy raced __ down the hill.
c. The sheriff wasn't sure which horse the cowboy raced desperately past__.

Fodor (1978) proposed three possible models of gap detection and filling. According to a "first resort" model the parser posits a gap following any verb which can be used transitively. A parser using a first resort strategy will make the right decision with sentences such as (1b) but it will incorrectly assume a gap follows the verb in the embedded sentence in sentences such as (1c). Alternatively the parser could adopt a "last resort" strategy. A last resort parser would assume a gap analysis only when a mandatory argument was missing or when the end of a sentence was reached and the parser still had a filler which had not been assigned a grammatical role. Thus a last resort parser would not garden-path on sentences such as (1c), but it would miss the gap on the first pass through sentences such as (1b).

Fodor rejected both of these models on the basis of sentences such as (2) and (3). The gap after "about" in sentence (2) seems to come as a surprise because the filler "book" has been assigned as the direct object of the verb "read" as the first resort model would predict. However, the first resort model makes the wrong prediction for sentence (3) in which readers do not seem to be garden-pathed by the possible gap following "sing".

(2) Which book did the teacher read to the children about__?
(3) What did the teacher sing about__?

As an alternative, Fodor proposed a lexical preference model in which the verb in the embedded sentence determines whether or not the parser posits a gap. Gaps are posited following verbs which are normally used transitively and thus "expect" an object, but not following verbs which are normally used intransitively. Some evidence for the lexical expectation model was provided in a recent paper by Clifton, Frazier, and Connine (1984) who demonstrated that sentences with filler-gap constructions are understood more quickly when the correct interpretation is congruent with the lexical preference of the verb.

Stowe (1984) has recently suggested an all-resorts model as a fourth possibility. According to this model, a gap is postulated following a verb that is optionally transitive but the filler is taken as the object of the verb only if it is plausible. If the filler is not a plausible object, the gap analysis is rapidly rejected.

We conducted three experiments to test predictions made by these models using sentences in which normally transitive or intransitive verbs (e.g., "raced" and "hurried", respectively) were placed in sentences in which the filler was either the object of the verb in the embedded sentence (early gap) or the object of a preposition that ended the sentence (late gap). Transitive and intransitive expectation verbs were chosen from the norms in Connine et al. (1984). Two fillers were chosen for each sentence type: a noun which was a plausible object of the verb and a noun which was an implausible object. Example materials for a transitive and an intransitive expectation verb are presented in (4) and (5). The two nouns in parentheses are the plausible and implausible fillers, respectively. Early gap sentences are presented in (4a) and (5a) and late gap sentences in (4b) and (5b).

(4) a. The sheriff wasn't sure which (horse, rock) the cowboy raced__down the hill.
    b. The sheriff wasn't sure which (horse, rock) the cowboy raced desperately past__.

(5) a. The district attorney found out which (witness, church) the reporter asked__about in the meeting.
    b. The district attorney found out which (witness, church) the reporter asked anxiously about__.

Experiment 1 used 32 sets of materials similar to those illustrated in (4) and (5). Sixteen sets were constructed using intransitive expectation verbs and sixteen sets were constructed with transitive expectation verbs. Plausibility of the filler was crossed with the position of the gap resulting in four sentences for each transitive and intransitive expectation verb. These sentences were counterbalanced across four presentation versions. The test sentences were intermixed with ungrammatical and gramatical fillers sentences, including some ungrammatical sentences with filler-gap constructions. Twenty-four University of Rochester volunteers served as subjects. Their task was to decide whether or not they understood each sentence. The sentences were presented visually on a CRT.

The logic of the experiment was as follows. The early gap model predicts that sentences with late gaps should be more difficult to understand than sentences with early gaps, because readers should initially assume that the filler is the object of the verb. In contrast, the late gap model assumes that sentences with early gaps will be more difficult to understand than senences with late gaps because the readers will initially miss the gap following the verb. The all-resorts model predicts that late gap sentences with plausible fillers should be more difficult than late gap sentences with implausible fillers because readers will have chosen the

early gap analysis for the plausible fillers (recall that plausibility refers to the plausibility of the filler as an object of the verb). In contrast the lexical expectation model makes the same prediction as the early gap model for the transitive expectation verbs and the late gap model for the intransitive expectation verbs.

Table 1 presents the percentage of sentences judged comprehensible for each of the conditions. For the intransitive expectation verbs early gap sentences were less frequently understood than late gap sentences. Plausibility of the filler affected comprehension of the early but not the late gap sentences. For the transitive expectation sentences, early gap sentences more more likely to be understood than late gap sentences. In addition, plausibility of the filler affected both the early and late gap sentences suggesting that with transitive expectation verbs readers attempted to associate the filler with the verb even for the late gap sentences. Surprisingly, late gap sentences with plausible fillers were understood more easily than late gap sentences with implausible fillers, suggesting that readers do not find it easier to recover from an early gap misanalysis when the filler is implausible.

TABLE 1

	Verb Expectation			
	Transitive		Intransitive	
	Early Gap	Late Gap	Early Gap	Late Gap
Plausible Filler	76%	65%	66%	85%
Implausible Filler	58%	51%	45%	81%

The results of Experiment 1 strongly support Fodor's lexical expecation model. They suggest that readers attempt to associate fillers with transitive expectation verbs but not intransitive expectation verbs. Experiments 2 and 3 examined the on-line processing of the sentences used in Experiment 1 in order to test these predictions. In both of these experiments subjects read sentences one word at a time pressing a response key when they were ready to read the next word. After approximately 30% of the sentences, the subject was asked to repeat the sentence aloud. The subject was also required to answer a true-false question following about 30% of the sentences. The prediction was that reading times would be longer following implausible fillers than following plausible fillers if subjects attempted to associate the filler with the verb. For late gap sentences plausibility effects should only obtain with transitive expectation verbs. For early gap sentences plausibility effects should be observed earlier for transitive expectation verbs. Experiment 2 used early gap sentences and Experiment 3 used late gap sentences. Tables 2 and 3 present the mean reading time per word for Experiments 2 and 3, respectively. There were 33 subjects in Experiment 2 and 28 in Experiment 3.

TABLE 2

Condition	Position in Sentence: Verb (raced)	Preposition (down)	Object 1 (the)	Object 2 (hill)
Transitive Plausible	506	515	502	738
Transitive Implausible	569	569	541	758
Plausibility effect	63	54	39	20
Intransitive Plausible	554	561	536	734
Intransitive Implausible	538	576	555	766
Plausibility effect	-16	25	19	32

TABLE 3

Condition	Position in Sentence: Subject1 (the)	Subject2 (cowboy)	Verb (raced)	Adverb (quickly)	Preposition (towards)
Transitive Plausible	484	498	496	555	683
Transitive Implausible	472	494	569	548	716
Plausibility effect	-12	-4	73	-7	33
Intransitive Plausible	445	487	517	651	724
Intransitive Implausible	458	484	513	648	734
Plausibility effect	13	-3	-4	-3	10

The results of Experiments 2 and 3 confirm the predictions made by the lexical expectation model. For early gap sentences a plausibility effect was found for sentences with transitive expectation verbs beginning with the verb, whereas the effect was weaker and did not begin until later for sentences with intransitive expectation verbs. For late gap sentences there was a plausibility effect for the sentences with transitive expectation verbs but not for the sentences with intransitive expectation verbs. Taken together the results of these experiments clearly demonstrate that lexical expectation controls initial gap detection and gap filling in the processing of sentences with long-distance filler-gap dependencies. This is in accord with the general claims about the importance of lexical structure in parsing made by Ford, Bresnan and Kaplan (1983).

References

Clifton, C., Frazier, L., & Connine, C. (1984) Lexical expectations in sentence comprehension (1984). Journal of Verbal Learning and Verbal Behavior, 23, 696-708.

Connine, C., Ferreira, F., Jones, C., Clifton, C., & Frazier, L. (1984) Verb frame preferences: Descriptive norms. Journal of Psycholinguistic Research, 13, 307-319.

Fodor, J.D. (1978) Parsing strategies and constraints on transformations. Linguistic Inquiry, 9, 427-474.

Ford, M., Bresnan, J., & Kaplan, R. (1983) A competence-based theory of syntactic closure. In J. Bresnan (Ed.), The mental representation of grammatical relations (pp 727-796) Cambridge, Mass: MIT Press.

Stowe, L. (1984) Models of gap-location in the human parser. Dissertation distributed by the Indiana University Lingustics Club.

1. This research was supported by NSF grant BNS-8217378.

Predicting Conversational Reports of a Personal Event

Yvette J. Tenney
Bolt Beranek and Newman Inc.
Cambridge, MA 02238

This study addresses the question of how conversational reports are generated. Although previous research has been concerned with how one response is generated in response to another (Grice, 1975; Schank, 1977), a fundamental question that has not been investigated is how a speaker decides what to say when given the freedom to introduce a number of topics. The problem of deciding what to say is also relevant for written compositions (Hays and Flower, 1980). Yet little is known about the selection of ideas for discourse.

Kintsch (1980) has described the task of generating discourse as a problem of searching through memory for subject matter that meets the constraints of subject, audience, and discourse type. According to this view, factors that affect the salience of events in memory should be important in predicting topic choices. Furthermore, the influence of memory factors should be particularly clear in the case of conversations, where topic decisions have to be made rapidly and spontaneously.

What determines the salience of events in memory? Gamst (1982) has speculated that "interests, needs, concerns, and point of view" are important, while Schank (1980) has postulated that "unusual, important, painful, or otherwise notable" aspects of an event are likely to be accessible. Both of these factors were investigated in the present study.

The purpose of the study was to examine how speakers select items from long-term memory to produce conversational reports about real events. More specifically, the goal was to see if it was possible to predict what a particular speaker would say on the basis of individual concerns and the particular outcome of events. To meet these needs, it was necessary to find conversational topics that were predictable in advance, likely to be met with varying degrees of concern, and associated with a range of possible outcomes. Conversations about the birth of baby were selected because they met these requirements.

The study was carried out in three stages. In the first stage, couples who were expecting a baby filled out a questionnaire about their concerns one month prior to the birth. In the second stage, participants tape recorded phone conversations in which they announced the arrival of the baby. The third stage consisted of a follow-up questionnaire.

It was expected that both prior concerns and the outcome of events would affect the selection of topics. The first hypothesis was that subjects would be more likely to mention topics of high prior concern than topics of low concern. A number of investigations have shown that a subject's schema, or point of view, influences what is encoded and recalled about narratively depicted events (e.g., Anderson, 1978). Work on mental models (Gentner &

Stevens, 1983) has shown that memory for physical phenomena (e.g., the trajectory of a ball) is shaped by naive beliefs. The present study extends this line of research by examining the effects of prior concerns on the reporting of personal events.

A second hypothesis was that subjects would be more likely to mention topics that had an unusual outcome than topics that had an ordinary outcome. Several lines of evidence support this prediction. Robinson (1980) found that subjects were able to retrieve memories of unusually pleasant or unpleasant events more quickly than memories of neutral events. More generally, research has shown that subjects pay attention to aspects of an event that cannot be inferred on the basis of prior knowledge (e.g., Gibbs & Tenney, 1980).

Method

Subjects

Twelve couples participated in the study. Seven were expecting their first child, four their second, and one their fourth.

Materials

Materials consisted of a prenatal and a postnatal questionnaire concerning seventeen topics related to labor and delivery (e.g., difficulty of labor, father's role, use of the birthing room), the baby (e.g., name, sex, appearance), and activities during the postpartum period (e.g., breastfeeding, rooming-in, sibling visits). The prenatal questionnaire consisted of twenty-five questions (e.g., How concerned are you about possible discomfort to the mother during labor and delivery? 1-5 scale). The postnatal questionnaire consisted of twenty-eight questions on the same topics (e.g., How did the degree of discomfort to the mother during labor and delivery compare to what you had expected? 1-5 scale).

Procedure

One month prior to the mother's due date, the experimenter administered the prenatal questionnaire separately to father and mother and showed the couple how to record their calls. The postnatal questionnaire was administered one month after the birth.

Results

Questionnaire Results

Prenatal. Responses on the prenatal questionnaire were converted into the numbers 1 to 5, where 5 indicates the greatest concern. A subject's concern for a topic was categorized as high if the subject's score was above the mean for fathers or mothers, respectively, and low if it was below.

Postnatal. The outcome of each of the topics was categorized as unusual or ordinary. Responses were converted to the numbers 1 to 5, where 5 indicates the most favorable outcome and 1 the least favorable outcome. Outcome scores

were categorized as unusual if either of the extremes (1 or 5) was selected.

Assignment of Topics to Expectation x Outcome Categories

Each of the seventeen topics rated by a subject on the pre- and postnatal questionnaires was assigned to one of four expectation x outcome categories: high concern-unusual outcome, high concern-ordinary outcome, low concern-unusual outcome, low concern-ordinary outcome. Degree of concern was determined by responses on the prenatal questionnaire, while unusualness was determined by responses on the postnatal questionnaire.

Frequency of Mention of Topics

The recorded phone conversations yielded 90 separate reports. Each report was scored for mention of each of the seventeen topics by the investigator and a second, independent rater. In order not to bias the coding on the basis of outcome, both negative and positive statements about a topic were counted (e.g., mention of use as well as non-use of drugs counted for the topic of natural childbirth).

Analysis of Selection Rules

For each subject, the likelihood of mentioning each of the seventeen topics was defined as the proportion of conversations in which the subject mentioned the topic. Thus a subject who mentioned natural childbirth in three out of six conversations had a likelihood of mention for that topic of .50. The likelihoods for all the topics that fell into the same expectation x outcome category for a particular subject were averaged together. Table 1 shows the likelihood of mentioning topics in each of the four expectation x outcome categories, averaged across the nineteen subjects who had conversations.

Table 1

Average Likelihood of Mentioning Topic

High Concern		Low Concern	
Unusual Outcome	Ordinary Outcome	Unusual Outcome	Ordinary Outcome
.446	.302	.287	.206

The likelihood data were analyzed in a two-way analysis of variance with concern (high, low) and outcome (unusual, ordinary) as within subject factors. The results revealed a significant main effect of concern, $F(1,18) = 6.70$, $p < .05$, a significant main effect of outcome, $F(1,18) = 5.22$, $p < .05$, and no interaction between concern and outcome, $F(1,18) < 1$, $p > .05$.

Discussion

This study was concerned with the question of what makes something interesting or worthy of mention. Given all the possible topics one might mention in describing an event, what determines which ones will be reported? The answer turns out to depend upon both the speaker's prior concerns and on

the events themselves.

The first hypothesis, that subjects would be more likely to mention topics of high than low concern, was supported by the data. Although the prenatal questionnaire was not designed to identify specific childbirth models, it was expected that subjects' models would be reflected in their responses. For example, one possible model is that labor is like an illness, requiring medical intervention. A contrasting view is that labor is a physical challenge that can be met by adequate preparation. It was expected that subjects with physical challenge models would be more likely than subjects with sickness models to give a high importance rating to the topic of natural childbirth.

Why were topics of high concern mentioned more frequently than topics of low concern?. A reasonable explanation is that subjects had more elaborate, models for those aspects for which they indicated strong concerns. A highly differentiated model would allow for more elaborate encoding of the event, by focussing attention on aspects that would otherwise be ignored. Consider, for example, the detailed description of labor techniques given by one of the mothers in the study who fell into the category of high concern-ordinary outcome on the topic of natural childbirth.

> "Well, I sort of invented my own breathing technique as I went along. [Oh great, everybody does it their own way.] You know, I couldn't count one-two-three-four and then pause and then one-two-three-four. So I did sort of, something sli..., slightly different, whatever you know worked for me at the time."

The second hypothesis, that subjects would be more likely to recall topics that had an unusual than an ordinary outcome, also received confirmation from the data. There are several possible explanations for this finding. First, it is adaptive for subjects to allocate attention to the unusual, since the routine can be inferred, by default, from prior knowledge (Gibbs & Tenney, 1980). Secondly, unusual events may be intrinsically salient because they involve strong affect. Robinson (1980) showed that the intensity, though not the direction, of affect associated with an event predicted retrieval time on a test of autobiographical memory. Finally, there are the demands of good conversation. Listeners expect the speaker to be maximally informative, which suggests a focus on the novel (Grice, 1975).

Thus, although there was a tendency for speakers to emphasize areas of personal concern in their choice of topics, they did talk informatively about aspects that had not been of particular concern when the outcome was unusual. For example, two subjects who differed in the importance they attributed to early bonding gave similar descriptions of the bonding period that they were permitted in the hospital. The subject who had been concerned about bonding said,

> "They gave me the baby almost immediately. They do that. I mean it's wonderful. We had her almost an hour and a half. We took pictures and everything and it was wonderful,"

while the subject who had been indifferent reported,

"They put her immediately, you know, her skin to my skin and they put a blanket over the two of us. [Aha] He was taking pictures and everything and... [Was it right on your tummy?] Oh Yeah, they put her right on me. [Oh nice] And um, you know, so it was really good."

To conclude, the study showed that it is possible to predict which speakers will talk about which general topics in naturally occurring conversations, given knowledge of their prior concerns and of what actually happened. However, there was considerable variety in how topics were handled. For example, the topic of the name was handled with humor ("Well, it was either 'Robin' or 'Blackbird'"), the topic of the baby's sex was treated with suspense ("It's...a baby!"), and finally, the topic of pain was handled philosophically, ("I just guess it dawned on me that there was only one way out and I had to do something. They weren't going to do anything for me"). It is this creative aspect of the reporting of personal events that poses the biggest challenge to our understanding.

Finally, the generation of ideas for discourse should be examined in other domains. Further research may show that the same memory processes apply to personal reports of weddings, trips, accidents, job offers, major purchases, and winning the lottery.

References

Anderson, R. C. Schema-directed processes in language comprehension. In A. Lesgold, J. Pelligreno, S. Fokkema, and R. Glaser (Eds.), Cognitive psychology and instruction. New York: Plenum, 1978.

Gamst, G. Memory for conversation: Toward a grammar of dyadic conversation. Discourse Processes, 1982, 5, 33-51.

Gentner, D., & Stevens, A. Mental Models. New York: Erlbaum, 1983.

Gibbs, R. W., & Tenney, Y. J. The concept of scripts in understanding stories. Journal of Psycholinguistic Research, 1980, 9, 275-284.

Grice, H. P. Logic and conversation. In P. Cole and J. L. Morgan (Eds.), Syntax and semantics, Vol 3: Speech acts. N.Y.: Academic Press, 1975.

Hayes, J. R. & Flower, L. S. Identifying the organization of writing processes. In L. W. Gregg & E. R. Steinberg (Eds.), Cognitive processes in writing. Hillsdale, N.J.: Lawrence Erlbaum, 1980.

Kintsch, W. Psychological processes in discourse production. Institute of Cognitive Science Technical Report No. 99, University of Colorado, Boulder, Colorado, 1980.

Schank, R. C. Rules and topics in conversation. Cognitive Science, 1977, 4, 421-441. 9, 71-82.

Schank, R. C. Language and memory. Cognitive Science, 1980, 4, 243-284.

Thematic Knowledge, Episodic Memory and Analogy in MINSTREL, a Story Invention System *

Scott R. Turner
Michael G. Dyer
Artificial Intelligence Laboratory
Computer Science Department
University of California
Los Angeles, CA 90024

Abstract

This paper examines the process of storytelling and story invention. It focuses on the use of themes, episodic memory, analogical mappings, planning and literary goals. A computational model of storytelling is presented and its implementation as the program MINSTREL is discussed. MINSTREL contains an episodic memory of stories and themes and uses these memories along with knowledge about the world of King Arthur's knights to invent interesting new stories.

Introduction

This paper is an overview of the MINSTREL project. MINSTREL is a computer program (under development) that models human story-telling behavior. MINSTREL tells stories in the King Arthur domain.

Previous Work

Meehan's TALESPIN program [Meehan 76] told stories about the lives of simple woodland creatures. The thrust of TALESPIN was planning; the process of telling a story involved giving some character a goal and then watching the development of a plan to solve that goal. In this example from [Meehan 76], John Bear has been given some initial knowledge about the world and a goal to satisfy his hunger. The resulting story:

> *John Bear is somewhat hungry. John Bear wants to get some berries. John Bear wants to get near the blueberries. John Bear walks from a cave entrance to the bush by going through a pass through a valley through a meadow. John Bear takes the blueberries. John Bear eats the blueberries. The blueberries are gone. John Bear is not very hungry.*

This story illustrates the strengths and weaknesses of TALESPIN. First, the story is very believable and logically consistent. This is a reflection of TALESPIN's strong planning component and illustrates (at least intuitively) that planning is an important component of storytelling. On the other hand, the story seems pointless and somewhat boring because TALESPIN is a planner operating on the level of character planning and not at any higher level. (Meehan realized this and added a component that forced the story to follow a pre-canned template.) So an interesting negative result of Meehan's work was that story invention based on planning could not produce interesting or memorable stories, thus requiring an ad hoc, pre-canned template. This result showed that a general theory of themes, morals or interesting situations is needed.

Story Themes

What is a story theme and what makes them interesting to the reader? Dyer [Dyer 1983] pointed out that one class of themes (or morals) consist of planning advice and some explanation for its validity, often provided in terms of the negative consequences that result when the advice is not followed. In "The Fox and the Crow" the theme is "Don't trust flatterers"; this says to avoid assuming a goal at the behest of someone else if they do so by appealing to some facet of you in an exaggerated way. Some examples of story themes are:

1. "Be kind to strangers": Choose plans that benefit others.

   POOR WOMAN AND THE PRINCE

   An old, poor woman finds a stranger who has been robbed and beaten. She nurses him back to health and discovers that he is a prince, and she is richly rewarded.

2. "Never say die:" Don't abandon goals when no solution is apparent.

   CANCER CURE

   Ethan suffered from terminal cancer and was in constant pain. The doctors said he might live another six months, but he despaired and decided to commit suicide. He went down to the drugstore to buy some poison and when he got there the druggist told him that a new miracle drug had just been discovered that would cure his cancer.

Story themes cover a broad range of planning advice at many levels. If we assume that these story themes arise from a knowledge structure the author posseses, then what can we say about these structures?

Dyer [Dyer 1983] suggested Thematic Abstraction Units (TAUs) as a knowledge structure to represent thematic knowledge. TAUs are characterized by adages and represent planning advice that can be couched either as a typical planning failure (i.e., "Throwing good money after bad") or as a rule for making a choice during planning (i.e., "Don't throw rocks if you live in a glass house.").

* The work reported here was supported in part by a grant from the Keck Foundation. The first author was also supported in part by a fellowship awarded by Hewlett Packard in conjunction with the AEA.

The theory of TAUs has two parts. The first says that TAUs represent important planning advice. The themes presented above represent rules that can apply at various points in the planning process. For instance, "Never say die" is advice to the mechanism that decides when to extinguish goals. Knowing to keep a goal alive even in the face of a lack of plans may avoid a failure. Thus these themes also deal with avoiding plan failures.

The second part of the theory of TAUs claims that TAUs organize episodic memory so that an episode containing a TAU is likely to cause a reminding of a related episode that contains the same TAU. For instance, the story:

FELLOWSHIP

John realized that he only had money for two more quarters of college, and decided to drop out immediately to look for a job. He went to Murphy Hall to fill out the proper forms and discovered that he had been awarded a fellowship.

tends to remind people of the previous story, "CANCER CURE" because they both embody the same TAU.

MINSTREL

Inspiration: *"Every good play must have a well formulated premise."* – Lajos Egri [Egri 1960]

How does an author decide upon a theme in the first place? How does a theme come to mind? Often the selection is based upon an interaction between the author's values and his personal experiences. We believe that the author recalls an episode from memory and its related theme and uses this as the basis for a story. This recall is based on the many inputs an author has: his immediate goals, the environment around him, things he has lately been told, and so on.

MINSTREL models this behavior by accepting as input conceptualizations which are then used to index memory. If the conceptualization recalls an episode with a related TAU, then this is used to start the story telling process. If not, then MINSTREL awaits more input.

The Use of Theme in Plotting

Once a theme has been selected, how is it used to guide the story telling process? The theme of a story is instantiated in the events that make up the story (the plot). Some of these events are very crucial to the theme, others are less important or entirely irrelevant. The theme is used to build the initial skeleton of events for the story. How is this derived from the theme?

One way to derive a skeleton plot from a theme is to build it. A theme of the form "Planning behavior X is good" lends itself to a plot of the form:

1. Give character A behavior X. Have character A use behavior X in goal/plan situation G. Have character A receive some good effect as a result of G.

This scheme has a number of problems. First, it produces a limited number of plots, and makes learning new plots very difficult.

Second, the plot as built by these rules is too general. Filling in such a plot requires "guessing" appropriate behaviors, roles, etc., to fill in the plot.

Finally, it creates some difficult planning problems. Given some behavior X (i.e., "Never say die" – don't abandon goals when out of plans), generating a situation that requires this behavior is a potentially expensive problem since it involves undirected search through a large space of goals, plans and world knowledge. A better method for deriving a skeletal plot from a theme is to recall a story involving the same theme and borrow the plot (or plot elements) from it.

The Role of Episodic Memory in Plotting

At the time the author is searching for a plot, he has available to him his theme, his initial inputs, and at least one recalled episode (the one that prompted the theme). This material may be enough to provide him with further remindings. Even if it does not, he has one episode to borrow a plot from.

This method avoids the above objections to plot rules. The number of plots is limited only by the number of relevant remindings, the episodes recalled provide a great deal of material for later use by the story teller, the planning problem is already solved and new plots can be learned by generalizing and mutating the recalled episodes. Given an episode that illustrates the theme, borrowing a plot from the episode involves mapping the pieces of the episode into the story domain.

Plotting Through Analogical Mappings

There are two facets to this analogical process. First, we must maintain our story knowledge through the mapping. That is, we must remember how the elements of the episode relate to and exemplify the theme. Secondly, we must map world features of the recalled episode into our story domain.

Maintaining the story knowledge can be done by retaining the high level memory structures (such as TAUs) that index the remembered episode. Maintaining the TAU through the mapping retains the abstract knowledge about the theme that is embodied in the episode. Through the TAU we can identify important components of the theme, such as the planning failure and the negative consequence.

Performing an analogy on world features requires recognizing elements of the episode in terms of their functions, and mapping these into elements with equivalent functions in the new domain. Thus, a car

in a 20th century story might map into a chariot in the medieval domain because they both function as vehicles, and a job interview might map into an audience with the king, because they both are examples of social interactions with a possible social superior.

From Plotting to Story

Given a theme and an initial plot, what more is left to do before we have a story? One task left is fleshing out the plot. A character at this stage might be represented as a role (i.e., KNIGHT) and have no other features.

Instantiating features of a plot requires planning and world knowledge. Suppose, for example, that we need to develop a situation where our main character, a KNIGHT, faces a physical risk. Using our world knowledge about KNIGHTs we consider some typical KNIGHT-ly plans that can result in physical damage, namely, fighting a monster.

Author-level Goals

Another task at this stage of inventing a story is fulfilling author-level goals. The importance of author-level goals to storytelling has already been pointed out by Dehn [Dehn 1981]. Author-level goals embody "good writing techniques," goals like building suspense, character development, creating pathos, etc. There are two facets to such goals. First, we must know when to activate these goals. Second, we must have some techniques or plans for achieving them.

MINSTREL looks at a class of author-level goals that are intended to create an emotional response in the reader. One example of this kind of goal is creating suspense. MINSTREL has the following rules that indicate *when* it is useful to try and create suspense:

1. When you are developing a scene in which a character is about to achieve a solution to a crisis goal.
2. When you are developing a scene in which a character is at physical risk.
3. When you are developing a scene in which two characters have competing goals.

There are two ways to increase the drama or suspense of a scene. The first is by increasing the importance of the main character's goal in that scene. The second is by increasing the reader's anticipation of the solution to the goal.

Increasing the goal importance:

1. Make the goal more important to the main character.
2. Create goal conflicts.
3. Increase the rewards and punishments.

Increasing the reader's anticipation:

1. Eliminate favorable solutions to the goal.
2. Make the main character use a dangerous plan.
3. Insert a secondary incident between the assumption of the goal and its conclusion.
4. Make the goal important to the reader.

Above we showed how a scene with a KNIGHT fighting a monster might arise when we needed the KNIGHT to be at physical risk. When this scene is created, the suspense rules fire (it is a scene involving physical risk) and suspense techniques are applied to make the scene more dramatic. For instance, if the technique "Insert a secondary incident" was used, the KNIGHT would lose his sword in the midst of the battle, forcing him to suddenly scramble for another weapon.

A Control Structure for Storytelling

To this point we have discussed the process of storytelling as a linear one: discover a theme, build a plot, flesh out the plot, fulfill literary goals. The storytelling process is not at all linear.

To see why this is, consider the example above where we constructed a scene with a KNIGHT fighting a monster in the course of fleshing out another scene. Constructing this fight scene would cause us to be reminded of other memorable fight scenes. This reminding brings with it a wealth of imagery that can be used to further flesh out the current fight scene, and it may also recall a new theme and plot. Suddenly the author, concerned with fleshing out a scene, has become reminded of another story, and finds himself thinking about the theme and plot of that story. He has suddenly shifted back to the stage of storytelling concerned with theme and plot.

How is this situation to be handled in a storytelling program? These remindings are the essence of creativity; what we would like to do is jump from concern to concern as they arise. This suggests a task agenda-style control mechanism. This is very similar to the agenda mechanism Lenat used in his AM program [Lenat 1976]. It is very interesting that two programs concerned with creativity in vastly different domains should use the same control structure, and it is likely that this reflects something important about the nature of creativity. The ability to be interrupted with fortuitious remindings and the ability to be able to react to this by jumping from task to task, level to level, seems to be crucial to creativity.

Episodic Memory

In this section we briefly discuss how the process of storytelling affects episodic memory.

Once finished, and at its various stages of development, the story is a conceptual representation of an episode, much like any episode in episodic memory. These episodes are added to episodic memory, to be available during future storytelling sessions as an index to themes, sources for fleshing out and so on.

Similarly, fragments of stories may be built and discarded during the course of fleshing out a story. If

our story initially contains a scene where a knight finds a sword in a tree, only to have that scene excised in order to raise the suspense in a subsequent scene, the scene where the knight finds the sword is not discarded. It remains, indexed in memory, where it may later be recalled.

Thus episodic memory becomes a rich storehouse not only of the finished stories, but also the partially completed stories and story fragments that were generated along the way. Writers become better writers by writing, a process that builds their store of episodes, and provides them with a greater range of material for future writing endeavors.

Interactions and Examples

In this section we give some examples of the material presented above and show how it might interact to produce a story. In this case, episodic memory already contains a conceptual representation of the following story (a simplified synopsis of "It's a Wonderful Life"):

A banker is kind to many of the members of his community and loans them money when they are in need. He then misplaces some money and is threatened with loss of his bank. At the heighth of his despair his friends arrive with the money needed to save the bank.

This story is indexed by TAU-GOOD-DEEDS-REWARDED, which also contains the advice "Be generous to others and they may return the favor." Episodic memory also contains information about bankers in addition to knowledge about the King Arthur domain.

The input to MINSTREL is: A banker died.

In this particular case, the input recalls "It's a Wonderful Life," not because it involves TAU-GOOD-DEEDS-REWARDED, but because they are both interesting incidents involving bankers (for a discussion of indexing by content see [Kolodner, 1980]). When the banker incident is recalled MINSTREL decides to tell a story about generousity (specifically, TAU-GOOD-DEEDS-REWARDED). In addition, the mention of "death", an important goal to the author, brings the goal of P-HEALTH to mind.

The theme has been selected, so now plotting gets priority. Looking at the recollected episode, the plotter finds this skeleton:

Person X has skill Y. X uses Y to help Z. Z is grateful to X. (1) Calamity befalls X. Z saves X from calamity (because of 1) X is grateful to Z.

We now map the first help situation into the King Arthur domain. There is no analog of banking in the King Arthur domain, so we abandon the direct analogy. Instead we attempt to recall an interesting help situation – using the P-HEALTH goal we've been thinking about – and recall an event where a hermit helped to heal a knight.

Using this recollection, we build the first help scene and instantiate the roles of the two main characters:

An old woman of the woods knows how to heal. The old woman heals a knight. The knights is grateful to the old woman. Calamity befalls the old woman. The knight saves the old woman. The old woman is grateful.

As new scenes are created, tasks fire which check the scene for consistency. One of these tasks notes that the knight is healed without being in need of healing, and so creates a scene where the knight is injured (and again, creates a scene which places the old woman near the knight).

A knight is travelling through the woods and encounters a troll. The knight defeats the troll but is injured. An old woman of the woods is picking berries and finds the knight. The old woman heals the knight. The knights is grateful to the old woman. Calamity befalls the old woman. The knight saves the old woman. The old woman is grateful.

Note that these new scenes have just been created backwards from the order in which they will be told. In general, the storytelling process skips around through the story, not necessarily attacking it in temporal order. The planning component in particular sometimes works backwards, determining things which should have happened, and sometimes forward, determining what will happen. This reduces the cognitive load. Telling a story only requires a theme and key scenes. The entire story does not need to be solved at once.

We generate the second help situation in a similar way:

An old woman of the woods knows how to heal. A knight fights a troll and is injured. The old woman is out picking berries and finds the injured knight. The old woman heals a knight. The knight is grateful. The old woman is attacked by a dragon. The knight saves the old woman. The old woman is grateful.

At this point the previously mentioned rule for building suspense in scenes involving physical danger fires on the scene of the old woman being attacked by the dragon. One of the rules for building suspense is:

> Build suspense in a physical fight scene by making the protagonist physically weaker.

Unfortunately, the old woman is already weak (knowledge we have about the role of being an old woman), so this rule fails. Another rule is:

> Build suspense by eliminating all solutions to the problem except the one that will work.

In this case, one plan for saving one's life is to run

from the threat. MINSTREL eliminates this by trapping the old woman in her cottage:

An old woman of the woods knows how to heal. A knight fights a troll and is injured. The old woman is out picking berries and finds the injured knight. The old woman heals a knight. The knight is grateful. The old woman is trapped in her cottage by a dragon. The knight saves the old woman. The old woman is grateful.

Now focus turns to the knight saving the old woman. Again this is recognized as a scene worthy of suspense, so the above rules fire again. This time, the first rule:

> Build suspense in a physical fight scene by making the protagonist physically weaker.

can fire. The knight is made physically weaker by making him injured. The planner now looks for a way that he might be injured. One way is if he was involved in battle. Another way is if a previous healing wasn't completely effective. This is chosen and we have:

An old woman of the woods knows how to heal. A knight fights a troll and is injured. The old woman is out picking berries and finds the injured knight. The old woman heals a knight. The knight is grateful. The old woman is trapped in her cottage by a dragon. The knight, still weak from his old wounds, saves the old woman. The old woman is grateful.

At this point, knowledge of style and storytelling generation devices come into play.

> **Galahad was out in the woods one day when he encountered a troll. He did battle with the troll, and killed the troll, but was himself injured and laid near death on the forest floor. He was discovered by Glinda, an old woman of the woods, who applied her healing art to him and helped him recover from his wounds. Galahad thanked the old woman and promised to return the favor someday.**
>
> **Later, Glinda found herself trapped in her cottage by a dragon. She was terrified. Just then Galahad rode up. Still weak from his wounds, he battled the dragon and killed it, saving Glinda. Glinda wept her thanks on Galahad's shoulder.**

Future Work

MINSTREL currently generates a few simple stories in the King Arthur domain. Up until now, most work has been done on the planning and episodic memory components of MINSTREL. The heuristics for suspense have also been developed, but these parts have not yet been combined. There are a number of interesting topics left to address. Some of these are:

1. A large class of TAUs were recognized as arising out of planning failures. What other classifications of themes and thematic memory structures are there?
2. MINSTREL currently does nothing with multiple themes or plots. How can two themes be woven together into a single story?
3. MINSTREL ignores a large number of literary goals. Characterization, for instance, is recognized as being very important for good writing.

We believe that MINSTREL will provide a good environment within which to examine these more complex issues.

Conclusions

MINSTREL is a program to model human story telling behavior. We have seen that the basis for story-telling is a cycle of writing-rewriting that is fueled by information derived from the theme of the story and from fortuitious remindings that occur during the process of fleshing out the story. This process involving episodic memory and reminding-driven construction is one of the basic processes behind human creativity.

References

Dehn, Natalie, Story Generation after TALE-SPIN, in *Proceedings of the 7th IJCAI*, pp. 16-18.

Dyer, Michael G., *In Depth Understanding*, The MIT Press, Cambridge, MA, 1983.

Egri, Lajos, *The Art of Dramatic Writing*, Simon and Schuster, New York, 1960.

Kolodner, Janet L., *Retrieval and Organizational Strategies in Conceptual Memory: A Computer Model*, Lawrence Earlbaum Associates, Hillsdale, New Jersey, 1984.

Lenat, Douglas B., *AM: An Artificial Intelligence Approach to Discovery in Mathematics as Heuristic Search*, Stanford AI Lab, Memo AIM-286 (Ph.D. Dissertation), Stanford University, Dept. of Computer Science, 1976.

Meehan, James R., *The Metanovel: Writing Stories by Computer*, Technical Report 74 (Ph.D. Dissertation), Yale University, Dept. of Computer Science, 1976.

Schank, Roger C., *Dynamic Memory*, Cambridge University Press, Cambridge, 1982.

Spatial inferences and discourse comprehension

Karl F. Wender and Monika Wagener
Technische Universitat Braunschweig

ABSTRACT

Theories of discourse comprehension and memory for text usually assume a propositional format in which information is stored. In agreement with the work on mental imagery we argue that information from texts may also be remembered in a spatial representation. Inference processes with spatial relations depend on the format of the mental representation. In two experiments we employed a priming technique to show spatial properties of mental representation. The first one using narratives failed to yield positive results. The second experiment using spatial descriptions supported the hypothesis. Decision times in a priming task were dependent on spatial distances. The relationship between inference processes and the form of the mental representation is discussed.

This research investigates the interaction between the mental representation of a text and inference processes that operate on this representation. The main idea is that inference processes depend to a certain extent on the form of the mental representation.

Aspects of mental representation of discourse have been investigated by many researchers in recent years. One question is whether information is stored in a propositional network or in a more analogous format. For information from discourse it is not obvious how objects and events described in a text are mentally represented. It is conceivable that the type of representation depends on the reading task or the goal of the reader.

Most models of text comprehension propose a propositional format. On the other hand, if someone has to make spatial judgements of some sort it might be more efficient to construct a spatial mental representation. Some authors, most notably Johnson-Laird (1983), argue that mental representations of texts often take on an analogous, spatial form.

The notion that subjects may use different strategies depending on the nature of the task is supported by results from Ohlsson (1984). He was able to show that in a spatial reasoning task subjects switched between two procedures called series formation and elimination. In present terms series formation would imply the construction of a spatial mental representation whereas elimination would correspond to inference processes using a propositional representation.

Graesser (1977) showed the influence of a spatial representation on sentence comprehension. It is our goal to show that such spatial representations are also constructed during reading of discourse. This is an intuitively plausible hypothesis. However, it is not that easy to find empirical support.

Experiments by Mani and Johnson-Laird (1982) and by Ehrlich and Johnson-Laird (1982) give experimental support to these notions. However, their experimental materials were not stories but rather simple descriptions of the spatial relation between objects. An experiment by Black, Turner, and Bower (1979) demonstrates the role of point of view in narrative comprehension. This result also suggests a spatial representation.

The present study uses a different experimental technique. A procedure which has been successfully used by several authors is the priming technique (Meyer and Schvaneveldt,1971). A priming effect has already been observed in studying representations of sentences and discourse. Ratcliff and McKoon (1978) showed that in a recognition task words from the same sentence primed each other more than words from different sentences. Guindon and Kintsch (1984) used this method to demonstrate that readers had formed macropropositions during reading. In a recent study by McNamara, Ratcliff, and McKoon (1984), the priming technique was applied successfully to spatial representations of maps with routes and cities.

Experiment 1

In our first experiment, we try to show that the priming method works also on spatial representations of narrative texts.

Method

We employed a procedure that has been used by McKoon & Ratcliff (1981) to prove the influence of instrumental inferences.

In our experiment, subjects read short stories that were presented sentence by sentence on a CRT screen. Subjects were instructed to read each

sentence and to press a button when they felt that they had understood the sentence. Following the last sentence a single word appeared on the screen. Subjects had to decide as fast as possible whether this word had been mentioned in the preceding text or not. Decision time was measured.

Subjects

Subjects were 24 students of the Technische Universität Braunschweig. They were paid DM 10,-- for their participation.

Material

Each subject read 24 short stories plus 6 additional ones for warming-up purposes. Half of the 24 stories were experimental stories. The remaining 12 were distractor stories. Each experimental story consisted of five or six sentences describing everyday events. In each story three relevant objects were mentioned. The stories were constructed in such a way that two of the objects were spatially close to each other. The third one was further away. An example is the following story about "Hans smoking at school: After school, Hans is smoking in the classroom. This, of course, is not allowed and he must be very careful. Suddenly a teacher comes around the corner. Hans just manages to throw the cigarettes under the table. Previously he had hidden the cigarettes behind the curtain. With an innocent smile he looks to the table."

The three relevant objects are the table, the curtain, and the cigarettes. At the end of the story the cigarettes and the table are close to each other whereas curtain and cigarettes are farther apart. On the other hand, the text was written in such a way that the propositional or network distance between table and cigarettes is equal to the distance between curtain and cigarettes.

In the last sentence in the story the last word mentioned is table which served as the close prime. To investigate the difference in priming effect between close and distant primes we wrote a second version of each text in which the last sentence was replaced by. "With an innocent smile he looks to the curtain." This sentence should activate the concept curtain which served as the distant prime. Thus, decision time should increase.

To control for semantic associations between table and cigarettes as well as between curtain and cigarettes, there were two corresponding versions in which the roles of table and curtain were interchanged. Thus, curtain was the close prime and table was the distant prime.

Procedure

Texts were presented sentence by sentence on a CRT screen. Subjects were instructed to step on a pedal when they had understood a sentence. Then this sentence was replaced by the next one. Subjects were instructed to imagine each described scene as vividly as possible. After the final sentence the target word appeared and subjects had to respond by pressing a yes or a no button. They had to decide whether the target word had been presented in the text.

We expected a priming effect of the concepts contained in the sentence immediately preceding the target. In particular we looked at the priming effects of table vs. curtain with respect to cigarettes. If subjects form a mental representation like a semantic network then the decision times for the target word cigarettes should not differ under the two priming conditions. If, on the other hand, subjects construct a spatial representation, then the closer prime should have a facilitating effect on

the target. Thus the decision time following the prime table should be shorter compared to the time following the prime curtain.

Results and Discussion

The results of the experiment were not as expected. Mean decision times were 1103 ms for the close prime and 1088 ms for the distant prime condition. This difference is not significant and does certainly not support the hypothesis of a spatial representation.

There may be several reasons why the experiment failed. Perhaps our texts did not give enough clues for a spatial representation. Or it may be that subjects had to read too many stories. Furthermore, it is possible that the experimental task did not neccessarily require a spatial representation. Some subjects reported that they instead compiled a list of concepts mentioned in the text and then compared the target with this list.

Experiment 2

Method

To avoid the problem of too many stories we to used a priming technique that does not need stories as foils. Furthermore, we changed the material completely, bebause stories like those in Experiment 1 always contain many things more than just spatial relations.

An additional crucial change concerned the learning task for the subjects. They now learned short texts that described spatial configurations of five common things. To make sure that all subjects formed a similar spatial representation they had to arrange real objects according to each description. The learning phase of the experiment was followed by a priming phase and a verification test.

In the priming phase subjects had to decide whether pairs of objects belonged to the same configuration or not. Following the spatial representation hypothesis the decision time should depend on the distance between the two objects of the judged pair.

In a following the verification test subjects had to decide whether sentences of the form "object A is in relation R to object B" were true with respect to a specified configuration. Again decision time should depend on the distance between objects.

Subjects

Another group of 24 students participated in this experiment.

Materials

Eight short texts were written describing the spatial relationship between five objects. For example: "Vegetables: The paprika is in front of the cucumber. Left to the cucumber is the potato. In front of the potato is the onion. The tomato is in front of the onion."

Priming phase. From each configuration two pairs of words are of special interest. These are called the close pair and the distant pair according to the distances between the two respective objects. The spatial relationship between the two words of each of these pairs was not explicitly stated in the text but could be inferred by the reader. The predictions for these pairs were as follows. If subjects construct a spatial representation then the priming effect should depend on the spatial distance between prime and target within each pair. Thus, decision time should be shorter for the close pair than for the distant pair. If subjects do not use a spatial representation then decision times should be approximately equal.

There were two versions of each configuration in which the close pair

and the distant pair were interchanged to control for the strength of semantic associations between the words of a pair. For example, in the second version of the text "vegetables" onion - paprika was the distant pair and cucumber - tomato was the close pair.

Verification test. Subjects read sentences that either agreed with the description or that contained a contradiction. For example "The onion is to the left of the paprika." would be a correct statement although it was not given in the original description. We call such a probe sentence an inference. Furthermore, we distinguish between close inferences, like the one given above, and distant inferences. A distant inference is for example: The potato is behind the tomato. The distance between these objects is twice as long as the distance between the objects of the close inference Following the spatial hypothesis distant inferences should take longer to verify than close inferences.

Procedure

In the first part of the experiment subjects had to learn the configuration extensively. In the second part we measured first the reaction times for the item recognition and then the verification times for complete sentences. The instruction given in the beginning did not explain the priming procedure and the verification test. The instruction only mentioned that some retention test would follow.

Learning phase. Each subject read eight short descriptions plus three additional ones for warming-up. Two different groups of subjects (Group A and Group B) were assigned to the two balanced versions of each story. Both groups got the same objects but with different configurations. Each description consisted of four sentences printed on a card. Subjects were instructed to read the text on the card as long as they wanted and to memorize them. Then they got a box containing one exemplar for each of the five objects. With these exemplars subjects had to build up the described configuration on a table in front of them.

Priming phase. The priming procedure took place at a CRT screen connected to a PDP-11. First, the title of one of the descriptions was presented on the screen. Subjects were instructed to visualize as vividly as possible the corresponding configuration. When they were ready they stepped on a pedal. The title disappeared and a fixation point was presented for 1 s. Then the first word was shown for 250 ms. There was an interstimulus intervall of 250 ms before the second word followed. This word remained on the screen until the subject responded. Subjects were instructed to rest their index fingers on two touch keys. They had to respond with yes or no by lifting one of their index fingers because reaction time was measured through the interrupt.

Verification Test. A similar procedure was used as in the priming test. Each trial began with the presentation of a title. Then eight test sentences followed. Half of these were correct and half were incorrect probes. The foils used the same objects, but placed them into wrong spatial relationships.

Results

Priming data. Decision times were analysed by a 2x2 analysis of variance. The factors were type of probe (close vs. distant) and groups of subjects. The mean decision times for the close and distant probes were 752 ms and 850 ms respectively. The mean decision time for the stated probes was 730 ms which is very close to the close pair. However, the stated probes

were not included in the analyse because they contained different words. The difference between the distant probe and the close probe was significant ($minF'(1, 19) = 7.38$, $p < .02$). The difference in reaction times between the two groups of subjects did not reach significance. There was no significant interaction between the type of probes and the groups of subjects.

The percentages of correct responses under all conditions were very high. The mean percentages for the close pairs and the distant pairs were both equal to 98.5 %. The percentage was 98.9 % for the stated probes. The mean percentage of correct answers for all pairs including the foils was 98.8. There were no significant differences.

Verification test. The verification times were not as expected. Averaged over both groups of subjects the verification of the close inference took the longest time vs. groups of subjects. The stated probes were verified faster than both inference types.

The percentage of correct responses was generally lower than in the priming data. But again there were no significant differences between the types of probes or the groups of subjects. The percentages were: 91.7 for the close inference, 91.2 for the distant inference, 94.0 for the stated probes, and 94.1 averaged for all probes including the foils.

Discussion

The main result of our second experiment is the priming effect. The close pair was recognized significantly faster than the distant pair. This result is in accordance with our hypothesis that subjects have built a spatial representation. However, we have to interpret this result with some caution.

To begin with, we have not proven that a spatial representation is the only one that can explain our data. What our data imply is that subjects did not use a representation corresponding to a propositional encoding of the original description. In such a propositional network the distances within the close and the distant pair would be equal. Therefore, under the usual assumptions about the spread of activation, the decision times should be approximately equal. On the other hand, we cannot reject the possibility that subjects have formed a propositional net. It might be that they elaborated this network in a way that would also lead to a prediction of our data.

The following consideration is an indirect argument against a propositional representation. From the work of Guindon and Kintsch (1984) one would expect that subjects, when using a propositional representation, would also form macropropositions. A macroproposition which would help to remember a particular configuration could be a proposition stating that the five objects belonged to the given configuration. If a subject had formed this macroproposition it would be unlikely that she or he did not use it in the recognition task. However, from such a macroproposition one would predict equal recognition times for the close and distant pairs. This expectation is contradicted by our data.

If subjects have in fact built a spatial representation then we assume that this representation has properties that are in some sense functionally equivalent to those of an actual scene. This means that the space is continuous and distances are determined by an Euclidian metric. There are, however, other possibilities.

The first one is that the space may be discontinuous. In that case

points in the space might resemble something like a lattice with unit distances in between. The distance between two points may then be determined by the number of intervening objects. In such a representation the relation between the objects of the close pair would be inherently given as unity. This would not be true with respect to the distant pair. In the hypothetical lattice the distance between the two objects would then be three. Our data do not allow us to distinguish between the continuous and discontinuous case.

Another possibility would correspond to the findings of McNamara et al. (1984) about the representation of cities on a map. Distances between cities may be either determined by the Euclidian metric or by the length of a route between them. It is conceivable that in our experiment subjects also have placed objects in their mental representation on something like a route. This route did not exist in the actual scene but could be constructed from the sequence of objects and their relationships mentioned in the text. In fact, some of our subjects reported that in the beginning they had built such a sequence. However, it appears that this must have been an intermediate state. Because the length on such a route and the number of intervening objects are equal for the close pairs and the distant pairs, the recognition times should be equal. Hence, our data allow us to reject this possibility for the representation of the configurations.

Coming to the inference processes it seems that the relationships between the objects of the close and distant pair that were not mentioned in the text have been integrated into the mental representation by the reader. Hence, there should be no difference between the stated relations and the implicit relation of the close pairs. This was in fact the case although comparison is not easily made because the probes involved different words.

Verification test. With respect to the verification task, our data are inconclusive. The only significant effect was an interaction that is difficult to interpret. Furthermore, the data show very large standard deviations (1500 ms and above). The explanation we like to offer is that the verification task was very difficult.

The reason might be an overload of working memory. It is possible that subjects needed almost the total capacity of their working memory for the image of the given configuration. In that case it might be impossible to execute all necessary processes: to comprehend the probe sentence, to hold the image, and to compare the meaning of the sentence with the imagined configuration. A capacity problem would arise, for example, when the reader has constructed a spatial image for the configuration and when he or she tries to construct a second one from the sentence to be verified. It is difficult to operate with two images at once. Hence, parts of the first configuration gets lost and the reader has to start over again. This enlarges the verification times and especially the standard deviations and masks other effects.

General Discussion

As said at the outset, it is our goal to show that readers construct spatial representations during discourse comprehension. Furthermore, we want to show that the inference processes on spatial representation are different from those on a propositional format. We have not yet reached this goal completely. For reasons discussed above the first experiment failed to produce the priming effect. In the second experiment our subjects got an additional task. Besides reading the texts they had to build the configurations with real objects. Obviously, this task has visual properties. Hence, we cannot claim that subjects have constructed their spatial mental

representations exclusively during reading. However, it appears that subjects relied on the spatial representation when responding to the priming task. The time between prime onset and target onset used in our second experiment is short enough to ensure that subjects could barely start a process that was under their attentional control. We assume that our results are effected by an automatic process as discussed by Ratcliff and McKoon (1981).

The results are one step into the desired direction. What we have shown is that the priming technique is capable to reveal spatial information when people remember spatial descriptions. This was in doubt after our first experiment. The next step has to be a change in the experimental procedure such that subjects only read texts without any additional visual task intervening between reading and probing. The question then is whether the priming effect, dependent on distances, will be obtained again.

Let us consider now the relationship between the priming technique and inference processes. The task for the subject may be paraphrased as 'Did the configuration include the following two objects?'. In a propositional representation an inference process might consist of at least the following subprocesses: First, a search process must find nodes representing the two objects. Then the process has to find out whether the objects are linked to a proposition representing the configuration. Finally, the process has to decide whether the relations labeling the links can be interpreted in such a way that the question can be answered affirmatively. As suggested by the results of Ratcliff and McKoon (1978) this inference process takes less time when two concepts are linked together in the same proposition during reading. But this process should not depend on spatial distances unless the representation has some sort of spatial properties.

In a spatial representation the question can be answered simply by searching for the two objects. If they are found in the mental spatial representation then, by this virtue, they belong to the configuration. That is, in a spatial representation the include relation is inherently contained. The medium of representation is the spatial relation. The inference process then reduces to a search process. If this search process has equivalent temporal properties as the visual search the amount of time needed should depend on spatial distances in the mental representation. This was shown by our second experiment.

References

Black, J. B., Turner, T. J., & Bower, G. H. (1979). Point of view in narrative comprehension, memory, and porduction. Journal of Verbal Learning and Verbal Comprehension, 18, 187-198.

Ehrlich, K. & Johnson-Laird, P. N. (1982). Spatial descriptions and spatial continuity. Journal of Verbal Learning and Verbal Behavior, 21, 396-406.

Graesser, A. C. (1977). Sentence memory and comprehension. Unpublished doctoral dissertation, University of California, San Diego.

Guindon, R. & Kintsch, W. (1984). Priming macrostructures: Evidence for the primacy of macropropositions in the memory for text. Journal of Verbal Learning and Verbal Behavior, 23, 508-518.

Johnson-Laird, P. N. (1983). Mental models. Cambridge: Cambridge University Press.

Mani, K. & Johnson-Laird, P. N. (1982). The mental representation of spatial descriptions. Memory & Cognition, 10, 181-187 .

McKoon, G. & Ratcliff, R. (1981). The comprehension processes and memory structures involved in instrumental inference. Journal of Verbal Learning and Verbal Behavior, 20, 671-682.

McNamara, T. P., Ratcliff, R., & McKoon, G. (1984). The mental representation of knowledge acquired from maps. Journal of Experimental Psychology: Learning, Memory, and Cognition, 10, 723-732.

Meyer, D. E. & Schvaneveldt, R. W. (1971). Facilitation in recognizing pairs of words: Evidence of a dependence between retrieval operations. Journal of Experimental Psychology, 90, 227-234.

Ohlsson, S. (1984). Induced strategy shifts in spatial reasoning. Acta Psychologica, 57, 47-67.

Ratcliff, R. & McKoon, G. (1978). Priming in item recognition: Evidence for the propositional structure of sentences. Journal of Verbal Learning and Verbal Behavior, 17, 403-417.

Ratcliff, R. & McKoon, G. (1981). Automatic and strategic priming in recognition. Journal of Verbal Learning and Verbal Behavior, 20, 204-215.

Cognitive Processing Strategies for Complex Addition

Keith F. Widaman, David C. Geary, Pierre Cormier, & Todd Little
Department of Psychology
University of California
Riverside, CA 92521

Abstract

Simple and complex addition problems were presented for true/false verification to 22 subjects across two times of measurement to test the general model for simple and complex addition proposed by Widaman, Cormier, & Geary (1985). Models fit to average RT data revealed that subjects were processing complex problems columnwise, beginning with the units column. Column sums seemed to be obtained through an incrementing process, and subjects exited problems as soon as a column error was encountered. Group level models were the same across complex problem types and for both times of measurement. However, individual level analyses suggested that nearly half of the subjects used a different processing strategy to obtain column sums for the second time of measurement. Results support the multi-staged model proposed by Widaman et al. (1985), but individual level results suggest that information processing models developed from group data may not represent the processing strategies used by all subjects, or the same subjects at different times.

Information processing models for cognitive addition in adults include counting based, direct memory access, and memory network models (Parkman & Groen, 1971; Ashcraft, 1982). Groen & Parkman (1972) found that the true sum of simple addition problems, and the smaller addend (MIN) of these problems predicted nearly equal amounts of reaction time (RT) variance. Groen & Parkman (1972) interpreted these results as suggesting that all simple addition facts were retrieved directly form memory, but a counting based (MIN) procedure was used with memory retrieval failure.

Ashcraft & Battaglia (1978) found that the true-sum-squared was the best predictor of simple addition RT. The authors concluded that there was a network representation for simple addition that resembled a square, printed addition table with entry nodes for the digits 0-9 on two adjacent sides. The point of intersection of the two entry nodes represented the correct sum. Entry nodes were assumed to be unevenly spaced; the network was "stretched" in the region of larger sums. Thus, larger sums resulted in longer vector distances and therefore longer RTs.

The network approach to cognitive arithmetic has successfully predicted performance in adults (Ashcraft & Stayzk, 1981), children (Ashcraft & Fierman, 1982), and for production tasks (Ashcraft, Fierman, & Battaglia, 1984). However, this model has been developed from simple addition and multiplication tasks (Stayzk, Ashcraft, & Hamann, 1982) and has not been generalized beyond complex addition tasks for sums greater than 30 (Ashcraft & Stazyk, 1981).

Widaman, Cormier, & Geary (1985) have proposed a general model for cognitive addition. The proposed model (for verification tasks) is multi-staged, and begins with encoding the type of problem, and then

encoding the addends (columnwise) beginning with the units place. The units addends are summed either through a network of stored facts or procedurally. If there are more addends in the units column, the process recycles, and an additional addend is encoded and summed onto the sum of the first two addends. This process continues until all digits in the units column are summed. The obtained unit sum is compared with the stated sum, and a decision as to correctness is made. If the sum is incorrect the problem is terminated and a "no" response is given. If the units sum is correct, and there are more columns to be added, the process "loops back." The digits for the next column are encoded, then summed, etc. The looping is continued until all columns are completed, or until one column sum is incorrect, at which point the problem is self-terminated.

Widaman et al. (1985) verified this model for simple and complex problems, and found that complex (2-column) problems were done columnwise using a memory network to obtain column sums. Further, a carry operation was highly significant, and subjects tended to exit the problem as soon as an error was encountered. However, the most complex problems used by Widaman et al. (1985) contained 2 double-digit addends or 3 single-digit addends. The present study sought to replicate and extend the findings of Widaman et al. (1985) by testing their model with extremely complex problems. Further, to assess the reliability and generalizability of this model, the stability of processing models across two times of measurement and individual differences (Hunt & McLeod, 1978) in processing strategies were also examined.

Method

Thirty undergraduates were presented with 240 addition problems in a one hour session. This set consisted of 30 correct and 30 incorrect randomly selected: 1) two single-digit addends, 2) two double-digit addends, 3) two triple-digit addends, and 4) two four-digit addends. These problems were readministered to 22 of the 30 subjects one week later.

For incorrect problems, column sums differed from correct sums by ± 1, or ± 2. For complex problems the placement of the error was evenly distributed among the individual column sums, and each problem contained only one error. Similarly, a carry operation (column sum 10) was necessary for about one-half of the cases (per column), and approximately one-half of these cases contained an error. The presentation of problems was random except for the following constraints: 1) there were no more than four consecutive presentations of any one problem type, 2) there were no more than four consecutive presentations of correct or incorrect problems, and 3) same sums or addends were never presented consecutively.

Problems were presented for true/false verification in column form on an Apple II equipped with a cognitive testing station. Subjects pressed one response key to indicate if the presented problem was correct and a second key if the presented problem was incorrect. Reaction time and response accuracy were recorded.

Results and Disscussion

Models for addition were fit to average reaction time data using multiple regression techniques. Models fit to RT data included Parkman & Groen's (1971) five counting based models, Ashcraft's (1982) true-sum-squared parameter, and Widaman et al.'s, (1985) reinterpretation of Ashcraft's (1982) model. Specifically, Widaman et al., (1985) argued

that Ashcraft's memory network would be better represented by the sum of the digits squared (SSQ), rather than the square of the summed digits, because each digit squared would represent the seperate vectors on the search through the addition table. For complex problems (2-4 column), models were fit according to the stages outlined by Widaman et al. (1985). Specifically, models were fit in columnwise fashion, with a carry parameter specified if the preceding column sum exceeded 9. A self-terminating parameter was also tested. The self-terminating parameter would represent an executive process (Sternberg, 1982) process, whereby subjects would "exit" the problem as soon as an incorrect column sum was encountered. Finally, the above (e.g. MIN) simple addition parameters were fit for each column sum.

The error rate was 5.8 percent and 5.6 percent for the first (T1) and the second (T2) time of measurement, respectively. Less than one percent of the RT data were deleted as outliers (exceeding 2.58 S.D. of individual RT means).

Parkman & Groen's (1971) MIN (smallest digit) model provided the best fit for simple addition RTs. The MIN model with the truth parameter denoting an intercept difference between true and false problems provided the best fit for both T1 and T2. Consistent with previous findings (Groen & Parkman, 1972; Ashcraft, 1982; Widaman, et al., 1984), false problems were processed more slowly than were true problems. Across sessions parameter estimates decreased, suggesting that subjects were becoming more efficient as a function of practice. Overall, these results suggest that subjects used an incremenenting model for simple addition problems.

Results revealed that similar columnwise parameters provided the best fit to the RT data for all (2-4 column) complex problems, across sessions. The specific column sums were best predicted by the sum of the column (unitsum, tensum, etc.), followed by column MIN, and the Widamam, et al. (1985: sum of digits squared: SSQ) parameter, respectively. Parameter estimates, again, decreased across times of measurement, suggesting that subjects were becoming more efficient. The column sum parameter suggests that subjects were using some type of incrementing process to obtain column sums, however the speed of the incrementation makes it unlikely that it was language based (Landauer, 1962).

The carry/self-terminating (carryst, car10st, carhunst) parameters represented a carry operation if the preceeding column sum was correct and exceeded 9 otherwise the problem was self-terminated and a response "no" executed. The carry parameter by itself was highly significant, and the addition of the self-terminating portion increased the explained RT variance an average of 2.8 percent for T1, and 4.1 percent for T2. Accordingly, subjects were exiting complex problems as soon as an error was encountered, and were using this executive process more efficiently during the second time of measurement.

Individual RTs were fit to the three models best representing the group level data. For simple addition, the MIN, SUM, and true-sum-squared models, along with the truth parameter, were used. For complex problems, columnwise SUM, MIN, and SSQ (sum-digits-squared), with the carry/self-terminating and truth parameters, were used.

For simple addition (1- column) problems the group data suggested the MIN was the processing strategy used. Indeed, the majority (13, and 12,

for T1 and T2, respectively) of individual RT data were best fit by the MIN parameter. However, Ashcraft's (1982) true-sum-squared provided the best representation of RT data for 9 and 7 (of 22) subjects for T1, and T2, respectively. In all, 11 subjects maintained the same strategy for T1 and T2. Five of the remaining subjects switched to the more mature true-sum-squared strategy for T2. The remaining subjects switched from a memory network strategy to an incrementing (SUM or MIN: Groen & Parkman, 1972) strategy. For simple addition the processing strategies appeared stable for about 50 percent of the subjects. The remaining subjects appeared to use a different strategy for the second session.

For complex problems (2-4 column), the columnwise SUM with carry/self-terminating and truth parameters provided the best fit for the group data. For 2-column problems there were equal numbers of individuals using the columnwise SUM and MIN proceudres for both T1 and T2 (8 and 9 persons, respectively), with about half as many subjects using the SSQ algorithm. In all, 12 of 22 subjects used the same procedure across sessions, but there was no clear pattern for those subjects switching procedures.

Strategies for 3-column problems showed the greatest stability with 15 of 22 subjects showing no change across times of measurement. However, unlike 2-column problems, subjects seem to use the columnwise SUM (12 and 13 persons) or SSQ (8 and 8 persons) strategy for T1 and T2, with only a few subjects using the MIN strategy. For T2, equal numbers of subjects switched to the SUM or SSQ strategy. Finally, the most complex problems showed the least stability across sessions, with slightly less than one-half of the subjects utilizing the same strategy for T1 and T2. The columnwise MIN strategy provided the best fit for 11 subjects and the columnwise SUM for 9 subjects at T1. There was a clear shift to the columnwise SUM strategy for T2, with this strategy being favored over MIN for T2 for most subjects (14 and 4 persons, respectively).

Individual level results suggest that about 50 percent of the subjects used the same columnwise processing strategies across T1 and T2. Patterns of change suggest that about one-half of those changing use a more mature algorithm (e.g. counting to memory network) for T2. Overall, these results suggest that algorithms estimated for group data represented individual level data fairly well, but did not reflect algorithms used for almost half of the subjects, and did not reflect individual changes in processing stategies across sessions.

In conclusion, both group level and individual level data support the model proposed by Widaman et al. (1985). For complex addition subjects used a columnwise (e.g. Poltrock & Schwartz, 1984) processing strategy beginning with the units column. The proposed executive (Sternberg, 1982), self-terminating strategy was supported. Subjects exited incorrect problems as soon as a column error was encountered and were becoming more efficient in the use of this metacognitive strategy across sessions.

The finding that the MIN algorithm best represented simple addition performance does not support Ashcraft's (1982) memory network model. The use of an incrementing stategy (Groen & Parkman, 1972) may be due to the fact that simple problems were presented randomly with more complex problems. Thus, the use of an incrementing strategy for complex problems may have led to the use of an incrementing strategy for simple problems.

For complex problems, columnwise algorithms used suggested that subjects

were using an incrementing stategy to obtain column sums; however, because of the speed of the incrementation it is unlikely that it is language based (Landauer, 1962). Alternately, subjects may have been moving slowly through a memory network, because of the complexity of the multi-column problems (Polltrock & Schwartz, 1984). However, these results suggest that subjects were using a procedural, rather than a memory based algorithm to complete complex problems.

Overall, the Widaman et al. (1985) model seems to reflect subjects processing stategies accurately for complex addition. The results from the individual level analyses suggest that subjects were using similar processes across sessions, but computational algorithms seemed to vary.

References

Ashcraft, M.H. (1982). The development of mental arithmetic: A chronometric approach. Developmental Review, 2, 213-236.

Ashcraft, M.H. & Battaglia, J. (1978). Cognitive arithmetic: Evidence for retrieval and decision processes in mental addition. Journal of Experimental Psychology: Human Learning and Memory, 4, 527-538.

Ashcraft, M.H. & Fierman, B.A. (1982). Mental addition in third, fourth, and sixth graders. Journal of Experimental Child Psychology, 33, 216-234.

Ashcraft, M.H., Fierman, B.A., & Bartolotta, R. (1984). The production and verification tasks in mental addition: An emperical comparison. Developmental Review, 4, 157-170.

Ashcraft, M.H. & Stazyk, E. H. (1981). Mental addition: A test of three verfication models. Memory & Cognition, 9, 185-196.

Groen, G.J. & Parkman, J.M. (1972). A chronometric analysis of simple addition. Psychological Review, 79, 329-343.

Landauer, T.K. (1962). Rate of implicit speech. Peceptual & Motor Skills, 15, 646.

Parkman, J.M. & Groen, G.J. (1971). Temporal aspects of simple addition and comparison. Journal of Experimental Psychology, 89, 335-342.

Poltrock, S.E. & Schwartz, D.R. (1984). Comparative judgements of multi-digit numbers. Journal of Experimental Psychology: Learning, Memory, and Cognition, 10, 32-45.

Sternberg, R.J. (ed.) (1982). Handbook of human intelligence. Cambridge: Cambridge University Press.

Widaman, K.F, Cormier, P., & Geary, D.C. (1985). A general model for cognitive addition. Manuscript submitted for publication.



New and recent titles from ACADEMIC PRESS...

HANDBOOK OF DISCOURSE ANALYSIS
Edited by TEUN A. VAN DIJK

Volume 1: Disciplines of Discourse
1985, 320 pp., $59.50, ISBN: 0-12-712001-7

Volume 2: Dimensions of Discourse
1985, 280 pp., $53.50, ISBN: 0-12-712002-5

Volume 3: Discourse and Dialogue
1985, 250 pp., $49.50, ISBN: 0-12-712003-3

Volume 4: Discourse Analysis in Society
1985, 215 pp., $49.50, ISBN: 0-12-712004-1

Set price for 4 volumes: $172.50

STRATEGIES OF DISCOURSE COMPREHENSION
TEUN A. VAN DIJK and WALTER KINTSCH
1983, 418 pp., $38.50, ISBN: 0-12-712050-5

LOCALITY IN LINGUISTIC THEORY
PETER CULICOVER and WENDY K. WILKINS
1984, 296 pp., $44.50, ISBN: 0-12-199280-2

PARSING NATURAL LANGUAGE
Edited by MARGARET KING
1983, 308 pp., $23.00, ISBN: 0-12-408280-7

MATHEMATICAL PROBLEM SOLVING
ALAN H. SCHOENFELD
Scheduled for publication in October 1985,
420 pp., ISBN: 0-12-628870-4

Titles in the COMPUTERS AND PEOPLE Series

FUNDAMENTALS OF HUMAN-COMPUTER INTERACTION
Edited by ANDREW MONK
1984, 312 pp., $26.50, ISBN: 0-12-504580-8

DEVELOPMENTS IN EXPERT SYSTEMS
Edited by M.J. COOMBS
1984, 254 pp., $15.00, ISBN: 0-12-187580-6

INTRODUCTION TO LOGIC PROGRAMMING
CHRISTOPHER JOHN HOGGER
1985, 288 pp., $46.00, ISBN: 0-12-352090-8 (cloth)
1985, 288 pp., $19.50, ISBN: 0-12-352092-4 (paper)

A Volume in the COGNITIVE SCIENCE Series

AGREEMENT AND ANAPHORA
A Study of the Role of Pronouns in Syntax and Discourse
PETER BOSCH
1983, 272 pp., $39.00, ISBN: 0-12-118820-5

JOURNALS

JOURNAL OF MEMORY AND LANGUAGE
Edited by MARCEL JUST and PATRICIA CARPENTER
Volume 24, 1985, 6 issues
Annual Subscription Rates: In the U.S.A. and Canada: $105.00
Outside the U.S.A. and Canada: $123.50

INTERNATIONAL JOURNAL OF MAN-MACHINE STUDIES
Executive Editors: B.R. GAINES and D.R. HILL
Volumes 22-23, 1985, 12 issues
Annual Subscription Rate: $337.00
ISSN: 0020-7373

COGNITIVE PSYCHOLOGY
Edited by EARL HUNT
Volume 17, 1985, 4 issues
Annual Subscription Rates: In the U.S.A. and Canada: $99.00
Outside the U.S.A. and Canada: $116.50
ISSN: 0010-0285

JOURNAL OF MATHEMATICAL PSYCHOLOGY
Edited by JAMES T. TOWNSEND
Volume 29, 1985, 4 issues
Annual Subscription Rates: In the U.S.A. and Canada: $120.00
Outside the U.S.A. and Canada: $143.50
ISSN: 0022-2496

SAMPLE ISSUES ARE AVAILABLE ON REQUEST.
Journal subscriptions are for the calendar year and are payable in advance.

Send payment with order and save postage and handling.
Prices are in U.S. dollars and are subject to change without notice.

ACADEMIC PRESS, INC.
(Harcourt Brace Jovanovich, Publishers)
Orlando San Diego Petaluma New York London
Toronto Montreal Sydney Tokyo
ORLANDO, FLORIDA 32887

COGNITION

International Journal of Cognitive Psychology

COGNITION is an international journal which publishes theoretical and experimental papers of the highest quality covering all aspects of the study of the mind. Contributions include papers dealing with research in the field of psychology, linguistics, neurophysiology, ethology, philosophy and epistemology.

Editor:

Jacques Mehler
Laboratoire de Psychologie
54, bvd. Raspail
F-75006 Paris, France

Associate Editors:

Thomas G. Bever
Columbia University, New York, USA
John Morton
MRC, London, U.K.
Michael Posner
University of Oregon, Eugene, USA

Editorial Associate:

Susana Franck, Paris, France

Subscription Information:
The subscription price for Vols. 19–21 in 9 issues (1985) is US $146.50/Dfl. 396.00
(add US $24.50/Dfl. 66.00 for postage/handling).
Total price for 1985: US $171.00/Dfl. 462.00.
ISSN 0010-0277

Individual subscribers may order a personal subscription (exclusively for personal use) at the reduced rate of US $83.25/Dfl. 225.00 (incl. postage and handling). **Personal subscriptions must be prepaid.**

The Dutch guilder price is definitive.
US $ prices are subject to exchange rate fluctuations.

Special Volume 18 on

Visual Cognition

Guest Editor: **Steven Pinker**

The past decade has seen an enormous leap in our understanding of visual cognition, thanks to the development of new experimental techniques and new models inspired by the study of computer vision within artificial intelligence. This special volume of **Cognition** (*Vol. 18, 1984*) contains a sample of work at the cutting edge of research in visual cognition. It includes original theoretical and empirical reports on shape recognition, visual imagery, visual thinking, and mental maps, using the methodologies of computational analysis and computer modelling, experimental psychology, human neuropsychology, and individual differences research. Included with the reports is a tutorial overview written by the Guest Editor intended to introduce graduate students and nonspecialists to the issues, models, and terminology used in the reports themselves. The volume should be invaluable to any reader interested in keeping abreast of the rapid progress and new insights arising from the study of visual cognition.

CONTENTS:
S. PINKER, Visual Cognition: An Introduction. D.D. HOFFMAN and W.A. RICHARDS, Parts of Recognition. S. ULLMAN, Visual Routines. R.N. SHEPARD and S. HURWITZ, Upward Direction, Mental Rotation and Discrimination of Left and Right Turns in Maps. S.M. KOSSLYN, J. BRUNN, K.R. CAVE and R.W. WALLACH, Individual Differences in Mental Imagery Ability: A Computational Analysis. M.J. FARAH, The Neurological Basis of Mental Imagery.

Single copies of this special volume are available at Dfl. 140.00/US $54.00 directly from the Publisher.
All orders should be accompanied by payment.

Please send this coupon (or a photocopy) to:
ELSEVIER SCIENCE PUBLISHERS
P.O. Box 211, 1000 AE Amsterdam
The Netherlands

In the USA and Canada:
ELSEVIER SCIENCE PUBLISHING CO., INC.
Journal Information Center
52 Vanderbilt Avenue
New York, NY 10017, USA

ELSEVIER

COGNITION, Vols. 19–21 (1985)

☐ Please enter a subscription for 1985 at US $171.00/Dfl. 462.00
☐ Please send *Visual Cognition* (Vol. 18) at US $54.00/Dfl. 140.00
☐ Please send a free specimen copy of COGNITION

Name _____

Address _____

_____

Signature

NH-I&B/PSYCH-57/JRL/6068d

At the forefront of research in Cognitive Science...

Just Published

Cognitive Psychology:
An Essay in Cognitive Science

George Mandler

A comprehensive exploration of contemporary cognitive psychology by a major figure in the field. Mandler stresses the evolution of consciousness as the focal issue of cognitive theory and applies this approach to specific problems in memory and emotion.

0-89859-537-1 (cloth) *1985/128pp./$19.95*
659-9 (paper) $6.95

Method and Tactics in Cognitive Science

W. Kintsch, J. R. Miller, P. G. Polson, eds.

A complete introduction to the field, focusing on original research. Linguists, cognitive psychologists, and AI researchers discuss their varied approaches and latest results.

327-1 *1984/368pp./$29.95*

Mental Models

D. Gentner, A. L. Stevens, eds.

A pioneering effort to compile and describe interdisciplinary research on the formal nature of human knowledge about the world.

242-9 *1983/360pp./$29.95*
A volume in The Cognitive Science Series

Reflections on Reasoning

R. S. Nickerson

A clearly written introduction to reasoning that guides high school and college students, teachers, and parents through straightforward discussions of the nature of logic, belief, and argument.

762-1 *Summer 1985/200pp.(tent.)/$19.95(tent.)*

Teaching Thinking

R. S. Nickerson, D. N. Perkins, E. E. Smith

A survey of recent and current work on the development of approaches to the teaching of thinking skills in the classroom.

539-8 *Summer 1985/456pp.(tent.)/$24.95(tent.)*

Mechanisms of Attention:
Attention and Performance XI

M. I. Posner, O. S. M. Marin, eds.

Reports new findings on how to measure the electrical and peformance consequences of information that does not reach conscious awareness. Offers new insights on the integration of concepts of selectivity into traditional models of visual and auditory psychophysics.

639-4 *Summer 1985/650pp.(tent.)/$59.95(tent.)*

Special offer to members of the Cognitive Science Society
$36.00 on prepaid orders

The Cognitive Science Series

D. A. Norman, A. Ortony, Series Editors

A two-part series designed to foster study and research in the various fields that comprise "cognitive science."

The Tutorial Series in Cognitive Science introduces the problems, concepts, and methods of Cognitive Science to advanced undergraduate and graduate students in contributing disciplines.

Technical Monographs and Edited Collections present original contributions that advance research in the field.

LAWRENCE ERLBAUM ASSOCIATES, INC., PUBLISHERS
365 Broadway, Hillsdale, New Jersey 07642

New from Bradford Books

Artificial Intelligence

The Very Idea

John Haugeland

The idea that human thinking and machine computing are "radically the same" provides the central theme for this illuminating and provocative book.

$14.95

A Cognitive Theory of Metaphor

Earl R. MacCormac

Metaphor plays a central role in human thought, particularly in modern science, yet Earl MacCormac is one of the few people writing specifically about metaphor and its role in thinking. In this book, he presents the first original and unified cognitive theory of metaphor using philosophical arguments which draw upon evidence from psychological experiments and theories.

$22.50

Visual Cognition

edited by Stephen Pinker

These essays tackle some of the central issues in visual cognition: How do we recognize objects in front of us? How do we reason about objects when they are absent and only in memory? How do we conceptualize the three dimensions of space? Do different people do these things in different ways? And where are these abilities located in the brain? This research, which appeared as a special issue of *Cognition*, is at the cutting edge of cognitive science.

$17.50 Softcover

Beyond Analytic Philosophy

Doing Justice to What We Know

Hao Wang

This cogent and knowledgeable critique of the tradition of modern analytic philosophy focuses on the work of its central figures–Russell, Carnap, and Quine–and finds it wanting. In its place, Hao Wang unfolds his own original view of what philosophy could and should be.

$17.50

Knowing Who

Stephen E. Boër and William Lycan

This is the first detailed study to explore the little-understood notions of "knowing who someone is," "knowing a person's identity," and related locutions. It locates these notions within the context of a general theory of believing and a semantical theory of belief- and knowledge-ascriptions.

$22.50

Conceptual Change in Childhood

Susan Carey

This case study of children's acquisition of biological knowledge between ages 4 to 10 reveals fundamental connections between issues in developmental psychology, epistemology, and history and philosophy of science.

$15.00

Now available in paperback

The Nature of Psychological Explanation

Robert Cummins

A major contribution to current discussions in the philosophy of mind, and philosophical problems of cognitive science and artificial intelligence.

$8.95

28 Carleton Street
Cambridge, MA 02142

THE MIT PRESS

