Artificial Intelligence Project

Final Technical Report

ARO Contract: DAAG29-84-K-0060

Artificial Intelligence Laboratory
The University of Texas at Austin

Austin, Texas 78712

2

Artificial Intelligence Project
Final Technical Report
ARO Contract: DAAG29-84-K-0060

Period of Contract: June 25, 1984 - August 31, 1990

Artificial Intelligence Laboratory
Department of Computer Sciences
University of Texas at Austin
Austin, Texas 78712

(512) 471-7316

Authors: Gordon S. Novak Jr.
Robert F. Simmons
Bruce W. Porter
Vipin Kumar
Robert L. Causey

MASTER COPY - - - - KEEP THIS COPY FOR REPRODUCTION PURPOSES

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY (Leave blank)	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	1990	Final Report 6/25/84-8/31/90

4. TITLE AND SUBTITLE

Artificial Intelligence Project

5. FUNDING NUMBERS

DAAG29-84-K-0060

6. AUTHOR(S)

Gordon S. Novak Jr

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

Univ of Texas at Austin
Austin, TX 78712-1188

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

U. S. Army Research Office
P. O. Box 12211
Research Triangle Park, NC 27709-2211

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

ARO 21683.61-MA-AI

11. SUPPLEMENTARY NOTES

The view, opinions and/or findings contained in this report are those of the author(s) and should not be construed as an official Department of the Army position, policy, or decision, unless so designated by other documentation.

12a. DISTRIBUTION/AVAILABILITY STATEMENT

Approved for public release; distribution unlimited.

12b. DISTRIBUTION CODE

13. ABSTRACT (Maximum 200 words)

The research performed under this contract was supervised by several different faculty members. This report summarizes the research that was performed, making reference to published papers and technical reports for the details. One goal was to support large numbers of students, through Fellowships as well as Research Assistantships, in order to increase the pool of people who are well-trained in A.I. A total of 68 students received financial support from this contract. There have been 21 Master's degrees and 13 Ph.D.'s (with 5 more Ph.D.'s expected by December) awarded to students who were supported by this contract or worked on the research projects. Another aspect of this contract was to provide eduation in A.I. to Army personnel. This was done through short courses held on campus and at Army locations.

14. SUBJECT TERMS

Artificial Intelligence, Automatic Programming, Physics Problem Solving, Knowledge Acquisition, Device Descriptions, Parallel Architectures

15. NUMBER OF PAGES

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
UNCLASSIFIED	UNCLASSIFIED	UNCLASSIFIED	UL

(Partial)

Contents

Over

(KR)

1 Executive Summary

1.1 Introduction

The Artificial Intelligence Project was a large contract to create a Center of Excellence in Artificial Intelligence at The University of Texas at Austin. We believe that this goal has been achieved. As a measure of our success, the following table shows the number of papers and invited talks, by institution, at the Eighth National Conference on Artificial Intelligence (AAAI-90) held in Boston, MA, from July 29 - August 3, 1990. This conference was heavily refereed, with an acceptance rate of less than 18% of the papers submitted.

Papers and invited talks at AAAI-90:	
Stanford	10
Illinois	9
Carnegie-Mellon University	8
University of Texas at Austin	7
Toronto	6
M.I.T., Xerox PARC, IBM Watson	5
USC, Bell Labs, NASA Ames, MCC	4
Rutgers, Michigan, Tokyo, U. Mass, UCLA, Brown	3
Texas A&M, U. Penn, Schlumberger, SRI, Maryland, etc.	2

As this table illustrates, The University of Texas at Austin had the fourth-highest number of papers and was ranked ahead of well-regarded universities such as M.I.T. The stature of The University of Texas in A.I. owes much to the support of the U.S. Army Research Office, and we are deeply grateful for this support.

1.2 Research

The research performed under this contract was supervised by several different faculty members. We have organized the research reports by faculty supervisor. These reports summarize the research that was performed, making reference to published papers and technical reports for the details. The papers and technical reports have previously been furnished to ARO; we will be happy to furnish additional copies on request. The report includes full listings of faculty and student publications, as well as a Technical Report List that gives abstracts of all AI Technical Reports over the contract period and includes an order form.

1.3 Student Support

Since this Center of Excellence contract had additional aspects beyond research support, we have included data about these.

One goal of ARO was to support large numbers of students, through Fellowships as well as Research Assistantships, in order to increase the pool of people who are well-trained in A.I. A total of 68 students received financial support from this contract. There have been 21 Master's degrees and 13 Ph.D.'s (with 5 more Ph.D.'s expected by December) awarded to students who were supported by this contract or worked on the research projects; others of these students are still working on their degrees. We have included a table listing all the students who have received support from this contract and the status of each. We have found that the ARO fellowship support, together with other graduate fellowships provided by the University, has allowed us to attract top-quality students to our graduate program.

1.4 Education for Army Personnel

Another aspect of this contract was to provide education in A.I. to Army personnel. This was done through short courses held on campus and at Army locations; we presented 29 short courses, covering approximately 2,379 pages of lecture material, that were attended by a total of approximately 935 personnel. Several visiting scholars from the Army spent time in residence in our laboratory. Two Army officers were enrolled in the Ph.D. program during this period. We also furnished to ARO a videotaped course on Artificial Intelligence and Expert Systems, comprising 60 hours of videotaped instruction with programming exercises and course notebooks, for duplication and use within the Department of Defense at no charge. (This course sells commercially for $9,000.00 per copy.) We have included tables describing our educational activities for the Army.

1.5 Support of Army Activities

Our faculty supported Army activities by attendance at Army conferences, by making presentations to Army visitors, and by participation in the evaluation of Lockheed's AirLand Battle Management System. We have included a table listing interactions with the Army.

1.6 Leverage

The ARO contract was instrumental in helping us to win additional grants from equipment and software manufacturers. The total of retail prices of these equipment and software grants exceeded the amount of the ARO contract, so it can be argued that we achieved greater than 100% leverage. The ARO contract was very helpful because we could tell the equipment manufacturers that we had the personnel support money to support the research that would be done using their equipment. From the manufacturer's point of view, this was leverage for their grant, so the existence of the ARO contract made it more attractive for them to give equipment to us. The following table summarizes the donations we have received from manufacturers.

Donor:	Equipment:	Approximate Value:
Hewlett Packard	12 HP 9836 workstations	$822,000
Xerox	30 Dandelion Lisp Machines	$1,400,000
Hewlett Packard	60 HP 9000/300 series workstations	$5,000,000
Texas Instruments	20 Explorer Lisp Machines	$1,000,000
Intellicorp	6 copies of KEE software	

We also benefited from leverage in the form of external student support. Several students were supported by company-paid educational fellowships or by working part-time for local companies. These students worked on the ARO research projects, but we did not have to pay them; such students are listed as "Students with Other Support" in the student tables.

2 Research Supervised by Gordon S. Novak Jr.

2.1 Automatic Programming by Reuse of Generic Algorithms

Much has been written about the so-called *software crisis*: programming takes too much time, costs too much, and produces programs that contain too many errors and are difficult to change. There is a great need to improve the productivity of programmers and the quality of the resulting programs. Given the large amount of money spent on software by DoD, this is especially important: improvements in programming are highly leveraged.

We believe that the approach most likely to succeed in making software production significantly easier is *software reuse*. In this approach, algorithms written previously are reused for an application that was not originally foreseen.

It can be argued that much of what human programmers do is adaptation of standard algorithms for particular applications. Much of computer science education involves teaching of standard algorithms for such tasks as sorting, tree searching, and so forth. Upon graduation, the student is expected to be able to select appropriate algorithms for an application and to recode them to work with the application's data structures in the programming language being used. In effect, the human programmer is acting as a compiler to specialize the generic algorithm for the application.

Software reuse is a *knowledge-based* approach; this can be contrasted with transformational approaches that seek to transform program specifications through successive stages until executable code is reached. In effect, such approaches require that algorithms be *reinvented* for each use. Invention of good algorithms is not easy, even for human experts. Our approach is to reuse the knowledge of human experts, stored in the form of generic algorithms.

Our work in the area of automatic programming is based on extension of the GLISP programming language and compiler [Novak 83]. GLISP is a high-level language with abstract data types that is compiled into Lisp. (The Lisp can be compiled to machine language by the Lisp compiler.) The GLISP compiler is able to compile a generic program, written in terms of abstract data types, into a form specialized for a particular application and the actual data types used for that application. Instead of having to rewrite an algorithm for an application, the programmer only needs to specify how the data of the application match the abstract data with which the algorithm is associated; this can be done easily, and in some cases automatically. The result is that a small amount of specification by the user of the system is expanded into a much larger program that would otherwise have to be written by hand.

The following sections describe the individual research projects in this area. Each of the projects addresses the goal of making use of computers for applications significantly faster and easier by reuse of knowledge stored in the form of generic algorithms.

2.1.1 Generation of Looping Programs by Menu Selection

This facility, described in [Novak 83], was completed prior to the start of the current contract; we describe it briefly for completeness and because some of our other projects use related techniques. The GEV data inspector associated with GLISP allows the user to inspect data in a window according to its GLISP datatype description. GEV also allows simple looping programs to be created by menu selection. The user first specifies the operation to be performed in a loop, such as `TOTAL`, `AVERAGE`, or `MAXIMUM`. Next, the system presents a menu of all sets of data available in the current context that could be examined in a loop: for example, the user could select `CONTRACTS`, a set of contracts for a department. Next, the system presents the available data fields within the data type of the loop item, such as a `CONTRACT`, so that the user can select the data to be operated on within the loop, such as the data to be totaled. This process continues through subtypes of the loop data until a terminal datatype, such as a number, is reached. At that point, the menu specifications are used within a program skeleton to produce a GLISP program from the user's specifications; this program is then compiled and run to produce the desired answer. For example, the user might specify, through successive menus, `PROGRAM TOTAL CONTRACTS BUDGET LABOR`, producing a program to loop through the contracts and total the `LABOR` field from the budget of each contract.

The menu-based programming facility of GEV is simple enough to be used by managers or secretaries who are not programmers, but who may need to write simple programs. It produces in seconds programs of a few dozen lines that might take a skilled programmer twenty minutes to write. Thus, it is a small-scale demonstration of our goals of ease of use and productivity improvement by expansion of a small specification into a larger program.

2.1.2 Reuse of Library Programs by Reformatting Data

Reuse of programs stored in a library is potentially an easy way to get the benefit of a program at little or no cost. Unfortunately, in practice such reuse typically requires that the user understand the documentation of the library program and prepare the input data in the form that the library program requires; this may require writing a program to reformat the data. Man Lee Wan's LINK program [Wan 85] automates this process. A description of the data required, in the GLISP data description language, is associated with the library program; a description of the user's data is also assumed to exist. From these two descriptions, LINK performs a menu-based negotiation to determine how the data needed by the library program can be obtained from the user's data. For example, a program that draws a pie chart might expect as input a list of pairs, where each pair consists of a label for a slice of the pie and a number representing the size of the slice. The user might specify that, from a set of data representing divisions of a company with various data about each division, he would like a pie chart of the food divisions, with the label of each pie slice being the name of the division and the size being net profit of the division. From these specifications, LINK produces an interface program that produces a data set in the proper format from the given data set, then calls the application program with this data set.

Interfacing programs by modifying the data is an efficient method when the data sets involved are relatively small.

2.1.3 Negotiated Interfaces: Reuse by Program Specialization

The GLISP compiler is able to specialize an abstract algorithm, written in terms of abstract data types. for particular data types that are instances of the abstract types. This is done by means of a *connection type* whose stored implementation is the actual data type, whose superclass is the abstract type, and whose computed properties express the features needed by the abstract type in terms of features of the actual type.

Fredrick Hill's NI program [Hill 85] allows a connection type to be constructed automatically from specifications of how given data match the abstract data required by the algorithm: these specifications are acquired through an easily used menu-based negotiation process Following the negotiation process, the generic algorithm is compiled using the connection type, which results in a version of the algorithm that is specialized for the user's data. The compiled version of the algorithm is expressed in Lisp and is symbolically optimized.

Hill used as an example a generic program for drawing diagrams of tree structures. Programs that can draw diagrams of tree structures directly from the original data are seldom found in program libraries because of the wide variation in data structures that can be used to represent trees. Hill's NI system and the GLISP compiler can specialize a single generic tree-drawing algorithm to work with a variety of different tree structures. In the negotiation process, the user specifies the node of a tree, how to tell whether a node is terminal, how to get the children of a node, and what printed representation is desired for a node. The result is a *connection type* that describes how a user's data structure can be viewed as a tree; compilation of the generic algorithm through the connection type results in a specialized version of the algorithm to draw the user's tree in the desired fashion. A given tree can be drawn in various ways; for example, a family tree could be drawn to show only the females. Data that might not normally be thought of as a tree can be viewed as one: for example, an integer can be viewed as a tree by dividing its binary representation into left and right halves until a single bit or a value of zero is reached. These examples illustrate how the same generic algorithm can be automatically recompiled to operate on radically different data.

Hill's work is an inspiring demonstration of the potential power of reuse of generic algorithms. Using his program, several pages of specialized Lisp code can be produced in a few seconds. We are continuing this line of research with the work of Novak and Sayrs (described in a later section).

2.1.4 Acquisition of Data from Text

Although much progress has been made in generating sophisticated output from computers, there has been less progress on the input side. Most computer input must still be generated by keyboard input in rigid formats. Text scanners are becoming commonplace and inexpensive; however, text that has been scanned and is stored in ASCII form still is not usable for computation, especially if its format is even slightly variable.

Margaret Reed-Lade's program [Reed-Lade 89] performs acquisition of data from semi structured text, in the form received from a Kurzweil scanner. By analysis of the text, her program is able to recognize regularities in format (e.g., rows and columns, or repeated groups of running text) and to infer data types from the form of the data (e.g., the string `(512) 471-7316` is recognizable as a telephone number). The program displays the text on a graphical display, with boxes surrounding groups of text and features of the text displayed for each box. The user can add titles for sections of text and modify their features. When this is finished, the result is a grammar that is used by a "fail-soft" parser to read the text into a structured data set. A GLISP data description is also generated for the data set. This allows the data to be used with the other GLISP tools; for example, the user could go rapidly from printed page input to internal data set to a pie chart derived from the data. Reed-Lade's program is able to handle both tabular data and running text.

Ms. Reed-Lade was employed at Lockheed in Austin while completing her thesis, so she was not paid from contract funds.

2.1.5 Analyst's Workbench

Analyses performed for defense and national security decision makers often involve physical calculations as well as data retrieval. These calculations may involve physical laws, unit conversions, and views of data as being of different types (for example, a quantity of oil might be viewed in terms of its volume, weight, energy content, or cost). There is a need for analysts to be able to rapidly perform such calculations without having to write specialized programs in conventional programming languages, which is slow and error-prone.

Yusuf Mauladad wrote a program [Mauladad 87] that allows the user to graphically connect physical laws and quantities to perform an analysis. Known quantities are propagated through the network to derive values for unknowns.

Ms. Ruey-Juin Chang is developing an Analyst's Workbench that will allow an analysis to be constructed rapidly by a human analyst using graphical displays. The analysis may involve existing databases, geographical information stored in maps, physical laws, and conversion of measurement units. A simple example question is *What fraction of the Texas population lives within 100 miles of Austin?* This question might be answered (approximately - for many questions, only approximate answers are possible) by displaying a map of Texas, requesting a circle of radius 100 miles centered at Austin, asking for the set of counties that are contained within the circle, then asking for the sum of populations over that set. Constraint propagation is used to reduce the amount of user input that is required to specify an analysis. By bringing together a collection of analysis methods, unit conversions, physical laws, databases, and the ability to automatically interface them, the Analyst's Workbench will allow complex analyses to be constructed rapidly and easily.

2.1.6 Rule Language for GLISP

Christopher Rath wrote a rule language for use with the GLISP system [Rath 86]. Such a rule language could be useful for cases where a rule-based expert expert system is needed, for example to advise a programmer in data structure selection.

Rath was supported by Bell Labs during his graduate work, so he was not paid from this contract.

2.1.7 Recent Work

Our ultimate goal is to be able to collect a set of algorithms such as those found in Knuth's *Fundamental Algorithms* and to be able to specialize an algorithm for an application in a few seconds. Hill's system is a small-scale demonstration of this goal: it allows several pages of code to be generated effortlessly in a few seconds; however, it works only for a limited range of algorithms. We wish to extend this range to create a truly useful programming tool. A programmer should be able to specify the desired implementation of an application at a high level, by specifying data structures and algorithms to be used. The selected algorithms should then be automatically specialized into runnable code for the selected data structures.

We believe that selection of data structures and algorithms to be used for a sizeable application is difficult to do automatically because it requires a global view of all the uses of the data. The data structures and algorithms that are selected must be efficient (in computation, storage, response time, etc.) for all aspects of the application, or must trade off the various efficiency measures in an appropriate way. We assume that a skilled human programmer will be needed to do this, at least in the near future. However, we wish to give the human programmer powerful tools for implementing those decisions; this will also make it more reasonable to experiment with different implementations.

Gordon Novak and Brian Sayrs (working at Lockheed in Palo Alto, CA) are extending the previous techniques to allow use of more complex data structures and combinations of algorithms. A simple example that exposes many of the salient issues is *priority queues*. At an abstract level, we can say that a priority queue is composed of an *indexing method* that maps from an integer (the priority) to something (in this case, a queue) and a *queuing method* that maintains a queue of things. There are many possible indexing methods (array, linked list, binary tree, etc.) and many possible queuing methods (simple linked list, linked list with end pointer, circular queue within an array, etc.). Any combination of these methods could be used to construct a priority queue, and any might be the method of choice for some application. We want to allow the programmer to easily specify a priority queue with exactly the desired representation and to automatically generate efficient code for the priority queue from the pieces of code associated with the selected components.

Views of Data Existing programming languages have a very restricted notion of data type: a given data record has only one type, it is represented as a unitary data structure, and it has no relation to other data types (except as subrecords). If the data structures actually used by programmers are to be described, these restrictions must be removed.

In actual applications, data often have multiple "types". For example, if records are kept in a queue, from the standpoint of the queue manipulation program they are queue records, and their data contents do not matter; from the standpoint of the program that processes records removed from the queue, they are data records, and any queue data (such as pointer fields) do not matter. We have implemented *views* that allow a given data type to be viewed in multiple different ways. By specifying a view, the generic algorithms associated with a view can be automatically specialized. Note that a single data structure could have multiple views of the same kind: for example, a `STUDENT` record could be viewed as a sorted linked list record with sorting on different fields (name, grade average, etc.). Views interact with clusters (described below).

Clusters of Data Types Data types often are related to other data types; these relationships need to be represented in order to allow more complex algorithms to be specialized by the compiler. For example, consider a pointer to a record. In standard programming languages, a pointer type can only be derived from the record type; usually, a pointer is the machine address of the record in main memory. In general, though, there can be other types of pointers: a pointer could be the disk address of a print buffer, or a pair `(row, column)` that denotes a square on a checkerboard. These examples illustrate that a pointer may have its own data structure and types, and that the memory substrate into which it points must also be part of the specification. We have developed a *cluster* description that allows clusters of related types to be described and to mutually reference each other. Inheritance must be handled properly for clusters, so that instead of a single type inheriting properties from a single supertype (as in object-oriented programming), each type of a cluster inherits from its corresponding type in a super-cluster and relationships between types at a given level are maintained.

Automatically Producing Cluster Views A first version of a program to produce the data types required to view a given data type as an instance of a cluster has been implemented. For example, given an arbitrary user data structure that might be viewed as a linked list, the system will automatically produce such a view, asking for choices if multiple such views are possible. Given the views that are produced, all of the standard linked list algorithms will be automatically specialized when asked for.

Library of Generic Programs We have begun construction of a library of generic programs that can be specialized as needed for applications. The library not only is useful as programming knowledge that can be reused, but also serves to stress the compiler and representation techniques to reveal weaknesses. We have also tried to stress our techniques by producing "exotic" test data structures that are similar to those used in actual applications but that cannot be handled "within the system" by existing programming languages.

Our library currently includes a variety of operations on linked lists (`first`, `rest`, `length`, `nreverse`, `copy-list`, etc.), several kinds of queues built on linked lists (front pointer queue, two-pointer queue, circular queue with end pointer), priority queues built on queues, sorted linked lists, and some operations on trees.

2.2 Physics Problem Solving

The goal of our research on physics has been the development of methods by which a computer program can understand an *informal* statement of a physics problem, such as a textbook "story problem" expressed using English text and diagrams, and can construct from it a *formal model*, expressed in terms of "physics objects" (such as a point mass or rigid body) and relationships, from which a solution to the problem can be found according to physical laws. We have focused primarily on statics and Newtonian mechanics. Although these areas are considered simple by physicists, they contain many challenging problems and are important in engineering practice. In addition, we believe that this domain is sufficiently complex that it will require development of techniques that will extend to many other domains.

Solving problems such as those we are addressing is important in several respects. Physical analysis is required for engineered products of all kinds, and especially for military hardware that is often at the leading edge of technology. Human analysis of such problems is slow, costly, and often error-prone: machine-aided analysis could potentially be faster, more accurate, and more likely to be done (because it would be easy to do and inexpensive). The difficulty of engineering analysis often causes analyses to be skipped altogether, approximated, guessed at, or done incorrectly; the result may be bad design, resulting in system failure or very expensive rework. Improved engineering analysis could have large payoffs in improved quality of military hardware, ability to meet specifications, reduced failures, and reduced costs for correcting design flaws. In addition, a thorough understanding of the process of solving physics problems could help teaching of physics and thus improve the country's scientific infrastructure. Such improvement is badly needed: a recent study found that only 17% of American high school seniors can solve a story problem involving more than one step.

We believe that the most fundamental process in solving a physics problem is *model building*, that is, deciding how to model the real-world objects and relationships of the problem by "physics objects" (such as point masses) whose behavior is governed by physical laws. Although the writing and manipulation of equations are the most easily observable steps in human solving of physics problems, model building precedes equation generation, and correct equation generation depends upon it. Chi *et al.* have found that experts and novices differ when asked to sort physics problems into piles of "similar" problems: novices sort problems based on surface features (e.g., "problems involving pulleys"), while experts sort on the basis of physical deep structure (e.g., "force problems"). This difference reflects differences in the model building process, at which the experts are much more skilled.

Model formulation is a nontrivial process. For a given problem, there will be many physical interactions that in principle are operating, but in practice should be ignored (e.g., electrostatic forces between spacecraft), often because their effects are relatively small in the context of the problem at hand. The skilled problem solver must ignore the insignificant factors, but must not omit any important factors.[1] Frequently, parts of the required model are unspecified in the problem statement and must be inferred from context. The set of equations resulting from the model must be mathematically tractable. The size and complexity of the equation set is dramatically reduced by selection of appropriate coordinate systems that exploit symmetry to reduce the dimensionality of the model and eliminate additive constants.

Our early research on physics problem solving resulted in the ISAAC program, which is able to understand rigid body statics problems stated in English, solve the problems, and draw diagrams of them. This research brought to light many interesting problems that we are currently pursuing.

2.2.1 Understanding English Text and Diagrams

Humans often use diagrams to communicate spatial information: this is especially true in the military domain, where standard diagrammatic forms of representing battlefield situations have been developed. There has been relatively little work to date on input of freely generated diagrams to computers.

William Bulko developed the BEATRICE system [Bulko 89, Novak and Bulko 90], which allows specification of a physics problem by a combination of English text and a diagram. English is a poor language for specifying spatial relationships of much complexity. Human technical communication relies heavily on the use of diagrams, and most textbook physics problems are specified by a combination of English text and a diagram. Bulko's system allows diagrams to be input through an interactive graphics program: the user selects picture items from menus, adds them to the diagram, and scales and moves them into position. As a side effect, a symbolic description of the diagram is created for input to the understanding program. The symbolic description is in a form that could reasonably be produced from a written diagram by a visual scanning system. The English text and diagram representation together are input to a blackboard system that opportunistically parses the English, combines diagram elements (e.g., if `theta` appears between two lines forming an acute angle, the two lines and `theta` could be grouped as a representation of an angle), and establishes *coreference* between text elements and picture elements. The result is a unified description of the problem that can serve as input for a problem solver.

2.2.2 Model Representation for Physics Problems

Hyung-Joon Kook has written a program, called APEX [Kook and Novak 90], that allows the physical models chosen for a problem to be represented within the computer. Most systems to date have combined the real-world problem description and physical model in a

[1]Sometimes it is suggested that *all* applicable equations be written, and that they then be reduced to the correct set by mathematical approximation. This is infeasible for several reasons. First, the set of potentially applicable interactions is unbounded. Second, in order to determine mathematically that some interactions could be eliminated, unavailable data would be required. Third, the algebra involved quickly becomes intractable, even for computer algebra systems.

single representation; this limits the scope of problems that can be solved. By separating the model from the real-world representation, several benefits are gained: a single object can have more than one representation, as needed for different parts of the analysis; multiple objects can be combined into a single physical system for analysis; results obtained from one part of the analysis can be propagated "backward" through the views to real-world objects and to other parts of the analysis. Kook's system takes as input a semantic network similar to that produced by Bulko's system. A human expert can select the models to be used: a graphical display shows the relationships of objects and models. Finally, equations are produced from the models.

In a fully developed physics problem solver, the determination of the models to be used for a given problem would be performed automatically. Kook has provided the interfaces to allow an expert system to replace the human expert in specifying models, and has written a small expert system to demonstrate this capability.

2.2.3 Work in Progress on Physics Problems

Two Ph.D. candidates, Xiang-Seng Lee and Hiow-Tong See, are continuing dissertation research on how an expert problem solver program can choose the proper models for analysis of a given problem. Lee, who has a Ph.D. in physics, is working on more difficult problems that are underspecified, have symbolic values rather than numeric values, have nontrivial geometry and temporal relations, and may require creation of "gedanken" features. An example of such a problem is:

> A boy is seated on the top of a hemispherical mound of ice with a radius R. He is given a very small push and starts sliding down the ice. What is the height of the point at which he leaves the ice if the ice is frictionless?

Solving this problem requires representing the geometry of a cross-sectional view of the hemisphere, creation of a point on this representation to represent the point at which the boy leaves the ice, determination of the forces involved, vector analysis, and symbolic manipulation of the resulting equations. Lee's program is currently able to solve this example problem.

Hiow-Tong See is working on solving a larger number of simpler problems involving a wide range of physical principles. His work should result in formalisms for modeling these problems and a set of rules that can determine what principles to use for analysis of an informally stated problem.

2.3 References

Bulko, William C., "Understanding Text with an Accompanying Diagram", *Proc. First International Conference on Industrial and Engineering Applications of AI and Expert Systems*, Tullahoma, TN, 1988, pp. 894-898; Technical Report AI-88-69, March 1988.

Bulko, William C., "Understanding Coreference in a System for Solving Physics Word Problems", Ph.D. Dissertation, Technical Report AI-89-102, May 1989.

Hill, Fredrick, "Negotiated Interfaces for Software Reusability", Master's Thesis, Technical Report AI-85-16, Dec. 1985.

Kook, Hyung Joon, "A Model-based Representational Framework for Expert Physics Problem Solving", Ph.D. Dissertation, Technical Report AI-89-103, May 1989.

Kook, H. J. and Novak, G., "Representation of Models for Solving Real-world Physics Problems", *Proc. 6th IEEE Conf. on A.I. Applications*, March 1990, pp. 274-280. Also Technical Report AI-88-93, Dec. 1988.

Kook, H. J. and Novak, G., "Representation of Models for Expert Problem Solving in Physics", accepted for *IEEE Transactions on Knowledge and Data Engineering*, approx. Dec. 1990.

Mauladad, Yusuf, "CALCLAB: An Interactive Environment for Scientific Numerical Problem Solving", Master's Thesis, Technical Report AI-87-59, Aug. 1987.

Novak, G., "GLISP: A Lisp-based Language with Data Abstraction," *A. I. Magazine*, Vol. 4, No. 3, 1983. Reprinted in *Readings from A.I. Magazine*, AAAI Press, 1988.

Novak, G., "TMYCIN Expert System Tool", Technical Report AI-87-52, 1987.

Novak, G. and Bulko, W. "Understanding Natural Language with Diagrams", *Proc. National Conference on Artificial Intelligence* (AAAI-90), Boston, MA, pp. 465-470, Aug. 1990.

Novak, G., R. Causey, V. Kumar, J. Werth, B. Porter, V. Hwang, "Introduction to Artificial Intelligence and Expert Systems" (videotaped course with exercises), Morgan Kaufmann Publishers, 1988.

Rath, Christopher, "A Rule Language for the GLISP Programming System", Master's Thesis, Technical Report AI-86-31, 1986.

Reed-Lade, Margaret, "Grammar Acquisition and Parsing of Semi-Structured Text", Master's Thesis, Technical Report AI-89-105, May 1989.

Wan, Man Lee, "Menu-Based Creation of Procedures for Display of Data", Master's Thesis, Technical Report AI-85-18, Dec. 1985.

3 Research Supervised by Bruce W. Porter

3.1 Knowledge Acquisition for Expert Systems

Research Assistants: Ray Bareiss, Rich Mallory

We recently completed a project which applied psychological theories of concept learning and classification to build an effective knowledge-acquisition tool. The tool, Protos, learns to perform heuristic classification, the problem-solving method used in most expert systems.

A major research problem that this project addresses is learning and using ill-defined categories, which occur in many "real" domains. In medicine, for example, clinicians must diagnose patients' disorders without strict definitions of each disorder. Descriptions of disorders within a category vary widely, and acquiring the knowledge to accurately assign cases to categories is difficult. Protos learns such categories and then classifies new cases by explaining their similarities to known exemplars.

Protos was applied to the task in clinical audiology of identifying a patient's hearing disorder from symptoms, test results, and history. An expert clinician instructed Protos with 200 cases—a level of training comparable to that received by student clinicians. After this training, Protos's classification accuracy was compared with that of clinicians and several other learning programs. Protos compared favorably with the best clinician and was significantly better than the other programs.

The results of the Protos research were distributed in two ways. First, research reports were published in *Artificial Intelligence* and *Machine Learning.* Using early versions of these papers, the Turing Institute reconstructed Protos for comparison with rule-learning programs. Second, a Common-Lisp version of Protos was distributed to many research laboratories. Their projects will help determine Protos's applicability to a broad range of domains.

Ray Bareiss completed his Ph.D. and is now a member of the computer science faculty at Northwestern University.

3.2 AI & Legal Reasoning

Research Assistants:
K. Branting (Doctor of Jurisprudence and Ph.D. graduate student),
Neil Cohen (Doctor of Jurisprudence and Ph.D. graduate student)

The goal of the artificial intelligence and legal reasoning project is to design and build systems capable of performing legal analysis by applying the reasoning of precedent cases to new factual situations and generating legal arguments. Legal expert systems that focus exclusively on legal rules fail to capture the ability of attorneys to reason about new cases that do not closely match the wording of legal rules and to create arguments about whether a given legal rule should apply. These fundamental legal skills require both knowledge of the explanations underlying precedent decisions and sufficient world knowledge to reason about whether such explanations are applicable to new cases. Developing a computer representation of the explanation structure of legal opinions and the large-scale knowledge-base of general information necessary for applying these explanation structures to new cases are basic objectives of the legal reasoning project.

An innovation of the legal reasoning project is the use of case-based reasoning as the basis for a model of legal concepts and problem solving. This approach, which draws on recent research in cognitive psychology, is an extension of Protos, which has demonstrated an expert level of performance in clinical audiology. However, legal reasoning presents a greater challenge than audiology because the descriptions of legal cases and legal explanations are, in general, considerably more complex than those in audiology.

Implementation is completed on a prototype system to perform reasoning about Workers Compensation under Texas law. The system contains a collection of legal rules derived from the controlling statute together with a library of precedent cases and their explanations. It identifies and retrieves precedent cases relevant to new cases, creates arguments about whether each new case is controlled by the precedent, and produces a complete analysis under the statute based upon the relevant precedents. We are currently evaluating the system with the assistance of law students and practicing attorneys.

Karl Branting will complete his Ph.D. in December of 1990.

3.3 Human Interface Design

Research Assistant: Brad Blumenthal

The goal of this research is to develop new techniques for making computerized design replay systems more flexible in adapting to minor differences in a new situation and in reusing as much of their relevant experience as possible. One aspect of this goal involves enabling such systems to automatically re-orient themselves when the current design episode does not exactly match previous ones. Another aspect is separating the kinds of design knowledge being used, so that minor differences in details do not prevent relevant experience from being reused.

We pursued this goal by designing and constructing a computer system to partially automate the design of metaphoric human interfaces for computer applications. The system consists of two components. The performance component is a design system which uses an agenda-based control mechanism combined with a large number of human interface design heuristics to produce a detailed, metaphoric interface specification from a generic application description. The learning component uses a derivational analogy based design replay algorithm which takes the history produced by the first component and uses it to guide subsequent design episodes using different metaphors. This research project concentrates on improving the performance of the learning component.

Brad Blumenthal will complete his Ph.D. in December of 1990.

3.4 Biology Knowledge Base

Research Scientist: Art Souther
Research Assistants: Jeff Rickel, Rich Mallory, James Lester, Liane Acker, Ken Murray

Our long-term research objective is to build knowledge bases for science that will support computer-based tutoring and other applications. As an initial exploration, we have built the Botany Knowledge Base — a large-scale, structured knowledge base in the domain of plant anatomy, physiology, and development. We have chosen this domain because it is relatively self-contained yet is representative of nonformal domains. Like most domains, botany is concerned with objects (*i.e.*, plant anatomy) and processes that change them (*i.e.*, plant physiology and development). It also incorporates both common sense and expert knowledge. For example, the common sense notion that living things require nutrients supports the expert knowledge that plant embryos consume endosperm.

With the help of Professor Bassett Maguire of the Zoology Department, we are extending the Botany Knowledge Base by concentrating on particular areas within plant physiology and adding new information about the plant's immediate environment, *e.g.*, soil conditions such as water and nutrient content, and atmospheric components such as carbon dioxide content and temperature. Our goal is to encode the knowledge which lies at the intersection of physiology and ecology. Two rapidly expanding disciplines comprise this knowledge: physiological plant ecology, which considers ecology from a physiological perspective, and environmental physiology, which considers physiology from an ecological perspective. The resulting Biology Knowledge Base is being used in the following research projects.

3.5 Knowledge Eng' ,eering

Computer Programmer: Erik Eilerts
Research Assistants: Rich Mallory, Brad Blumenthal, Jeff Rickel

During the last three years, we have developed a sophisticated software system that facilitates building large knowledge bases such as the Biology Knowledge Base. This system is fully operational and it provides the foundation for our research on human interface and explanation generation.

The system consists of two interrelated components, *Cypress* and *KnEd*. Cypress is a knowledge representation language based on Theo, a language developed by Tom Mitchell at Carnegie-Mellon University. We have added methods for representing rules and constraints.

In addition to this basic functionality, Cypress provides features important for building multifunctional knowledge bases. Of utmost importance is the ability to represent viewpoints, which are collections of facts that should be considered together. For example, the viewpoint "car as a manufactured artifact" contains information about raw materials and the assembly process, while the viewpoint "car as a consumer durable" contains information about purchase costs and longevity. A multifunctional knowledge base contains many, highly-integrated viewpoints for each concept.

Past research on using viewpoints for organizing knowledge has assumed that all viewpoints are represented explicitly. Unfortunately, explicitly representing viewpoints for a large knowledge base is infeasible. Our research addresses this problem with methods for creating viewpoints when they are needed. This is done using a relatively small number of general viewpoints, which we call "view types," that are instantiated for specific concepts.

KnEd is a software tool, built on Cypress, for displaying and editing a knowledge base. With KnEd's mouse and menu operations, a user can "navigate" through a complex structure and selectively display it both graphically and textually. Numerous editing operations are available, such as adding a frame to a graph, changing a frame's attributes, and creating a rule to compute information when required.

A unique feature of KnEd is the capability to edit the knowledge base by editing graphs. ¿From years of experience building the Biology Knowledge Base, we have found that graphs of knowledge structures are very effective for organizing domain knowledge and communicating with others. After a graph is used to reach consensus on an encoding of domain knowledge, it can be converted into Cypress's language of frames. With other systems for building knowledge bases, this conversion is done manually by a "knowledge engineer." However, we have automated this conversion process in KnEd, allowing a domain expert to extend and modify the knowledge base by creating and editing graphs.

KnEd can also convert from Cypress's language of frames to graphs. This ability allows a domain expert to view and edit the knowledge base without working with the details of frames.

3.6 Knowledge Acquisition

Research Assistants: Ken Murray

The major obstacle to building multifunctional knowledge bases results from their size and complexity. Knowledge base modifications that are intended to correct one shortcoming may conflict with existing knowledge and introduce new problems. For example, extending a drug therapy advisor (*e.g.*, Mycin) to minimize the number of drugs prescribed to each patient conflicts with other therapy goals, such as maximizing the number of symptoms covered by the prescribed treatment. Identifying how new information conflicts with existing knowledge is difficult: conflicts are often implicit, and the complexity of identifying interaction between new information and existing knowledge increases with the size of the knowledge base. Developing the technology to determine how new information interacts with existing knowledge is the principle requirement for supporting the construction and maintenance of very large, multifunctional knowledge bases, and it is the focus of our knowledge-acquisition research.

Knowledge integration is the process of incorporating new information into an existing knowledge base; it involves determining how the new information interacts with the existing knowledge. For the past three years we have been constructing KI, a tool that performs knowledge integration as it helps a domain expert extend the Biology Knowledge Base.

When provided with new information, KI retrieves relevant knowledge from the knowledge base and uses it to critique the new information. This involves identifying the ways in which existing knowledge corroborates or conflicts with new information.

Ken Murray plans to complete his Ph.D. during Spring semester of 1991.

3.7 Generating Explanations from a Large Knowledge Base

Research Assistants: Liane Acker, James Lester

We are beginning research on answering questions by generating coherent explanations from a large knowledge base. Current methods for generating explanations use schemata, or templates, of common explanation patterns. Because schemata restrict the content and organization of explanations, schema-driven tutoring systems are inflexible in three ways. First, when selecting the content of an explanation, schemata are insensitive to the user's knowledge and the previous discourse. Second, schema-driven generators cannot take advantage of opportunities to interject new information that is closely related to the current topic. Third, schema-driven generators cannot allow users to interrupt to ask questions: by responding to a question, an explanation generator may need to dramatically alter the remainder of its explanation, and schema-driven systems cannot revise explanations.

To address these problems, we are designing an explanation generator that will achieve flexibility in three ways. First, it will produce *integrative explanations* that relate new information to the user's existing knowledge. In producing an integrative explanation, we can define three networks of relevant concepts and relations. The *target* network is the set of concepts and relations that a system seeks to communicate to the user. The *base* network is the set of concepts and relations that model what the user already understands and is relevant in some way to the target. The *linking* network is the set of concepts and relations that relate the target to the base. To produce an integrative explanation, our system will determine the relevant target, linking, and base networks, and organize the knowledge in the linking and target networks in a manner that facilitates their integration into the base network.

Opportunism is the second way that our explanation generator will achieve flexibility. The system will actively seek opportunities to educate the user about concepts in the domain that are closely related to the topic being explained but are unknown to the user. Moreover, rather than interjecting this discussion in the middle of another topic, the system can relocate it to an appropriate place in its explanation.

Interruptability is the third way that our explanation generator will achieve flexibility. Interruptability presents a significant problem for an explanation generator. By responding to the user's question in the middle of an explanation, the system may need to radically change its plan for completing the explanation. For example, its response to the user's question may obviate the need to introduce concepts that were to appear later in the explanation, or it may allow the system to promote integration by changing the remainder of the explanation to mesh with the response.

We have built a prototype system which provides limited forms of integrative explanations, opportunism, and interruptability. We have used this system to produce explanations from portions of the Biology Knowledge Base. Because the system is not restricted to schemas. it generates different explanations for different users. This semester we will attempt to use the Penman system. for converting explanation structures into English. Penman, which has been in development at the University of Southern California's Information Sciences Institute for the past seven years. employs one of the largest machine grammars ever constructed

4 Research Supervised by Robert F. Simmons

4.1 Device Descriptions

Research Assistants: C. Chee, M. LaPolla, W. Lee, L. Obrst, D. Throop, K. Vanderwilt.

The goal of this project was to develop methodologies for producing device simulations that could display their operation interactively, generate English descriptions, answer questions. and provide mixed-initiative teaching and diagnostic capabilities. The project was directly concerned with computational models and animations of ordinary mechanical devices as described in "How Things Work" and other technical encyclopedias. Initial work on the Xerox Dandelion included animation and logic modelling of a two cycle gasoline engine. This experiment showed the necessity of using a work station with a larger memory and a faster CPU so we moved our efforts to a Symbolics 3600 Lisp machine. A four-cycle gasoline engine was animated and it proved satisfactory at the human interface, providing teaching capabilities for part names and cycle functioning. Techniques for integrating qualitative simulation methods with the control of animation of a piston pump were devised by Stuart Laughton and published as AI-85-19. W. Lee and C. Chiu studied the use of higher order derivatives to reduce ambiguities in qualitative simulation. A prototype general animation package was developed by Chin Chee as his MA Thesis, AI-TR 86-38, and simulations of a thermostat controlling a heat pump were studied. The four-cycle gasoline engine was represented as a semantic network that could be interpreted as a program that would actually run the simulation.

At this point Professor Kuipers had joined the faculty and had developed a strong group in qualitative simulation, so we changed our emphasis from the actual simulations to the understanding of the texts describing the devices. In subsequent years we broadened this goal to representation and understanding of a range of descriptive natural language texts.

Over the past years, our concerns with self-explanatory devices have focussed on semantic network representation of dynamic and other processes, and the production and analysis of natural language descriptions of them.

4.2 Representation

Research Assistants: Yeong Ho Yu, Jung Yung Seo

By adding to semantic-network representations such arc-names as sequence, variable, assign and arithmetic and Boolean operators, Yu showed that a device can be simulated, as well as described, and an interpreter can actually run the network as a simulation program. Our hope is that eventually we can develop procedures to analyze textual descriptions of processes into such representations, including the simulation model of the process.

Seo has developed a method for representing the multiple parse trees of ambiguous English sentences in the form of a syntactic graph. This representation is remarkably concise, and it has the further advantage of showing each point at which an ambiguity occurs. Current research with syntactic graphs has used them in semantic analysis to result in a comparable, still ambiguous form: the semantic graph. This research is almost unique in accepting the undeniable fact that natural language is an inherently ambiguous medium.

4.3 Text Analysis

Research Assistants: Yeong Ho Yu, Jung Yung Seo, Hae Chang Rim.

Yu has developed a method of compiling grammars into a network form that can parse sentences using a technique of multiple, parallel marker-passing. Each node in the network can recognize what to do with each of several different classes of markers. The result is a parallel parsing system that could run on parallel computers. Most recently he has included grammars for case and coherence analysis using the same parallel marker-passing approach. Yu has completed his doctoral dissertation describing these processes.

Seo has programmed an all-paths, bottom-up parser, guaranteed to parse sentences in no more than n-cubed cost. The output of this system includes the syntactic graphs mentioned in the previous section.

Seo used this parser and semantic and discourse analysis methods to compute discourse structures for expository text in a parallel fashion, using a dynamic semantic network architecture to compute a best interpretation for the text. He is particularly concerned to compute a graph of surface, discourse organization. He has completed his dissertation, entitled "Text Driven Construction of Discourse Structures for Understanding Descriptive Texts" and received his PhD in the summer of 1990.

Rim has devised programs for computing outline structures for disease description texts. The basis for this work is the use of rule forms for the identification of the type of schema slot that each clause signifies: definition, cause, symptom, treatment, side-effect, etc. The connectivities between sentences, i.e. the focal relations, are used to organize the analyzed text into outline form. Using the focal relations without schemas, it is still feasible to compute outlines from the semantic graphs. His methods have been tested on several multiple paragraph descriptive texts, and he has completed his dissertation to receive his PhD in December.

4.4 Text Generation and Revision

Research Assistants: Wing K. C. Wong, Loren Terveen

Wong uses general schematic planning techniques to select a set of propositions to communicate from the representation. A simple-sentence grammar is applied to each proposition to transform it to an English simple sentence. Then, successive phases of revision are applied to the set of such sentences, combining simple sentences into stylistically sophisticated forms. Revision in general is a process of reducing redundancy of expression by transforming full sentences into modifier forms or embedded expressions. Techniques described in the recent literature are used to achieve stylistic and pragmatic goals of communication.

Terveen has studied the idea of conversation as a collaborative process of mutual goal achievement. He has developed a prototype in which a computer system and a user engage in cooperative conversations in such a manner that both participants model each other's understanding at each stage in the collaborative communication. He is currently working at MCC and has published two papers describing his techniques in Human Factors conferences.

4.5 Neural Network Explorations

Research Assistants: Mark Ring, Yeong-Ho Yu

Mark Ring has implemented a theory of network learning first described by William James. and is experimenting with it in the environment of a robot explorer, realized in a full color. neural network, computational system on a Hewlett-Packard workstation. The simulated robot is taught to avoid obstacles and to find target locations. Ring has now developed his theory to account for hierarchical concept learning and the resulting use of concepts to control actions as neural network subroutines. He has won a NASA Research Fellowship to continue this research which he is proposing as his doctoral dissertation.

Yu has developed a neural net computation system for the Symbolics in Common Lisp. He has introduced the threshold constant *epsilon*, a number between 0 and 1 that determines if a match is sufficiently good to cease back-propagation on a particular pattern. He then invented the technique called *descending epsilon* to significantly improve the correctness ratio or percent of patterns correctly recognized at the end of a learning session. He also introduced a method of using redundant output to help guide the formation of weights between the input patterns and hidden units. Both of these new techniques have shown improved generalization capabilities when the resulting nets are tested with novel patterns. They were described and published at the San Diego Neural Network Conference.

Our most recent work has been to solve a long-standing problem in the use of neural nets for syntactic analysis of language: how to parse indefinitely long and deeply embedded sentences in the restricted fixed-size matrices representing neural nets. This work was published at the 5th Rocky Mountain Artificial Intelligence Conference at Las Cruces in June. As an important by-product of this work, we also have developed a most effective system for grammar acquisition - one that greatly simplifies and helps the linguistic effort required to develop a grammar We've also studied improved methods for accomplishing linguistic case analyses with neural net techniques as described in the Tech Report by Leow and Simmons [AI90-129].

Our general goal is to develop connectionist systems to deal effectively with sequential, language data and to learn appropriate behavior in recursively embedded, natural language environments. Some of the crucial work such as Ring's and Leow's will be continued at the Microelectronics and Computer Technology Corporation (MCC) in Austin.

5 Research Supervised by Vipin Kumar [2]

5.1 Design, Analysis, and Implementation of Parallel Algorithms

At the current stage of technology, it is possible to construct cheap but extremely powerful parallel computers by simply connecting a large number of sequential computers. But utilizing the massive power of these systems has been very difficult. One reason for this difficulty is that conventional algorithms and data structures were designed for sequential machines. There is much research needed to be done on algorithms and data structures for parallel computers. The goal of our research is to develop parallel algorithms and data structures for solving AI problems that can be modeled in terms of tree/graph search and to analyze their performance and scalability on various parallel architectures.

We have chosen to work on search problems because search is central to many problem-solving methods used in combinatorial optimization, pattern recognition, computer vision, speech recognition, decision making, planning, theorem proving and computer aided design. Most practical problems solved by search techniques are highly computation-intensive, and many such problems have to be solved in real time. Hence there is a great need for implementing search on parallel hardware. Search problems contain control-level parallelism as opposed to data-level parallelism, hence are more difficult to parallelize. This led many researchers to (incorrectly) believe that search problems have only a limited amount (less than one order of magnitude) of parallelism.

Many of the problems solved by search algorithms are NP-complete. Hence it may appear that there is no point in applying parallel processing to these problems, as the run time can never be reduced to a polynomial unless we have exponential number of processors. But the average time complexity of search algorithms on some problems is polynomial. For such problems, parallel processing can significantly increase the size of solvable problems. Many approximate branch and bound algorithms are known to run in polynomial time and their parallel versions can significantly increase the set of problems solvable in practice. Some applications using search algorithms (e.g. speech recognition, target recognition) require real time solutions. For these applications, parallel processing is perhaps the only way to obtain acceptable performance. For some problems, optimal solutions are highly desirable (e.g. floor plan optimization). For some decision problems (e.g. Test Generation, Circuit Verification), search is the only known solution.

We have formulated a number of parallel algorithms for solving problems that can be formulated as state-space search and implemented them on commercially available multiprocessors such as Sequent Balance 21000, Intel Hypercube, BBN Butterfly, Symult 2010, and Ncube/10. These parallel formulations were tested in the context of abstract problems such as the 15-puzzle problem, N-queens, the traveling salesman problem, as well as on practical problems such as optimizing floorplan of a VLSI chip, generating test pattern for a

[2]Dr. Kumar moved to the University of Minnesota in September, 1989. His student, Nageshwara Rao Vempaty (V. N. Rao), received his Ph.D. in August, 1990.

combinational circuit, and tautology verification. For many of these problems, we were able to demonstrate linear speedup up to 1024 processors (the largest configuration available) of Ncube/10, and up to 128 processors (the largest configuration available) on Intel Hypercube, Symult 2010, and BBN Butterfly. In this research, we developed concurrent data structures and scalable load balancing techniques that are useful in developing parallel formulations of AI heuristic search algorithms as well as other algorithms.

In particular, we have developed a parallel formulation of depth-first search which retains the storage efficiency of sequential depth-first search and can be mapped on to any MIMD architecture. This formulation is very generally applicable, and can also be used to exploit OR-parallelism in a logic programming language such as Prolog. At the heart of this parallel formulation is a dynamic load balancing scheme that divides the work between different processors. The effectiveness of the parallel formulation is strongly influenced by the load balancing scheme and architectural features such as presence/absence of shared memory, the diameter of the network, relative speed of the communication network, etc. We have analyzed these relationships and have determined characteristics that would make a multiprocessor ideally suited for this kind of parallel search.

We have also developed a data structure called *concurrent heap* that is very useful in parallel formulation of best-first heuristic search algorithms such as A* or Branch-and-Bound. This data structure allows many processors to simultaneously manipulate a single priority queue. Using this data structure, we are able to speedup A*/Branch-and-Bound algorithms linearly on the BBN Butterfly multiprocessor for over 100 processors.

While a parallel algorithm can be implemented in principle on many architectures, various architectures differ significantly in performance and scalability. To analyze the scalability of parallel algorithms and parallel architectures, we have developed the concept of *iso-efficiency analysis*. The iso-efficiency analysis has also given us the understanding of architectural features needed for good performance of parallel algorithms. We are continuing our research on the development of new parallel formulations and data structures, and investigation of suitable parallel architectures for these formulations.

5.2 Parallel Architectures for Logic Programs

A major drawback of parallel computers is that programming them is much harder than programming sequential computers. This problem can be circumvented if we write programs in a high-level language which expresses parallelism implicitly, and then map the programs automatically to a parallel architecture. This approach makes parallel processors just as easily usable as sequential computers.

Many high-level languages (such as purely functional languages, Horn-clause logic languages) are capable of expressing parallelism implicitly. We have chosen to investigate logic languages for the following reasons. Logic programs, unlike functional programs, are capable of expressing parallelism due to nondeterminism, i.e., OR-parallelism. Prolog (which is a logic programming language) is increasingly being accepted as a language for artificial intelligence (AI) programming. The execution of a logic program can be viewed as searching an AND/OR graph for a solution tree; therefore our ongoing research on sequential and parallel heuristic search is quite relevant to exploiting parallelism in logic programs. Furthermore, many problems and issues in exploiting parallelism from high-level languages are common no matter what language is used.

Potential parallelism in a logic program can be extremely large. A given sequence of subgoals B1,...,Bn can be solved by solving each subgoal Bi in parallel (AND-parallelism). Solution of a subgoal Bi can be attempted by trying all possible rules in parallel whose heads match Bi (OR-parallelism). Parallelism in AI programs due to search can be easily mapped to OR-parallelism. On the other hand, AND-parallelism refers to parallelism due to problem reduction, and is prevalent in AI and non-AI problems alike. A straightforward exploitation of AND- and OR-parallelism can generate an extremely large number of concurrent activities many of which may be redundant. For example, if two subgoals, B1 and B2 share variables, then many incompatible solutions of B1 and B2 may be generated. Since a practical parallel architecture can only have a finite amount of resources, it is necessary to develop an execution model which creates parallel activities in a judicious manner. A large number of parallel execution models for logic programs have appeared in the literature. Even after we have determined the execution model for exploiting parallelism, it is necessary to develop an efficient implementation of this model on a parallel architecture. In our research we are investigating various ways of exploiting parallelism in logic programs and architectures for implementing them.

5.2.1 Research on AND-Parallel Execution of Logic Programs

Exploiting AND-parallelism is hard due to the possibilities of binding conflicts and backtracking. A number of logic programming languages (e.g., Concurrent-Prolog, PARLOG, GHC) have been developed that deal with these problems by abandoning the backtracking feature of logic programming, and by requiring the programmer to explicitly specify dependencies between literals (via read-only annotations or mode declarations) to deal with binding conflicts. These languages, although quite useful for concurrent programming, change the semantics of logic programs by excluding "don't-know nondeterminism". Our research is concerned with "conventional" logic programs (i.e., ones that retain "don't-know nondeterminism"). We handle the problem of binding conflicts by dynamically detecting the dependencies among literals.

A number of solutions have been proposed to determine dependencies among literals of a clause. The early solution proposed by Conery and Kibler uses an ordering algorithm to determine dependencies at run time, but incurs substantial overhead. In response, other schemes were proposed by Chang, *et al.*, DeGroot, and Hermenegildo. These schemes sacrifice the degree of parallelism to reduce the run-time overhead. We have developed an execution model that uses tokens associated with shared variables to do the dependency analysis dynamically. We have also shown that this model provides roughly the same degree of parallelism as Conery's Model, and provides more parallelism than the schemes of Chang *et al.*, and DeGroot. Our model also performs more accurate intelligent backtracking than the ones presented earlier. We have developed an implementation of a slightly simplified version of this execution model. In this implementation tokens are represented in terms of bit vectors. The bit-vector implementation makes it possible to perform "approximate" dynamic dependency analysis and intelligent backtracking at low cost.

The goal of any parallel implementation for executing logic programs is to gain speedup over the best sequential implementation. The Warren Abstract Machine (WAM) has been recognized as the fastest and the most efficient sequential implementation for years. An implementation which is significantly different from the WAM can be an order of magnitude slower. Therefore, it is important to incorporate AND-parallelism in the WAM in such a way that most of its memory management efficiency and performance optimizations are retained. We have incorporated our bit-vector implementation in the WAM and tested its performance on Sequent Balance 21000, a shared-memory multiprocessor. Experimental results show that, for suitable programs, our parallel implementation can achieve linear speedup on dozens of processors. To the best of our knowledge, our implementation is the first actual WAM-based implementation of an AND-parallel execution model on parallel hardware.

5.2.2 Research on Intelligent Backtracking

When subgoals share variables, failure of a subgoal may mean that some other subgoal needs to backtrack and generate a new solution. This can be done easily (by backtracking to the subgoal to the left) if the subgoals are solved strictly sequentially. But when AND-parallelism is being exploited, the backtracking decision becomes nontrivial. An early solution developed by Conery and Kibler is incomplete, as it can miss solutions while performing backtracking. We have developed an algorithm for backtracking which is complete and is more intelligent than the ones previously proposed for AND-parallel execution. Our bit-vector implementation of this algorithm requires minimal run-time overhead. We have also developed a modification of this scheme to provide cost-effective intelligent backtracking in a sequential logic programming language such as Prolog. Dependency-directed backtracking has been investigated by many researchers, but in its full generality it tends to have tremendous overheads. Our intelligent backtracking scheme for Prolog can be viewed as its special case that can be efficiently implemented.

6 Research Supervised by Robert L. Causey

6.1 EVID: A System for Interactive Defeasible Reasoning

A "defeasible rule" is one that permits the inference of a conclusion from a conjunction of supporting evidential statements provided that this inference is not blocked (defeated) by other statements of defeating conditions (defeaters). There may be many different conjunctions of supporting evidence for a given conclusion. and there may be different defeaters that can block inferences of that conclusion from one or more of the conjunctions of supporting evidence. A given conclusion statement, say, concl(a,b,c), will be inferable or not according to the various combinations of evidence and defeaters that happen to hold. so defeasible reasoning is "nonmonotonic".

I have developed of a system, EVID, for defeasible reasoning (currently implemented in Prolog). Its reasoning behavior is similar in many respects to that of other nonmonotonic systems, when it is applied to rule sets for standard examples such as inheritance hierarchies with exceptions. Yet, the design and implementation are novel in several respects. A major aim of the project is to demonstrate that useful defeasible reasoning can be achieved without resorting to nonstandard logical connectives or rules of inference, so the design of EVID avoids such features. Instead, the deductions are guided by a few meta-predicates using standard Prolog with negation by failure.

Another major design goal is to facilitate user interaction, and much of the current program consists of a "logical interface" for this purpose. For instance, subject to some constraints that EVID enforces on the user's self-consistency, the user is permitted to defeat (partially for particular evidence, or totally for all evidence) most defeasible conclusions EVID infers from the user's knowledge base. EVID provides the user advice and justifications based on the user's own rule base and current data, but the user can choose to override this advice where appropriate. EVID should be especially useful for the development of decision support systems, and it may be viewed as a step towards the design of a "logical spreadsheet".

The first version of the EVID shell has been implemented and tested on many types of example problems and several types of small application programs. Recent research has also involved some detailed comparisons of the knowledge representation and reasoning capabilities of EVID with some other systems for nonmonotonic reasoning. In addition, Mr. Y. U. Ryu, graduate student in the Department of Management Science and Information Systems, is attempting to use EVID to model reasoning about norms and actions in bureaucracies. Other applications are expected elsewhere in 1991.

Review of Educational Programs

Robert L. Causey, Associate Director
Professor of Philosophy

The contract between the Army Research Office and the University of Texas included support for both basic research and AI education. The latter included graduate and post-doctoral education at the university, as well as the presentation of a variety of short courses for Department of the Army personnel. These courses began in the fall of 1984, and were taught by several faculty with the help of a number of graduate assistants. Some of the short courses were one-day overviews, some were three-day seminars on special technical subjects, and some were week-long courses on fundamental topics in AI.

Summary of Short Course Activities

The following are the short courses presented during FY 1985-89.
Total courses taught: 29.
Total number of attendees: about 935.

Fiscal Year 1985

Problem Solving Search and Knowledge Representation, Ft. Monmouth, 13-15 Nov 1984,
Attendance: 110 (est.). Presenter: Gordon Novak.

The Uses and Applications of Expert Systems, H. Diamond Labs, 30 Nov 1984,
Attendance: 35 (est.). Presenters: Gordon Novak and Robert Causey.

The Uses and Applications of Expert Systems, Ft. Leavenworth, 6 Feb 1985,
Attendance: 30 (est.). Presenter: Gordon Novak.

Problem Solving Search and Knowledge Representation, Redstone Arsenal, 25-27 Feb 1985,
Attendance: 150 (est.). Presenters: Gordon Novak and Vipin Kumar.

Introduction fo Artificial Intelligence Techniques, Univ. of Texas, 25-29 Mar 1985,
Attendance: 45. Presenters: Gordon Novak, Robert Causey, Vipin Kumar, Robert Simmons, and Bruce Porter.

LISP Programming, Univ. of Texas, 8-12 July 1985,
Attendance: 16. Presenters: Gordon Novak, Bruce Porter, and Vipin Kumar.

Fiscal Year 1986

Survey of Artificial Intelligence, AMC-HQ, 7 Oct 1985,
Attendance: 22 (est.). Presenters: Gordon Novak and Robert Causey.

Design and Construction of Expert Systems, BRL, 13-15 Nov. 1985, Attendance: 22 (est.). Presenter: Gordon Novak.

LISP Programming, Univ. of Texas, 6-10 Jan 1986,
Attendance: 23. Presenters: Gordon Novak, Robert Simmons, Vipin Kumar and Bruce Porter.

Introduction to Expert Systems, ALMC, Ft. Lee, 14-16 April 1986,
Attendance: 34 (est.). Presenter: Bruce Porter.

Survey of Artificial Intelligence, Ft. Harrison, 18 April 1986,
Attendance: 110 (est.). Presenters: Gordon Novak and Robert Causey.

Knowledge Acquisition and Knowledge Representation, USMA, West Point, 14-18 July 1986,
Attendance: 42. Presenters: Robert Causey and Bruce Porter.

Fiscal Year 1987

Introduction to Expert Systems, Ft. Detrick, 20-22 Oct 1986,
Attendance: 28. Presenter: Gordon Novak.

New Machine Architectures for Parallel Processing (with the University of Pennsylvania, Principal), Univ. of Pennsylvania, 9-11 Dec 1986,
Attendance: 6 (est.) Presenter from The University of Texas: Vipin Kumar.

LISP Programming, Univ. of Texas, 5-9 Jan 1987,
Attendance: 25. Presenters: Gordon Novak, Bruce Porter, and Robert Causey.

Overview of Machine Learning, Ft. Monmouth, 27-29 April 1987,
Attendance: 30 (est.). Presenter: Bruce Porter.

Knowledge Acquisition and Knowledge Representation, Univ. of Texas, 3-7 Aug 1987,
Attendance: 10. Presenters: Bruce Porter and Robert Causey.

<u>Fiscal Year 1988</u>

Natural Language Understanding and Machine Translation, Univ. of Texas, 6-8 Nov 1987,
Attendance: 14. Presenters: Robert Simmons, Gordon Novak, and Prof. Aravind Joshi of the University of Pennsylvania.

Computer Architectures for Parallel Processing in AI Applications (with the University of Pennsylvania, Principal), Univ. of Pennsylvania, 14-16 Dec 1987,
Attendance: 8 (est.). Presenter from The University of Texas: Vipin Kumar.

Knowledge Acquisition and Knowledge Representation, Univ. of Texas, 4-8 Jan 1988,
Attendance: 6. Presenters: Bruce Porter and Robert Causey.

Overview of Machine Learning, Army Research Institute, Alexandria, VA, 21-23 March 1988,
Attendance: 15. Presenter: Bruce Porter.

Introduction to Expert Systems, Rock Island, IL, 6-8 April 1988,
Attendance: approx. 40. Presenter: Gordon Novak.

Object-Oriented Programming in Lisp, Univ. of Texas, 18-22 July 1988.
Attendance: 10. Presenters: Gordon Novak and Robert Causey.

<u>Fiscal Year 1989</u>

Intro. to Parallel Architecture and Processing in AI Applications, (with the University of Pennsylvania, Principal), Univ. of Pennsylvania, 17-20 Oct 1988,
Attendance: 8. Presenter: Vipin Kumar.

Knowledge Acquisition and Knowledge Representation, Univ. of Texas, 9-13 Jan 1989, Attendance: 14. Presenters: Robert Causey and Bruce Porter.

Intro. to Expert Systems, AIRMICS, Atlanta, 1-3 Feb 1989.
Attendance: 25. Presenter: Gordon Novak.

Overview of Machine Learning, PM TRADE, Orlando, 27 Feb - 1 Mar 1989.
Attendance: 20. Presenter: Bruce Porter.

Natural Language Understanding, BRL, 12-14 Apr 1989.
Attendance: 15. Presenters: Robert Simmons, Gordon Novak and Mitch Marcus (of U. Penn.).

Object Oriented Programming in Lisp, Univ. of Texas, 5-9 June 1989.
Attendance: 22. Presenters: Robert Causey and Gordon Novak.

Course Materials

The following materials have been developed for the University of Texas short courses offered during FY 85 through FY 89:

Problem Solving Search and Knowledge Representation, 178 pp.
Introduction to Artificial Intelligence Techniques, 420 pp.
Lisp Programming Lecture Notes, 130 pp.
Survey of Artificial Intelligence, 150 pp.
Design and Construction of Expert Systems, 267 pp. + Appendices
Knowledge Acquisition and Knowledge Representation, 320 pp. + References and Exercises.
Knowledge Acquisition and Knowledge Representation, Revised 1987 for hands on lab course, 279 pp. + New Lab Exercises and Appendices
Parallel Architectures for Artificial Intelligence, approx. 200 pp.
Overview of Machine Learning, 235 pp.
Object Oriented Programming in Lisp, approx. 200 pp.

Video-Taped AI Course

During 1986 six University of Texas faculty members developed and presented a complete technical course on A. I. and Expert Systems for the Hewlett-Packard Corp. This course includes:

Approx. 60 hours of video-taped lectures on A. I. theory and programming.
A large notebook based on the lectures.
Student exercises on logic, LISP programming, and expert systems.
Exercise program code and an Emycin-like expert systems shell.

These materials have been furnished to the Army Research Office for duplication and use in training of DOD personnel at no charge. These materials have been published commercially through Morgan Kaufmann Publishers, and sold for $9,000.00 per copy.

Assessment

The attendees were asked to return course evaluation questionaires at the conclusion of each short course. The local course arrangers (POC's) sent copies of the evaluations to the ARO, and the evaluations have always been very favorable. Since different POC's have used different evaluation questionaires, it is not easy to summarize the overall results. However, the great majority of returned questionaires give the courses ratings in the top two positions on five-point scales, and the written comments were generally very favorable.

Graduate Students and Visiting Scholars from the Department of the Army

MAJ J. Richmann
Ph. D. program; later transferred to OR

LTC A. Andrews
Ph. D. program

R. Machuca
Visiting Scholar, 1986
White Sands Missle Range

J. Hobbs
Visiting Scholar, AY87-88
Army Logistics Center, Ft. Lee

R. Hernandez
Visiting Scholar, AY88-89
USMA, West Point

Trips in Support of Army Activities

Dr. Gordon S. Novak, Director
Artificial Intelligence Laboratory

Austin, TX, Jan. 9-11, 1990, Review of Air Land Battle Management project at Lockheed.

El Paso, TX, November 15-17, 1988. Co-Chair Session at U.S. Army Symposium on Artificial Intelligence Research for Exploitation of the Battlefield Environment.

Raleigh/Durham, NC, April 4-8, 1988. ARO Steering Committee meeting.

Baltimore, MD, February 22-23, 1988. U.S. Army Tech Base Advisory Group meeting.

Rougemont, NC, May 3-6, 1986. Electronics Research Strategy Planning Workshop.

Ft. Worth, TX, October 12-13, 1986. 1986 Frontiers in Education Conference to present a lecture on: "The Academic Perspective".

Washington, DC, June 16-19, 1986. Army Workshop on Future Directions in Artificial Intelligence.

Raleigh/Durham, NC, March 18-19, 1986. Army Research Office meeting to discuss short course planning.

Washington, DC, July 15-18, 1985. Army Workshop on Expert Systems Applications in the Military.

Washington, DC, July 1-2, 1985. Attend the planning meeting for Army-sponsored Expert Systems Conference.

Aberdeen Proving Ground, MD, May 23-24, 1985. Meet with Ballistics Research Lab. personnel to discuss short courses, BRL's A.I. work.

Raleigh/Durham, NC, April 29, 1985. Army Research Office meeting to discuss short course planning.

Monterey, CA, September 17-19, 1984. Workshop on Common LISP, sponsored by DARPA.

Dr. Robert Causey, Associate Director
Artificial Intelligence Laboratory

Ft. Huachuca, AZ, October 17-19, 1988. Invited presentation, "AI Applications: Today and Tomorrow", 1988 TECOM AI Conference.

Raleigh/Durham, NC, April 4-8, 1988. ARO Steering Committee meeting.

Leesburg, VA, April 13-15, 1987. U. S. Army Information Science and Technology Assessment for Research Conference.

Raleigh/Durham, NC, March 23-25, 1987. ARO Steering Committee meeting to discuss short course planning.

Raleigh/Durham, NC, March 18-19, 1986. Army Research Office meeting to discuss short course planning.

Washington, DC, July 15-18, 1985. Armv Workshop on Expert Systems Applications in the Military.

Baltimore, MD, May 23-24, 1985. Meet with Ballistics Research Lab personnel to discuss short courses, BRL's A.I work.

Philadelphia, PA, May 8-9, 1985. University of Pennsylvania and ARO regarding short course planning.

Raleigh/Durham, NC, April 29, 1985. Army Research Office meeting to discuss short course planning.

STATUS OF AI GRADUATE STUDENTS

1984 - August, 1990

STUDENTS SUPPORTED BY ARO

NAME	DEGREE/DATE	PROF	STATUS/SUPPORT	THESIS/DISSERTATION TITLE
Acker, Liane		Porter	PhD Candidate	Generating Explanations from a Large-Scale Knowledge Base
Arvindam, Sunil				
Bareiss, E. R.	PhD 8/88	Porter	Assistant Professor at Northwestern University	Protos: A Unified Approach to Learning Classification and Knowledge Representation
Berleant, J. D.		Kuipers	PhD in progress /NSF	Using Partial Quantative Information in Qualitative Reasoning
Bhattacharjee, A.	MA 1986	Simmons	AI Company, New Haven, CT	Interpreting Narratives with Evaluation Statements
Blumenthal, B.	Ph. D. exp. 12/90	Porter	Accepted tenure-track position at Cambridge University 1/91	Using Metaphor and Analogy to Partially Automate the Design of Human Interfaces for Computer Applications
Branting, L. K.	Ph. D. exp. 12/90	Porter	Accepted tenure-track position at Univ. of Wyoming 1/91	The Role of Explanation in Reasoning from Legal Precedents
Brill, E. D.			PhD coursework ARO Fellowship 87-89	
Bulko, W.	Ph. D. 5/89	Novak	Employed at IBM, Austin, TX	Understanding Text with An Accompanying Diagram
Byun, Y. T.	Ph. D. 5/90	Kuipers	Employed as a Professor of Computer Science, Seoul, Korea	A Robust Qualitative Method for Robot Spatial Learning
Chang, R. J.		Novak	PhD candidate	Analyst's Apprentice: A Knowledge-Based Analysis Workbench

NAME	DEGREE/DATE	PROF	STATUS/SUPPORT	THESIS/DISSERTATION TITLE
Chee, C.	MA 1986	Simmons	Information Systems Institute of USC, Santa Monica, CA	Intuitionistic Modelling and Simulation of Mechanical Devices
Christian, J. M.		Boyer	ARO Fellowship 86-88	Completion, Compilation, and Unification
Collins, Tim			ARO Fellowship 88-90	
Cooper, S.		Bledsoe	PhD coursework/NSF ARO Fellowship 84-86	
Crawford, J. M.	Ph. D. 9/90	Kuipers	Employed as Post-Doc Fellow AI Lab, UT Austin	Access Limited Logic: A Language for Knowledge Representation
Day, C. Clifton			ARO Fellowship 88-90	
Dutton, J. E.		Browne	ARO Fellowship 85-86 Employed at Info. Research Assoc.	
Fales, Christina L.			ARO Fellowship 85-86	
Eilerts, Eric	BS 6/88	Porter	Employed as Computer Programmer AI Lab, UT Austin	
Gove, A. N.			PhD coursework ARO Fellowship 87-89	
Gupta, Anshul				
Hartman, E.	Ph. D. 12/90	Kuipers	PhD in progress	Automatic Control Understanding for Natural Programs
Hernandez, Robert			*Visiting Scholar, 1988-89* *West Point, NY*	
Hill, Frederick	MA 1985	Novak	Employed at Sandia National Labs, NM	Negotiated Interfaces for Software Reusability
Hobbs, Jeff			*Visiting Scholar 87-88* *Army Logistics Center, Ft. Lee, VA*	
Jones, Thomas W.				

NAME	DEGREE/DATE	PROF	STATUS/SUPPORT	THESIS/DISSERTATION TITLE
Karimi, E.	MA 1984	Simmons		Computing Discourse Conceptual Coherence: A Means to Contextual Reference Resolution
Kong, E.		Kumar		
Kook, H. J.	PhD 5/89	Novak	Assistant Professor at King Sejong University, Seoul, Korea	A Model-Based Representation Framework for Expert Physics Problem Solving
Korner, K.	PhD 1986	Kumar	Faculty at USC	An Intelligent Remote File Server
Kunichitham, R.	MS 1988	Kumar	Peritus, Inc., California	Parallel Heuristic Search in Multiprocessors
LaPolla, M.	MS (Linguistics)	Simmons	Employed at Lockheed, Austin	
Laughton, S.	MA 1985	Simmons	Schlumberger, Austin	Explanation of Mechanical Systems Through Qualtiative Simulation
Lee, Wan L.			MCC, Austin	
Lee, Wan. Y.			PhD coursework, ARO Fellowship 87-89	
Lee, Wood. L.		Kuipers	PhD candidate/NSF	Constraining Qualitative Simulation
Lee, X. S.		Novak	PhD in progress	Temporal and Spatial Analysis in Solving Physics Problems
Lester, J.		Porter	PhD candidate ARO Fellowship 87-89	
Levinson, B.	PhD 1985	Rich/ Simmons	Faculty at UC Santa Cruz	A Self Organizing Retrieval System for Graphs
Lin, Y. J.	PhD 1988	Kumar	Bell Communications Research, Morristown, NJ	A Parallel Implementation of Logic Programs
Machuca, Raul			*Visiting Scholar, Jan-Oct 1986 White Sands Missle Range, NM*	
Mallory, R. S.		Porter	PhD coursework ARO Fellowship 87-89	

NAME	DEGREE/DATE	PROF	STATUS/SUPPORT	THESIS/DISSERTATION TITLE
Martinez-Guerra, Juan		Simmons	Employed at MCC, Austin, Tx. Masters in progress.	
Mauladad, Y.	MS 1987	Novak	Unisys, Mission Viejo, CA	CALCLAB: An Interactive Environment for Scientific Numerical Problem Solving
Murray, K.	PhD exp 6/91	Porter	PhD in progress	Learning as Knowledge Integration
Murray, W.	PhD 1986	Rich/ Porter	FMC Corp, Santa Clara, CA	Automatic Program Debugging for Intelligent Tutoring Systems
Needel, William I.	BS			
Obrst, L.		Simmons	MCC, Austin	
Pierce, David M.		Kuipers	Ph D Coursework	
Rajagopalan, R.		Kuipers	PhD coursework ARO Fellowship 87-89	
Richmann, MAJ. Jim			*PhD Program, transferred to O.R.*	
Rickel, Jeff		Porter	ARO Fellowship 88-90	
Rim, H. C.	Ph. D 12/90	Simmons	Accepted appointment as a Professor in Korean University	Computing Outlines from Descriptive Texts
Ring, Mark		Simmons		
Ross, J.	MA 1987	Porter	AI Group at Lockheed, Austin	Developing ES from Examples and Explanations
See, Hiow Tong		Novak	PhD in progress ARO Fellowship 84-86	Choosing Analysis Models for Expert Physics Problem Solving
Seo, J. Y.	PhD 8/90	Simmons	Accepted appointment as a Professor in Korean University	Text Driven Construction of Discourse Structures for Understanding Descriptive Texts
Shea, J. C.			ARO Fellowship 85-87	
Slepetis, Leona R.			ARO Fellowship 85-86	
Super, Boaz		Bovik	PhD coursework/NASA/ARO Fellowship 85-87	

NAME	DEGREE/DATE	PROF	STATUS/SUPPORT	THESIS/DISSERTATION TITLE
Temin, A. L.	PhD 1987	Rich/ Simmons	Software Productivity Consortium Reston, VA	Answering Questions about Program Behavior
Terveen, L.		Simmons	PhD in progress/employed at MCC	
Throop, D. R.	Ph. D exp. 12/90	Kuipers	PhD candidate/NSF	Quantative Operators in Design
Van Sickle, L.		Novak	Employed in Austin	
VanderWilt, J.		Simmons	Mead Data Company, Minneapolis, MN	
Vempaty, N. R.	Ph. D. 8/90	Kumar	Assistant Professor at Univ. of South Florida	Parallel Processing of Heuristic Search
Wan, M. L.	MA 1985	Novak	Hewlett Packard, Ft. Collins, CO	Menu-Based Creations of Procedures for Display of Data
Wang, C. K.	MS 1986	Mok	PhD in progress	A General Model for Performance Analysis of Parallel Logic Programs (MS)
Winghart, O.	MA 1985	Simmons	Employed at IBM Stockholm ARO Fellowship 85-87	Computational Treatment of Metaphor in Text Understanding: A First Approach (MA) Comprehending Simple Expository Paragraphs: A Recognition Model (PhD)
Wong, W. K.	MA 1986	Simmons	PhD in progress	GT: A Conjecture Generator for Graph Theory
Yu, Y. H.	MA 1984 PhD (exp.) 12/90	Simmons	PhD candidate	Understanding Text with Constrained Marker Passing

STUDENTS WITH OTHER SUPPORT

NAME	DEGREE/DATE	PROF	STATUS/SUPPORT	THESIS/DISSERTATION TITLE
Cohen, N.		Porter	PhD Candidate	
Duran, R.	MS 7/88	Porter	Employed at Southwestern Bell St. Louis, Mo.	Automatic Control Understanding for Natural Programs
Gibson, William H.	MA 1990	Causey		Clausewitz and the 21st Century
Holter, A.	MS 8/88	Causey		The *a priori* Status of Human Action
Hoverman, R. J., MD		Causey	PhD Candidate	Biological Basis of Moral Behavior
Martinich, L.	MS 1988	Kumar	Employed at IBM, Austin	A Parallel Implementation of the Alpha-Beta Search Algorithm
Mayes, R.		Causey	PhD Candidate	Toward a Cognitive Theory of Explanation
Petrie, C.	MA 1986	Simmons	Pursuing PhD/Employed at MCC	New Algorithms for Dependency- Directed Backtracking
Rath, C.	MA 1986	Novak	Employed at Bell Labs	A Rule Language for The GLISP Programming System
Reed-Lade, M.	MA 5/89	Novak	Employed at Lockheed, Austin	Grammar Acquisition and Parsing of Semi-Structured Text
Sayrs, B.		Novak	PhD candidate/ Employed at Lockheed, Palo Alto, CA	Reuse of Generic Algorithms for Specified Data Types
Tsukada, T.	MA 1987	Simmons	Returned to Hitachi, Japan	Using Dominator-Modifier Relations to Disambiguate A Sentence

FACULTY PUBLICATIONS
August 1984-August 1990

GORDON S. NOVAK, ASSOCIATE PROFESSOR AND DIRECTOR, AI LAB

Novak, G. S., Causey, R. L., et al. *Artificial Intelligence Project at The University of Texas at Austin*, University of Texas at Austin, Artificial Intelligence Laboratory AITR84-01.

Novak, G. S. "Artificial Intelligence and Expert Systems", College of Natural Sciences Newsletter, University of Texas at Austin, vol. 2, no. 1 (Spring 1984).

Novak, G. "Artificial Intelligence at the University of Texas at Austin", brochure describing AI work at University of Texas, 1984.

Novak, G. "Expert Problem Solving in Physics", *SIGART Newsletter*, No. 92, April 1985.

Novak, G. "Tutorial on LISP Programming", American Association for Artificial Intelligence, Menlo Park, CA, 1985. (also University of Texas at Austin, Artificial Intelligence AITR85-06, June 1985)

Novak, G. (ed.) "Artificial Intelligence Technical Reports, University of Texas at Austin, 1959-1985" Scientific Datalink, 1985.

Novak, G. TMYCIN Expert System Tool, University of Texas at Austin, Artificial Intelligence Laboratory AI87-52, April 1987.

Novak, G. "GLISP: A Lisp-Based Programming System with Data Abstraction", *Readings from the AI Magazine: 1980-1985* AAAI, 1987.

Novak, G., R. Causey, V. Kumar, J. Werth, B. Porter, V. Hwang, "Introduction to Artificial Intelligence and Expert Systems" (video-taped course with exercises), Morgan Kaufmann Publishers, 1988.

Novak, G. (with Hyung Joon Kook) "Representation of Models for Solving Real-World Physics Problems," *Proceedings of the 6th IEEE Conference on Artificial Intelligence Applications*, March 5-9, 1990, pp. 274-280. Also AI TR 88-93.

Novak, G. (with William Bulko) "Understanding Natural Language with Diagrams," National Conference on Artificial Intelligence (AAAI-90), Boston, MA, July 30-Aug 3, 1990., pp 465-470.

Novak, G. (with Hyung Joon Kook) "Representation of Models for Expert Problem Solving in Physics," accepted for *IEEE Transactions on Knowledge and Data Engineering*, about Dec. 1990.

BRUCE W. PORTER, ASSOCIATE PROFESSOR OF COMPUTER SCIENCE

Porter, B. "Learning Problem Solving: A Proposal for Continued Research", University of Texas at Austin, Artificial Intelligence Laboratory AITR85-03, 1985.

Porter, B. "Using and Revising Learned Concept Models: A Research Proposal", University of Texas at Austin, Artificial Intelligence Laboratory AITR85-05, 1985.

Porter, B. and Kibler, D. "A Comparison of Analytic and Experimental Goal Regression for Machine Learning", *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)*, 1985.

Porter, B. "Using and Revising Learned Concept Models"., *Proceedings of the Third International Machine Learning Workshop*, Computer Science Department, Rutgers University, 1985.

Porter, B., Bareiss, R., and Farquhar, A. "Learning Domain Knowledge from Fragments of Advice", *Recent Progress in Machine Learning*, Kluwer Academic Publishers, 1985.

Porter, B. and Kibler, D. "Experimental Goal Regression: A Technique for Learning Search Heuristics". *Machine Learning*, vol 2, 1986 (Also University of Texas at Austin, Artificial Intelligence Laboratory AITR86-20.)

Porter, B. and Bareiss, R. "Protos: An Experiment in Knowledge Acquisition for Heuristic Classification Tasks". *Proceedings of the International Meeting for Advances in Learning*, Les Arcs, France, 1986. (Also University of Texas at Austin, Artificial Intelligence Laboratory AITR86-35).

Porter B. and Kibler, D. "Experimental Goal Regression. A Technique for Learning Search Heuristics", *Journal of Machine Learning*, vol. 1, no. 4, 1986. (Also University of Texas at Austin, Artificial Intelligence Laboratory AITR86-20)

Porter, B. "A Review of the First International Meeting on Advances in Learning", *Journal of Machine Learning*, vol. 2, no. 1, 1987. (Also University of Texas at Austin, Artificial Intelligence Laboratory AITR86-37)

Bareiss, E., Porter, B. and Weir, C. "Protos: An Exemplar-Based Learning Apprentice", *Proceedings of the Fourth International Workshop on Machine Learning*, University of California at Irvine, Morgan Kaufmann Publishers, 1987. (Also University of Texas at Austin, Artificial Intelligence Laboratory AITR87-53.)

Bareiss, E. and Porter, B. "A Survey of Psychological Models of Concept Representation", University of Texas at Austin, Artificial Intelligence Laboratory AITR87-50.

(with R. Bareiss and R. Holte) "Knowledge Acquisition and Heuristic Classification in Weak-Theory Domains," The University of Texas at Austin, Artificial Intelligence Laboratory Technical Report #AI89-96 (accepted for publication in the *Artificial Intelligence Journal*).

(with R. Bareiss and K. Murray) "Supporting Start-to-Finish Development of Knowledge-Based Systems" (to appear in the *Machine Learning Journal*).

(with K. Murray) "Controlling Search for the Consequences of New Information during Knowledge Integration" (to appear in the *Proceedings of the Machine Learning Workshop*).

(with R. Holte and L. Acker) "Concept Learning and the Problem of Small Disjuncts" (to appear in the *Proceedings of the International Joint Conference on Artificial Intelligence*).

(with A. Souther, J. Lester, and L. Acker) "Using View Types to Structure Knowledge for Intelligent Tutoring" (to appear in the *Proceedings of the 11th Cognitive Science Society Conference*).

(with A. Souther, J. Lester, and L. Acker) "Generating Explanations in an Intelligent Tutor Designed to Teach Fundamental Knowledge," *Proceedings of the 2nd Intelligent Tutoring Systems Research Forum*. (An expanded version to appear in *Intelligent Tutoring Systems*, H. Burns and J. Parlett, editors; Lawrence Erlbaum Associates, publishers.)

"Similarity Assessment: Computation vs Representation" (to appear in the *Proceedings of the 2nd Case-Based Reasoning Workshop*).

(with R. Bareiss and K. Murray) "Gaining Autonomy During Knowledge Acquisition," The University of Texas at Austin, Artificial Intelligence Laboratory Technical Report #AI89-97.

(with A. Souther, J. Lester, and L. Acker) "Generating Explanations in an Intelligent Tutor Designed to Teach Fundamental Knowledge," to appear in *Intelligent Tutoring Systems*, H Burns and J. Parlett, editors; Lawrence Erlbaum Associates, publishers.

(with A. Souther, J. Lester, and L. Acker) "Generating Explanations in an Intelligent Tutor Designed to Teach Fundamental Knowledge," to appear in *Intelligent Tutoring Systems*, H. Burns and J. Parlett, editors; Lawrence Erlbaum Associates, publishers.

(with Lester, J.), "Generating Integrative Explanations: A Delayed-Commitment Approach " To appear in *Proceedings of the AAAI Workshop on Explanation*, AAAI-90, Boston, July 1990.

(with Acker, L., Lester, J., Souther, A.), "Generating Coherent Explanations to Answer Students' Questions," in Burns, H. & Parlett, J. (eds.), *Intelligent Tutoring Systems: Evolutions in Design*. Hillsdale, NJ: Lawrence Erlbaum, in press.

(with Lester, J., Acker, L., Souther, A.), "Generating Presentations of Domain Knowledge," *Symposium for Knowledge-Based Environments for Learning and Teaching, AAAI Spring Symposium Series*, Stanford University, March 1990, pp. 41-45.

(with Souther, A., Lester, J., Acker, L.), "Accessing Information from a Large-Scale Botany Knowledge Base Designed for Multiple Tasks," *Symposium for AI and Molecular Biology, AAAI Spring Symposium Series*, Stanford University, March 1990, pp. 141-143.

ROBERT F. SIMMONS, PROFESSOR OF COMPUTER SCIENCE AND PSYCHOLOGY

Simmons, R. *Computations from the English*, Prentice Hall, 1984.

Simmons, R. *Logic Processing for Natural Language*, Japanese translation of *Computations from the English*, translated by Yoneharu Fujita and Naoyuki Okada, Japan, 1990.

Simmons, R. "The Costs of Inheritance in Semantic Nets", *Proceedings of COLING-84*, Stanford, CA, July 1984.

Simmons, R. "From Menus to Intentions in Man-Machine Dialogue", *Proceedings of the 1st Annual Conference on AI Applications*, 1984.

Simmons, R. "Computer Science and Medical Information Retrieval", University of Texas at Austin, Artificial Intelligence Laboratory AITR-85-13, 1985.

Simmons, R. "Device Description Project", *SIGART Special Issue on Process Modelling*, July 1985.

Simmons, R. "Technologies for Machine Translation", *Proceedings of the International Symposium on Machine Translation*, Tokyo, 1986. (Also University of Texas at Austin, Artificial Intelligence Laboratory AITR85-14.)

Rim, Hae-Chang, and Simmons, R. F. "Extracting Data Base Knowledge from Medical Text", University of Texas at Austin, Artificial Intelligence Laboratory AITR86-42, Dec. 1986.

Simmons, R. "Man-Machine Interfaces-Can They Guess What You Want?", *IEEE Expert*, vol 1, no. 1, Spring 1986.

Jin, Wanying, and Simmons, R. "Symmetric Rules for Translation of English and Chinese", *Computers and Translation*, vol. 1, no. 3. (Also University of Texas at Austin, Artificial Intelligence Laboratory AITR86-26.)

Jin, Wanying, and Simmons, R. "Questions Answering with Rhetorical Relations", *Proceedings of the 3rd IEEE Conference on Artificial Intelligence Applications*, February 1987. (Also available as AITR86-40.)

Simmons, R. "A Text Knowledge Base for the AI Handbook", *Information Processing and Management*, vol. 23, no. 4, Sept. 1987.

Simmons, R. and Seo, Jungyun, *Syntactic Graphs: A Representation for the Union of All Ambiguous Parse Trees*, University of Texas at Austin, Artificial Intelligence Laboratory AITR87-64, January 1988.

Simmons, R. and Yu, Yeong-Ho, *Constrained Marker Passing*, University of Texas at Austin, Artificial Intelligence Laboratory AITR88-76, April 1988.

(with Yeong-Ho Yu) "Passing with Constrained Marker Passing," The University of Texas at Austin, Artificial Intelligence Laboratory Technical Report #AI89-104.

(with Hae Chang Rim and JungYun Seo), "Transforming Syntactic Graphs into Semantic Graphs," December 1989, accepted for presentation at the 28th annual meeting of the *American Association for Computational Linguistics*, 6-9 June 1990.

(with Yeong-ho Yu), "Bi-Directional Interactions of Levels in Natural Language Understanding," accepted for presentation at the 28th annual meeting of the *American Association for Computational Linguistics*.

(with Yeong-ho Yu), "Descending Epsilon in Back-Propagation: A Technique for Better Generalization," submitted to *IEEE Conf. on Neural Networks*.

(with Yeong-ho Yu), "Extra Output Biased Learning," submitted to *IEEE Conf. on Neural Networks*.

(with Hae Chang Rim and JungYun Seo), "Transforming Syntactic Graphs into Semantic Graphs," *Proceedings of the 28th annual meeting of the Association for Computational Linguistics*, Pittsburgh, PA, 6-9 June 1990.

(with Yeong-ho Yu), "Bi-Directional Interactions in a Parallel Text Understanding System, Artificial Intelligence Laboratory Technical Report #AI90-131, 1990.

(with Yeong-ho Yu), "Descending Epsilon in Back-Propagation: A Technique for Better Generalization," *Proceedings of the IEEE Conference on Neural Networks*, San Diego, CA, June 1990.

(with Yeong-ho Yu), "Extra Output Biased Learning," *Proceedings of the IEEE Conference on Neural Networks*, San Diego, CA, June 1990.

(with Yeong-ho Yu), "Training a Neural Network to be a Context Sensitive Grammar, *Proceedings of the Fifth Rocky Mountain Artificial Intelligence Conference*, Las Cruces, NM, June 1990.

(with Wee-Kheng Leow), "A Constraint Satisfaction Neural Network for Case Analysis," Artificial Intelligence Laboratory Technical Report #AI90-129, March 1990.

(with Yeong-ho Yu), "Truly Parallel Understanding of Text," *Proceedings of the National Conference on Artificial Intelligence* (AAAI-90), 1990.

(with Yeong-Ho Yu) "Passing with Constrained Marker Passing," The University of Texas at Austin, Artificial Intelligence Laboratory Technical Report #AI89-104.

DR. ROBERT L. CAUSEY, PROFESSOR OF PHILOSOPHY

EVID: A System for Interactive Defeasible Reasoning, Artificial Intelligence Laboratory Technical Report #AI90-119, January 1990.

EVID: A System for Interactive Defeasible Reasoning (revised version of above TR), accepted for publication in *Decision Support Systems* (probably will appear in 1991).

VIPIN KUMAR, ASSISTANT PROFESSOR

Kumar, V. "Integrating Knowledge in Problem Solving Search Procedures", *Proceedings of the 1984 Annual Conference of the Association for Computing Machinery (ACM-84)*, October 1984.

Kumar, V. "Parallel Branch and Bound Formulations for And/Or Tree Search", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 6, No. 6, November 1984.

Kumar, V. "Thoughts on Parallel Processing," *Byte* (special issue on multiprocessing), Vol. 10, No. 5, May 1985.

Kumar, V. "A Generalization of the AO* Algorithm", *Proceedings of the IEEE Computer Society's Ninth International Computer Software & Applications Conference (COMPSAC-85)*, Chicago, October 1985.

Kumar, V. *A General Heuristic Bottom-up Procedure for Searching and or Graphs*, University of Texas at Austin, Artificial Intelligence Laboratory AITR85-08, August 1985.

Kumar, V., Nau, D., and Kanal, L. "A General Paradigm for And/Or Graph and Game Tree Search", *Search in Artificial Intelligence*, Springer Verlag, 1988. (Also University of Texas at Austin, Artificial Intelligence Laboratory AITR85-09, August 1985.)

Kumar, V. "Branch-and-Bound Search", *Encyclopedia of Artificial Intelligence*, John Wiley, 1986. (Also University of Texas at Austin, Artificial Intelligence Laboratory AITR-11, August 1985.)

Kumar, V. "Depth-first Search", *Encyclopedia of Artificial Intelligence*, John Wiley, 1986.

Lin, Y., Kumar, V. and Leung, C. "An Intelligent Backtracking Algorithm for Parallel Execution of Logic Programs". *Proceedings of the Third International Conference on Logic Programming*, London, July 1986 (Also University of Texas, Artificial Intelligence Laboratory AITR85-22, March 1985).

Lin, Y., Kumar, V. "A Parallel Execution Scheme for Exploiting And-parallelism of Logic Programs", *1986 International Conference on Parallel Processing*, 1986.

Kumar, V., Lin, Y. "Parallel Architectures for Logic Programming", *Proceedings of Future Directions in Computer Architecture and Software*, May 1986 (extended abstract).

Lin Y. and Kumar, V. "A Parallel Execution Scheme for Exploiting And-parallelism of Logic Programs", *Proceedings of 1986 International Conference on Parallel Processing*, Chicago, 1986. (Also University of Texas, Artificial Intelligence Laboratory AITR-23, March 1986.)

Kumar, V. and Lin, Y. "A Framework for Intelligent Backtracking in Logic Programs", *Proceedings of the Sixth Annual Conference on Foundations of Software Technology and Theoretical Computer Science*, New Delhi, India, December 1986. (Also University of Texas, Artificial Intelligence Laboratory AITR-41, August 1987).

Kumar, V. and Lin, Y. "An Intelligent Backtracking Scheme for Prolog", *Proceedings of the 1987 International Symposium on Logic Programming*, September 1987.

Rao, V., Kumar, V. and Ramesh, K. "Parallel Heuristic Search on a Shared Memory Multiprocessor", *Proceeding of MCC/ University Research Symposium*, July 1987.

Kumar, V. and Kanal, L. (eds.) *Search in Artificial Intelligence*, Springer Verlag, 1988.

Kumar, V. and Lin, Y. "A Data-Dependency Based Intelligent Backtracking Scheme for Prolog". *Journal of Logic Programming*, vol. 2, no. 2, 1988.

Lin, Y. and Kumar, V. "An Execution Model for Exploiting And-parallelism in Logic Programs". *New Generation Computing*, vol. 5, no. 4, 1988.

Kumar, V. "The CDP: A Unifying Formulation for Heuristic Search, Dynamic Programming, and Branch & Bound Procedures" *Search in Artificial Intelligence*, edited by L. Kanal and V. Kumar, Springer Verlag, 1987.

Kumar, V. and Y. Lin, "An Intelligent Backtracking Scheme for Prolog", *Proceedings of the 1987 International Symposium on Logic Programming*, September 1987, San Francisco.

Rao, V. N., Kumar, V. and Ramesh, K., "A Parallel Implementation of the IDA* Algorithm", *Proceedings of the National Conference on Artificial Intelligence*, July 1987, Seattle, Washington.

Kumar, V., Rao, V. N. and Ramesh, K. "Parallel Depth-First Search on Multiprocessors, Part I: Implementation", *Journal of Parallel Programming*, 1988.

Kumar, V., Rao, V. N. and Ramesh, K. "Parallel Depth-First Search on Multiprocessors, Part II: Analysis", *Journal of Parallel Programming*, 1988.

Rao, V. N., and Kumar, V. "Concurrent Insertions and Deletions in a Priority Queue", *IEEE Transactions on Computers*, vol. 37, no. 12, 1988.

Rao, V. N., and Kumar, V. , "Superlinear Speedup in Parallel Search", *Proceedings of the Foundations of Software Technology and Theoretical Computer Science*, December 1988, New Delhi.

Kumar, V., Rao, V. N., and Ramesh, K. "Parallel Heuristic Search of State-Space Graphs: A Summary of Results", *Proceedings of the National Conference on Artificial Intelligence (AAAI-88)*, 1988. (Also University of Texas, Artificial Intelligence Laboratory AITR88-70, 1988)

Kumar, V., Rao, V. N. and Ramesh, K. "Parallel Depth-first Search on the Ring Architecture", *Proceedings of the 1988 International Parallel Processing Conference (AAAI-88)*, 1988.

Rao, V. N., and Kumar, V. "Concurrent Insertions and Deletions in a Priority Queue", *Proceedings of the 1988 International Parallel Processing Conference*, 1988.

Kumar, V. *AND-parallel Execution of Logic Programs on a Shared Memory Multiprocessor*, University of Texas, Artificial Intelligence Laboratory AITR88-74, 1988.

(with Y. Lin and A. Gupta) "Parallel Execution of Logic Programs," *AAAI Spring Symposium on Representation and Compilation in High Performance Theorem Proving*, March 1989.

(with V. Rao) "Parallel Heuristic Search on Multiprocessors," *AAAI Spring Symposium on Planning and Search*, March 1989.

(with V. Rao) "Load Balancing in Message Passing Multiprocessors," *HCCA4 (Fourth Conf. on Hypercubes Concurrent Computers and Applications)*, March 1989.

(with S. Arvindam and V.N. Rao) "Floorplan Optimization on Multiprocessors," *Proceedings of ICCD-'89 (International Conference on Computer Design)*, October 1989.

(with S Arvindam and V. Nageshwara Rao) "Efficient Parallel Algorithms for Search Problems: Applications in VLSI CAD" Accepted for presentation at the *3rd Symposium on Frontiers of Massively Parallel Computation-90*, College Park, MD.

(with Sunil Arvindam and V. Nageshwara Rao), "Floorplan Optimization on Multiprocessors," In the *Proceedings of the International Conference on Computer Design*, October 89. Also submitted to *International Journal on VLSI Design.*

(with V. Nageshwara Rao), "On the Efficiency of Parallel Depth-First Search," Submitted to the *International Conference on Parallel Processing*, August 1990.

"A General Heuristic Bottom-up Procedure for Searching AND/OR Graphs," accepted for publication in *Information Science.*

(with Y. Lin) "AND-Parallel Execution of Logic Programs on a Shared-Memory Multiprocessor," accepted for publication in the *Journal of Logic Programming.*

(coeditors L. Kanal and P. S. Gopalkrishnan), *Parallel Algorithms for Machine Intelligence and Vision*, Springer-Verlag, March 1990.

STUDENT PUBLICATIONS
August 1984-August 1990

Note: Publications co-authored by students and faculty are listed under Faculty Publications

Alcouffe, P. *Metaphorical Shift and The Induction of Similarities*, Master's thesis, University of Texas at Austin, Artificial Intelligence Laboratory AITR86-30, July 1986.

Bareiss, R. and Farquhar, A., *Fault Diagnosis Using Qualitative Simulation*, University of Texas at Austin, Artificial Intelligence Laboratory AITR86-25, April 1986.

Bhattarcharjee, Annanya, *Intrepreting Narratives with Evaluative Statements*", Masters thesis, University of Texas at Austin, Artificial Intelligence Laboratory AITR87-47, January 1987.

Biswas, J. and Matula, D. "Two Flow Routing Algorithms for the Maximum Concurrent Flow Problem", *Proceedings of the Fall Joint Computer Conference*, Dallas, 1986.

Blumenthal, Brad, "Replaying Design Episodes," *AAAI Spring Symposium on Artificial Intelligence and Software Engineering Working Notes*, March 1989.

Blumenthal, Brad, "An Architecture for Automating Metaphoric Interface Design," *Computer-Human Interaction Conference, Doctoral Consortium*, April 30-May 1, 1989.

Blumenthal, Brad, "Replaying Design Episodes," appearing in the *Proceedings of IJCAI '89 Workshop on Automating Software Design*.

Blumenthal, Brad, "Replaying Metaphoric Interface Design Experience," appearing in the *Prcceedings of the IJCAI '89 Workshop on Advanced Human Interface Design*.

Blumenthal, Brad, "Replaying Metaphoric Interface Design Experience," (to appear in the *Proceedings of the IJCAI Workshop on Advanced Human Interfaces*).

B. Blumenthal, "Empirical Comparisons of Some Design Replay Algorithms," *Proceedings of the Eighth National Conference on Artificial Intelligence*, AAAI-90, Boston, MA, 1990.

B. Blumenthal, "Incorporating Metaphor in Automated Interface Design," (expanded version) *Proceedings of the Third IFIP Conference on Human-Computer Interaction*, Interact-90, Cambridge, England, 1990.

Branting, L. *A Prototype Natural Language Understanding Legal Consultant*. Artificial Intelligence Laboratory, University of Texas at Austin, AITR87- 49.

Branting, L. Karl, *The Role of Explanation in Reasoning from Legal Precedents*, UTA AITR88-78, April 1988.

Branting, Luther Karl, "Representation and Reuse of Explanations in Case-Based Reasoning," The University of Texas at Austin, Artificial Intelligence Laboratory Technical Report #AI89-101.

Branting, Luther Karl, "Integrating Generalizations with Exemplar-Based Reasoning," to appear in the *Proceedings of the Eleventh Annual Conference of the Cognitive Science Society*,

Ann Arbor, Mich., August 16-19, 1989. Appearing in condensed form in the *Proceedings of the Second DARPA Workshop on Case-Based Reasoning*, Pensacola, Florida, May 31-June 2, 1989.

Branting, Luther Karl, "Representing and Reusing Explanations of Legal Precedents," *Proceedings of the Second International Conference on Artificial Intelligence and Law*, Vancouver, B.C., June 13-16, 1989.

Branting, L. Karl, "Representing and Reusing Explanations of Legal Precedents", appearing in the *Proceedings of the Eleventh Annual Conference of the Cognitive Science Society*, Ann Arbor, Michigan, August 16--19, 1989.

K. Branting, "Building Explanations with Rules and Structured Cases," to appear in the *International Journal of Man-Machine Studies*, 1990.

K. Branting, "Techniques for Retrieval of Structured Cases," *AAAI Spring Symposium on Case-Based Reasoning*, Palo Alto, CA, March 27-29, 1990.

K. Branting, "Precedent Retrieval by Match Refinement," to appear in the *Proceedings of the AAAI Workshop on Law and Artificial Intelligence*, Boston, MA, 1990.

Bulko, William, "Understanding Text with an Accompanying Diagram", *Proceedings of the First International Conference on Industrial and Engineering Applications of A. I. and Expert Systems*, Tullahoma, TN, 1988, pp. 894-898. Also UTA AITR88-69, March 1988.

Bulko, William Charles, "Understanding Coreference in a System for Solving Physics Word Problems," Ph. D. dissertation, The University of Texas at Austin, Artificial Intelligence Laboratory Technical Report #AI89-102.

Chee, C. *Intuitionistic Modelling and Simulation of Mechanical Devices*, Master's thesis, University of Texas at Austin, Artificial Intelligence Laboratory AITR86-38, December 1986.

Day, Coy C., "Filling Gaps in Parsed English Sentences," Artificial Intelligence Laboratory Technical Report #AI89-113, September, 1989.

Flatau, A. A *Program for Translating English Sentences into Lisp Programs.*, Artificial Intelligence Laboratory, University of Texas at Austin, AITR87- 48.

Hill, Fredrick, "Negotiated Interfaces for Software Reusability". Master's Thesis, AI TR 85-16.

Kook, Hyung Joon, "A Model-Based Representational Framework for Expert Physics Problem Solving," The University of Texas at Austin, Artificial Intelligence Laboratory Technical Report #AI89-103

Korner, K. *An Intelligent Remote File Server*, PhD dissertation, University of Texas at Austin, Artificial Intelligence Laboratory, AITR86-39, December, 1986.

Mallory, Richard S., "Sources of Classification Accuracy in Protos," Artificial Intelligence Laboratory Technical Report #AI89-118, December, 1989.

Mauladad, Yusuf N. *CALCLAB: An Interactive Environment for Scientific Numerical Problem Solving*, Masters thesis, University of Texas at Austin, Artificial Intelligence Laboratory AITR87-59.

Murray, K.S. "Multiple Convergence: An Approach to Disjunctive Concept Acquisition", *Proceedings of the International Joint Conference on Artificial Intelligence*, August 1987. (Also University of Texas at Austin, Artificial Intelligence Laboratory AITR87-56).

Murray, W. "Talus: Automatic Program Debugging for Intelligent Tutoring Systems", *Computational Intelligence*, 1987. (Also University of Texas at Austin, Artificial Intelligence Laboratory AITR86-32).

Petrie, C. *New Algorithms for Dependency-Directed Backtracking*, Master's thesis, University of Texas at Austin, Artificial Intelligence Laboratory AI- TR-33, September 1986.

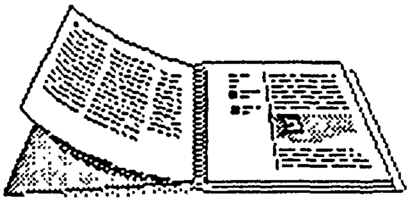
Reed-Lade, Margaret, "Grammar Acquisition and Parsing of Semi-Structured Text," (Master's Thesis) The University of Texas at Austin, Artificial Intelligence Laboratory Technical Report #AI89-105.

Ross, Joseph C., *Developing Expert Systems from Examples and Explanations*, University of Texas at Austin, Artificial Intelligence Laboratory, AITR87-63.

Wan, Mau-Lee, "Menu-Based Creation of Procedures for Display of Data", Master's Thesis, Technical Report AI 85-18, Dec. 1985.

Wang, C. *A General Model for Performance Analysis of Logic Programs*, Master's thesis, December 1986.

Wong, W. *A Theory of Argument Coherence*, University of Texas at Austin, Artificial Intelligence Laboratory AI-TR-29, July 1986.



Technical Reports Current Listing

Artificial Intelligence Lab University of Texas

• an order form is located at the end of the listings •

AI90-140 **An Assessment of Current Qualitative Simulation Techniques,** Pierre Fouche & Benjamin Kuipers, September 1990.

Qualitative modeling and simulation make it feasible to predict the possible behaviors of a mechanism consistent with an incomplete state of knowledge. Though qualitative simulation predicts all possible behaviors of a system, it can also produce spurious behaviors, i.e. behaviors which correspond to no solution of any ordinary differential equation consistent with the qualitative model In this paper we present and compare several significant techniques that attack this problem. They are. reasoning with higher-order derivatives, ignoring irrelevant distinctions, reasoning in the phase space representation, and reasoning about energy. The aim of this report is to provide a comprehensive view of all these techniques, by explaining their rationale, showing the problems they address and how they interact. Some remaining problems in qualitative simulation are also briefly discussed

AI90-139 **Reasoning about Energy in Qualitative Simulation,** Pierre Fouche & Benjamin Kuipers, September 1990.

Qualitative modeling and simulation make it feasible to predict the possible behaviors of a mechanism consistent with an incomplete state of knowledge. Though qualitative simulation predicts all possible behaviors of a system, it can also produce spurious behaviors which correspond to no solution of any ordinary differential equation consistent with the qualitative model. In this paper we present a method for reasoning about energy, which eliminates an important source of spurious behaviors. We apply this method to an industrially significant mechanism --- a non-linear, proportional-integral controller -- and show that qualitative simulation captures the main qualitative properties of such a system, such as stability and zero-offset control. We believe that this is a significant step toward the application of qualitative simulation to model-based monitoring, diagnosis, and design of realistic mechanisms.

AI90-138 **The Composition of Heterogeneous Control Laws,** Benjamin Kuipers and Karl Astrom, September 1990.

To design a control system to operate over a wide range of conditions, it may be necessary to combine control laws which are appropriate to the different operating regions of the system. The fuzzy control literature, and industrial practice, provide certain non-linear methods for combining heterogeneous control laws, but these methods have been very difficult to analyze theoretically. We provide an alternate formulation and extension of this approach that has several practical and theoretical benefits. First, the elements to be combined are classical control laws, which provide high-resolution control and can be analyzed by classical methods. Second, operating regions are characterized by fuzzy set membership functions. The global heterogeneous control law is defined as the weighted average of the local control laws, where the weights are the values returned by the membership functions, thereby providing smooth transitions between regions. Third, the heterogeneous control system may be described by a qualitative differential equation, which allows it to be analyzed by qualitative simulation, even in the face of incomplete knowledge of the underlying system or the operating region membership functions. Examples of heterogeneous control laws are given for level control of a water tank and for motion control of a mobile robot, and several alternate analysis methods are presented.

AI90-137 **Learning Hill-Climbing Functions as a Strategy for Generating Behaviors in a Mobile Robot,** David Pierce and Benjamin Kuipers, August 1990.

We consider the problem of a robot with uninterpreted sensors and effectors which must learn, in an unknown environment, behaviors (i.e., sequences of actions) which can be taken to achieve a given goal This general problem involves a learning agent interacting with a reactive environment: the agent produces actions that affect the environment and in turn receives sensory feedback from the environment. The agent must learn, through experimentation, behaviors that consistently achieve the goal. The difficulty lies in the fact that the robot does not know *a priori* what its sensors mean, nor what effects its motor apparatus has on the world.

We propose a method by which the robot may analyze its sensory information in order to derive (when possible) a function defined in terms of the sensory data which is maximized at the goal and which is suitable for hill-climbing. Given this function, the robot solves its problem by learning a behavior that maximizes the function thereby resulting in motion to the goal.

AI90-136 **Probabilities of Qualitative Behaviors from Probability Distributions on Inputs,** Daniel Berleant, July 1990, revised and updated August 1990.

Research on finding the different behaviors that an incompletely specified model can exhibit has concentrated on finding the plausible behaviors, and ruling out implausible ones. However, it would also be useful to estimate the probabilities of different behaviors. That is the goal of the present work

We show how to make inferences about the probabilities of the various qualitative behaviors a model could exhibit, when partial quantitative information in the form of intervals or probability distributions is given about values (such as initial values) of model variables.

The algorithm described does not make the assumptions that a model is linear or that probability distribution functions (pdfs) are of a particular class. It is demonstrated on a 1-tank model.

AI90-135 **Training a Neural Network to be a Context Sensitive Grammar,** Robert F. Simmons and Yeong-Ho Yu, July 1990.

A system is described for training a neural network to represent a context-sensitive grammar. The trained network is accessed by a shift/reduce parser to analyze indefinitely long, deeply embedded sentences. The parser is a brief sequential program that queries the trained network, manages pushdown stacks, and records states of the parse so that a final analysis may be shown. It is argued that this hybrid approach generalizes immediately to case analysis of indefinitely long sentences.

AI90-134 **Truly Parallel Understanding of Text,** Yeong-Ho Yu and Robert F. Simmons, July 1990.

Understanding a text requires two basic tasks: making inferences at several levels of knowledge and composing a global interpretation of the given text from those various types of inferences. Since making inferences at each level demands extensive computations, there have been several attempts to use parallel inference mechanisms such as parallel marker passing (PMP) to increase the productivity of the inference mechanism. Such a mechanism when used with many local processors, is capable of making inferences in parallel. However, it often poses a large burden on the task of composing the global interpretation by producing a number of meaningless inferences which should be filtered out. Therefore, the increased productivity of the inference mechanism causes the slow down of the task of forming the global interpretation and makes it the bottleneck of the whole system. Our system, TRUE, effectively solves this problem with the Constrained Marker Passing mechanism. The new mechanism not only allows the system to make necessary inferences in parallel, but also provides a way to compose the global interpretation in parallel. Therefore, the system is truly parallel, and does not suffer from any single bottleneck.

AI90-133 **Towards a Formalization of Access Limited Logic,** J.M. Crawford and Benjamin Kuipers, June 1990.

One of the fundamental problems in the theory of knowledge representation is the impossibility of achieving logical completeness and computational tractability, while maintaining expressive power. Fortunately, in most applications completeness is not necessary. What is necessary is that there be some comprehensible description of what deductions the system will be able to make. We present steps toward a theory of access-limited logic, in which access to assertions in the knowledge-base is constrained by semantic network locality relations. Where a classical deductive method or logic programming language would retrieve all assertions that satisfy a given pattern, an access-limited logic retrieves all assertions reachable by following an available access path. The complexity of inference is thus independent of the size of the knowledge-base and depends only on its local connectivity. Access-Limited Logic, though incomplete, still has a well defined semantics and a weakened form of completeness, *Socratic Completeness*, which guarantees that for any query which is a logical consequence of the knowledge-base, there exists a series of queries after which the original query will succeed. This paper presents an introduction to ALL and then a formal development of the soundness, Socratic Completeness, Partitional Completeness, and polynomial time complexity of ALL.

AI90-132 **Using Explanation-based and Empirical Methods in Theory Revision,** Dirk Ourston, March 1990.

This proposal presents an approach for automatic theory revision. In contrast to other systems, the approach is capable of modifying a theory which contains multiple faults and faults which occur at intermediate points in the theory. The approach uses explanations to focus the corrections to the theory, with the corrections being supplied by an inductive component. In this way the original structure of the theory is preserved to the maximum extent possible. Because the approach begins with an approximate theory , learning an accurate theory takes fewer examples than a purely inductive system. The approach also applies at any point in the expert system lifecycle when the expert system generates incorrect results. An initial prototype using the approach has been applied to the stock market domain, and shows significantly better results than a purely inductive learner.

AI90-131 **Bi-Directional Interactions in a Parallel Text Understanding,** Yeong-Ho Yu and Robert F. Simmons, March 1990.

Since understanding a natural language text requires a great deal of computation, there have been several attempts to use parallel inference mechanisms such as *parallel marker passing*. However, all those attempts suffer from an explosive number of inferences made at many independent local computational units. Finding the best inferences among those inferences requires a very costly filtering process which tends to be the bottleneck of the whole system. The main reason for such a problem is the lack of interactions among local processing units. In this paper, we present the bi-directional interactions in a parallel text understanding system which utilizes multiple levels of knowledge including syntax, semantics, and pragmatics. Each level interacts in two directions with adjacent levels to construct plausible inferences at the next higher level, while filtering out implausible ambiguities at the next lower level. As a result, the costly filtering step required in other systems can be eliminated and a truly parallel processing can be achieved.

AI90-130 **Descending Epsilon in Back-Propagation: A Technique for Better Generalization,** Yeong-Ho Yu and Robert F. Simmons, March 1990.

There are two measures for the optimality of a trained feed-forward network for the given training patterns. One is the global error function which is the sum of squared differences between target outputs and actual outputs over all output units of all training patterns. The most popular training method, back-propagation based on the Generalized Delta Rule, is to minimize the value of this function. In this method, the smaller the global error is, the better the network is supposed to be. The other measure is the correctness ratio which shows, when the network's outputs are converted into binary outputs, for what percentage of training patterns the network generates the correct binary outputs. Actually, this is the measure that often really matters. This paper argues that those two measures are not parallel and presents a technique with which the back-propagation method results in a high correctness ratio. The results show that the trained networks with this technique often exhibit high correctness ratios not only for the training patterns but also for novel patterns.

• an order form is located at the end of the listings •

AI90-129 **A Constraint Satisfaction Neural Network for Case Analysis,** Wee Kheng Leow and Robert F. Simmons, March 1990.

In this paper, we describe a neural network which performs case analysis using constraint satisfaction. Its design is based on a careful analysis of the computational requirements and constraints posed by the problem. This network is superior to those published in that it learns to capture rules similar to those used in symbolic systems. Consequently, it requires only minimal training to achieve good performance. Furthermore, its performance will not degrade as rapidly as the other networks when the problem size scales up.

AI90-128 **Extra Output Biased Learning,** Yeong-Ho Yu and Robert F. Simmons, March 1990.

One way to view feed-forward neural networks is to regard them as *mapping functions* from the *input space* to the *output space*. In this view, the immediate goal of *back-propagation* in training such a network is to find a correct mapping function among the set of all possible mapping functions of the given topology. However, finding a correct one is sometimes not an easy task, especially when there are local minima. Moreover, it is harder to train a network so that it can produce correct output not only for training patterns but for novel patterns which the network has never seen before. This so-called *generalization* capability has been poorly understood, and there is little guidance for achieving a better generalization. This paper presents a unified viewpoint for the training and generalization of a feed-forward network, and a technique for improved training and generalization based on this viewpoint.

AI90-127 **Transforming Syntactic Graphs into Semantic Graphs,** Hae-Chang Rim, Jungyun Seo, and Robert F. Simmons, March 1990.

In this paper, we present a computational method for transforming a syntactic graph, which represents all syntactic interpretations of a sentence, into a semantic graph which filters out certain interpretations, but also incorporates any remaining ambiguities. We argue that the resulting ambiguous graph, supported by an exclusion matrix, is a useful data structure for question answering and other semantic processing. Our research is based on the principle that ambiguity is an inherent aspect of natural language communication.

AI90-126 **CC: Component Connection Models for Qualitative Simulation - A User's Guide,** David W. Franke, Daniel L. Dvorak. January 1990.

The relation between part and whole is the key to describing the structure of a mechanism. Different modeling methods have different concepts of what should count as a "part" of a system, and how the parts should relate to each other. The mathematical, differential-equation-based approach to modeling taken in QSIM essentially says that the "parts" of a mechanism are the continuous variables that characterize its state, and their relations are mathematical constraints inherited from the physical structure of the system.

However, a physical system frequently consists of a set of components that relate through explicit connections (a form of description that is frequently more meaningful to a domain expert than the differential equations). This guide describes CC, a model-building program that accepts a component-connection description of a physical system and translated it to the qualitative differential equations of QSIM. CC provides facilities for component abstraction and hierarchical component definition, raising the level of abstraction for modeling via WSIM. Further, this component-connection paradigm provides the framework for information utilized in other model-based reasoning tasks such as diagnosis.

• an order form is located at the end of the listings •

AI90-125 **POS User's Manual,** David Throop, January 1990, revised and updated March 1990.

AI90-124 **QSIM Maintainer's Guide,** David Throop and the Qualitative Reasoning Group, January 1990, Revised and Updated April 1990.

AI90-123 **QSIM User's Manual,** Adam Farquhar and Benjamin Kuipers, January 1990.

AI90-122 **A Smooth Integration of Incomplete Quantitative Knowledge into Qualitative Simulation,** Benjamin Kuipers and Daniel Berleant. January 1990.

Qualitative and quantitative representations and inference methods provide alternate means for reasoning about the behavior of deterministic systems. The strength of qualitative reasoning is the ability to derive useful, though incomplete, conclusions from incomplete knowledge of the structure of a system. We show how quantitative information, even when very incomplete, can be integrated smoothly into the framework of qualitative reasoning.

Our algorithm, Q2, can draw more powerful conclusions than would be possible for a qualitative simulator alone, without sacrificing the expressive power and graceful degradation capabilities of qualitative simulation. Each qualitative behavior produced by QSIM implies a collection of algebraic equations defined over the terms appearing in the behavior description. In particular, landmark values are names for unknown real numbers, and so serve exactly as algebraic variables. Qualitatively distinct behaviors imply distinct sets of equations. The equations follow from the definitions of the qualitative constraints and fundamental theorems of the differential and integral calculus.

Incomplete knowledge of quantitative values, in the form of bounding intervals, can be propagated across the equations to produce either (a) a contradiction refuting the current qualitative behavior, or (b) a qualitative behavior description in which landmarks and other terms are annotated with quantitative ranges. We sketch the proof of soundness for Q2, discuss the use of mixed qualitative and quantitative reasoning for measurement interpretation, and present examples of model-based reasoning with QSIM and Q2 applied to diagnosis and design.

AI90-121 **Spatial Learning Mobile Robots with a Spatial Semantic Hierarchical Model,** Yung-Tai Byun. January, 1990.

The goal of this dissertation is to develop a spatial exploration and map-learning strategy for a mobile robot to use in unknown, large-scale environments. Traditional approaches aim at building purely metrically accurate maps. Because of sensorimotor errors, it is hard to construct accurately such maps. However, in spite of sensory and computation limitation, humans explore environments, build cognitive maps from exploration, and successfully path-plan, navigate, and place-find. Based on the study of human cognitive maps, we develop a spatial semantic hierarchical model to replace the global absolute coordinate frame used in traditional approaches. The semantic hierarchical model consists of three levels: control level, topological level, and geometrical level. The topological level provides the basic structure of the hierarchy.

At the control level, a robot finds places or follows travel edges which can be described by qualitatively definable features. The distinctive features allow development of distinctiveness measures. The robot uses these measures to find, with negative feedback control, the distinctive places by hill-climbing search algorithms, and the travel edges by edge-following algorithms. Distinctive places and travel edges are connected to build a topological model. This model is created prior to the construction of a global geometrical map. Cumulative location error is essentially eliminated while traveling among distinctive places and travel edges by alternating between the hill-climbing search control algorithms and the edge-following control algorithms. On top of the topological model, metrical information is accumulated first locally and then globally.

Using a simulation package with a robot instance, NX, we demonstrate the robustness of our method against sensorimotor errors. The control knowledge for distinctive places and travel edges, the topological matching process, and the metrical matching process with local geometry make our approach robust in the face of metrical errors. In addition to robust navigation at the control and topological levels, our framework can incorporate certain metrically-based methods and thus provide the best of both approaches.

AI90-120 **A Robot Exploration and Mapping Strategy Based on a Semantic Hierarchy of Spatial Representations,** Benjamin Kuipers and Yung-Tai Byun, January 1990.

We have developed a robust qualitative method for robot exploration, mapping, and navigation in large-scale spatial environments. Experiments with a simulated robot in a variety of 2D environments have demonstrated that our method can build an accurate map of an unknown environment in spite of substantial random and systematic sensorimotor error.

AI90-119 **EVID: A System for Interactive Defeasible Reasoning,** Robert L. Causey, January 1990.

I describe a system for interactive, automated defeasible reasoning. An application program using this system can infer a conclusion defeasibly from a conjunction of supporting facts together with an appropriate general rule. This particular inference of the conclusion might be defeated by additional facts, although other independent evidence could still support the conclusion on the basis of other rules The system has been implemented in Prolog, and includes an extensive logical interface that permits the user to interact with and override an application program's defeasible conclusions, subject to certain constraints on the user's consistency. Possible applications to decision support systems are described.

AI89-118 **Sources of Classification Accuracy in Protos,** Richard S. Mallory. December, 1989.

The exemplar-based learning program Protos has been trained to diagnose hearing disorders. This experiment was run to determine the relative contributions of various parts of its knowledge base to its classification accuracy. It appears that indexing knowledge (for the selection of appropriate examplars) is essential to accuracy but that matching knowledge (for the comparison of examplars with cases) is less important. This is consistent with the general theory of exemplar-based programs However, it may also be due to the peculiarities of the domain and Protos's training.

AI89-117 **Higher-Order Derivative Constraints in Qualitative Simulation,** Benjamin Kuipers, Charles Chiu, David T Dalle Molle & David Throop. October, 1989.

Qualitative simulation is a useful method for predicting the possible qualitatively distinct behaviors of an incompletely known mechanism described by a system of qualitative differential equations (QDEs). Under some circumstances, sparse information about the derivatives of variables can lead to intractable branching (or "chatter") representing uninteresting or even spurious distinctions among qualitative behaviors. The problem of chatter stands in the way of real applications such as qualitative simulation of models in the design or diagnosis of engineered systems.

One solution to this problem is to exploit information about higher-order derivatives of the variables We demonstrate automatic methods for identification of chattering variables, algebraic derivation of expressions for second-order derivatives, and evaluation and application of the sign of second- and third-order derivatives of variables, resulting in tractable simulation of important qualitative models.

Caution is required, however, when deriving higher-order derivative (HOD) expressions from models including incompletely known monotonic function (M+) constraints, whose derivatives beyond the sign of the slope are completely unspecified. We discuss the strengths and weaknesses of several methods for evaluating HOD expressions in this situation.

• an order form is located at the end of the listings •

We also discuss a second approach to intractable branching, in which we change the level of description to collapse an infinite set of distinct behaviors into a few by ignoring certain distinctions. These two approaches represent a trade-off between generality and power. Each application of these methods can take a position on this trade-off depending on its own critical needs.

AI89-116 **Abductive Explanation in Text Understanding: Some Problems and Solutions,** Hwee Tou Ng and Raymond J. Mooney, October, 1989.

Abduction is an important inference process underlying much of human intelligent activities, including text understanding, plan recognition, disease diagnosis, and physical device diagnosis. In this paper, we describe a language understanding system, called ACCEL, that is capable of constructing deep, causal explanations for natural language texts through the use of abduction. We describe some problems encountered using abduction to understand text, and present some solutions to overcome these problems ACCEL has been implemented and successful tested on a number of examples of expository as well as narrative text.

AI89-115 **Can Competitive Learning Compete? Comparing a Connectionist Clustering Technique to Symbolic Approaches,** J. Jeffrey Mahoney and Raymond J. Mooney, October, 1989.

Clustering unclassified data is a machine learning task with proven applications to real-world problems. This paper presents a comparison of Competitive Learning (a neural network based approach to data clustering) with established symbolic approaches. Some of the shortcomings of Competitive Learning are discussed, along with attempts at correcting these. The algorithm is extended to handle the performance task of missing feature prediction. Experimental results are compared with similar results of symbolic systems, such as Cluster/2 and Cobweb. In these experiments, Competitive Learning does not perform as well as its symbolic counterparts.

AI89-114 **Geometrical Motion Planning of Manipulators using a Continuous Curvature Model,** Benjamin Kuipers and Akira Hayashi, October, 1989.

We are trying to build an on-line motion planner for robot manipulators. It is intuitively obvious that the movement of a manipulator becomes more flexible as we increase the degree of freedom and thus make the manipulator system more redundant. But on the other hand, known algorithms for motion planning are intractable in terms of the degree of freedom, preventing us from designing a more flexible manipulator. Here we propose a new approach to the motion planning problem. We propose a planning system using a continuous curvature model which takes advantage of redundancy in motion planning without a combinatorial explosion. The continuous curvature model is an idealized manipulator which has an infinite number of rotational joints and is controlled by its curvature. After we get a solution for the continuous curvature model, we try to approximate it by an actual manipulator. By using the continuous curvature model, we can build a motion planning algorithm whose complexity is not related to the number of joints of an actual manipulator but to the complexity of its working environment, while taking advantage of its redundancy. The goal of our approach is to build an efficient and robust motion planner which could be applied to on-line control of a manipulator. In this paper, we introduce two early results of our approach. One is a design of a motion planning system using the continuous curvature model. The other is a simulation for testing some of the building blocks of the motion planning system. On the basis of these encouraging results, we claim that our approach has a significant potential and deserves further research.

AI89-113 **Filling Gaps in Parsed English Sentences,** Coy C. Day, Jr., September, 1989.

Gaps occur in natural language when sentence constituents are omitted as a result of movement rules, subjectless clause constructions, and compound phrases. In order to properly represent the information contained in a sentence, such gaps must be filled before the final, logical representation of the sentence is created. This paper examines the use of purely syntactic methods for eliminating gaps which exist after parsing. Syntactic graphs, rather than parse trees, are used to represent the surface structure of the sentences. It is shown that certain gaps can be filled using only syntactic information. Rules are presented for handling these gaps. There are other gaps, however, which cannot be filled syntactically. Hopefully, knowledge about the world or the context of the sentence can be utilized to fill these gaps.

AI89-112 **Proposal for Research: Representing, Utilizing, and Acquiring Teleological Descriptions**, David W. Franke, September, 1989.

Teleological descriptions capture the purpose of an entity, mechanism, or activity with which they are associated. While a teleological description of a mechanism is distinct from the structural and behavioral descriptions of that mechanism [Kuipers 1985], it is claimed here that a teleological description is constructed with references to elements of structural and behavioral descriptions, just as behavioral descriptions reference elements of the structural description. To further elucidate the distinction between behavior and function, consider the steam-release value in a boiler example in [Kuipers 1985] and the functional (teleological) and behavioral descriptions given there. The function (purpose) of the steam-release value is to prevent an explosion, while the behavior is that the pressure does not exceed a certain value.

The nature of qualitative behavioral descriptions and their derivation has been explored in [de Kleer, Brown 1985], [Forbus 1985]], [Kuipers 1985], and [Williams 1985]. The understanding gained from these investigations into representing and deriving behavior provides the foundation for the next step, namely the investigation of representing and deriving teleological descriptions.

The goal of this work is to study the nature of teleological descriptions and their (domain independent) representation. Applications of teleological knowledge are described, and a technique for deriving teleological descriptions is detailed. A language for teleological descriptions is proposed, and examples demonstrate how teleological descriptions are constructed hierarchically from behavioral descriptions and other teleological descriptions. These examples also demonstrate the independence of this language with respect to the particular behavioral description language chosen. Section 2 describes uses for teleological descriptions, and Section 3 describes the problem of representing teleological descriptions and the representation approach proposed here. Section 4 describes techniques for deriving teleological descriptions. Section 5 details the research plan.

AI89-111 **Qualitative Simulation Extensions for Supporting Economics Models**, David Murry Bridgeland, September 1989.

To be more powerful, financial expert systems need qualitative causal models in addition to associative knowledge. QSIM is a useful tool for building such models, particularly for representing and reasoning with accounting relationships. However, the causal models for financial expert systems also depend upon economics models, and simulating such models requires extensions to QSIM. In particular, QSIM is unable to simulate the combination of average and marginal parameters, or constant time delays between actions and their effects; and it cannot treat certain statics problems dynamically. I identified and implemented ways to overcome these limitations, so a larger class of economics models can now be simulated qualitatively.

AI89-110 **Induction Over the Unexplained: A New Approach to Combining Empirical and Explanation-Based Learning**, Raymond Mooney, Dirk Ourston, Shoiw-yang Wu, August 1989.

This paper presents a new approach to combining empirical and explanation-based learning called *Induction Over the Unexplained* (IOU). Unlike other approaches to integrated learning which use one method to focus the other or provide it with information, IOU allows each method to learn a different part of the overall concept description. It is particularly suited for learning concepts which have both aspects which can be explained in terms of intentionality or causality as well as other "nonexplanatory" or "conventional" aspects which cannot be explained by the existing domain theory. An initial nonincremental feature-based implementation of IOU is presented together with empirical data which demonstrated the advantage of IOU over a pure empirical or analytical system and other integrated learning systems such as IOE.

AI89-109 **Limiting the Use of Learned Rules to Insure the Utility of Explanation-Based Learning,** Raymond Mooney, Siddarth Subramanian, August 1989.

The *utility problem* in explanation-based learning concerns the ability of learned rules or plans to actually improve the performance of a problem solving system. Previous research on this problem has focused on the amount, content, or form of learned information. This paper examines the effect of the *use* of learned information on performance. Experiments and informal analysis show that unconstrained use of learned rules eventually leads to degraded performance. However, constraining the use of learned rules helps avoid the negative effect of learning and lead to overall performance improvement. Search strategy is also shown to have a substantial effect on the contribution of learning to performance by affecting the manner in which learned rules are used. These effects help explain why previous experiments have obtained a variety of different results concerning the impact of explanation-based learning on performance.

AI89-108 **An Integral Theorem Prover and the Role of Proof Planning,** James D. Baker and Shahriar Zand-Biglari, July 1989.

This report results from an experiment in building a theorem prover for the theory of Riemann integration. A PROLOG based prover was developed where the user can input theorems from the terminal or by calling files, can select rule bases, can control the depth of search and recursion cycles, and can select to view a complete proof or only the success path. Approximately 50 theorems have been proved and a set of six cases are discussed. The lack of proof planning is believed to be a major bottleneck in automated theorem proving and the system was modified to accept a user defined plan which resulted in a substantial improvement in capability. As a result, initial steps toward automated proof planning are suggested based on selectively managing the size and content of the search space.

AI89-107 **Qualitative Simulation of Dynamic Chemical Processes,** David Dalle Molle, May 1989 (Ph.D. Dissertation).

Qualitative simulation is a promising technique for analyzing dynamic systems with incomplete knowledge. The QSIM algorithm for qualitative simulation provides a framework for constructing qualitative versions of process models normally represented by ordinary differential equations. Previous work with the QSIM algorithm modeled systems with relatively simple dynamics. In this work, qualitative models were developed for a variety of dynamic chemical processes including first-, second-, and third-order linear processes, nonlinear processes such as chemical reactors, multivariable systems, and a process under PI control. In addition, extensions to the QSIM algorithm were developed for reasoning about the higher-order derivatives that help lead to tractable simulations of many of these systems.

Much of this work focused on identifying the necessary structure of the qualitative model that would minimize the number of spurious predictions. For some systems, the qualitative model required redundant representations of the process equations in order to capture quantitative knowledge lost in formulating the qualitative representation. Furthermore, many models needed additional reasoning techniques to cope with ambiguity inherent in qualitative mathematics. The most effective extensions placed additional constraints on higher-order derivatives to eliminate large classes of spurious behaviors. For every system studied, the qualitative model predicted all of the behaviors expected from a traditional quantitative analysis but without precise knowledge of the process parameters.

When quantitative information was included in a qualitative model, the number of predictions generated was reduced and the resulting behaviors were more descriptive.

AI89-106 **Concept Learning and the Problem of Small Disjuncts**, Robert Holte, Liane Acker, and Bruce Porter, May 1989.

Ideally, definitions induced from examples should consist of all, and only, disjuncts that are meaningful (e.g. as measured by a statistical significance test) and have a low error rate. Existing inductive systems create definitions that are ideal with regard to large disjuncts, but far from ideal with regard to small disjuncts, where a small (large) disjunct is one that correctly classifies few (many) training examples. The problem with small disjuncts is that many of them have high rates of misclassification, and it is difficult to eliminate the error-prone small disjuncts from a definition without adversely affecting other disjuncts in the definition. Various approaches to this problem are evaluated, including the novel approach of using a bias different than the "maximum generality" bias. This approach, and some others, prove partly successful, but the problem of small disjuncts remains open.

AI89-105 **Grammar Acquisition and Parsing of Semi-Structured Text**, Margaret Reed-Lade, May 1989 (Master's Thesis).

This thesis describes software that acquires a grammar for semi-structured text and parses the text into a database. Semi-structured text can be generally characterized as text which is laid out on a page in a structured but not necessarily rigid fashion. The thesis describes related work, presents a user session, provides details of the grammar acquisition and parsing processes, and suggests directions for future work. The software described in the thesis provides a means by which ASCII files of semi-structured text could be automatically entered into a dynamically defined database structure.

AI89-104 **Parsing with Constrained Marker Passing**, Yeong-Ho Yu and Robert F. Simmons, May 1989 .

Marker Passing is an inference mechanism used with *semantic network* representations in NLP to find all paths between any pair of network nodes. Theoretically, each node in the network can be assumed to be a simple processor that activates each node/processor to which it connects, and the procedure can be seen to be accomplished as a parallel search that succeeds each time the initial pair of nodes are connected by some path of activations. Unfortunately, most such paths are irrelevant and meaningless, so a filtering process is required to select those paths that make relevant and sensible inferential connections. Although previous research in the area has appreciated the potential value of parallel marker passing, the filtering has usually been a serial process, and the resulting system if implemented in a parallel machine would find filtering a bottleneck that would sharply limit the parallel processing gains. This paper presents a mechanism for accomplishing the filtering as an integral aspect of the parallel marker passing thus removing the expensive bottleneck of serial filtering. Furthermore,this paper demonstrates the usefulness of this mechanism by applying it in parsing English sentences.

AI89-103 **A Model-Based Representational Framework for Expert Physics Problem Solving**, Hyung Joon Kook, May 1989 (Ph.D. Dissertation).

Real-world physics problems are presented in terms of informal, real-world objects and the relationships among them. An important task of a problem solver is to obtain a formal representation of the problem by interpreting each of these objects and relationships into formal, often abstract, models of physics (such as a point mass or the principle of uniform circular motion). Representation of such models has been the main research issue in building APEX (*A Physics Expert*), a computer program developed for solving elementary physics problems. The main areas of research APEX addresses are ways to represent the conceptual models of the domain, the development of methods to obtain the representation of a problem in terms of these models, and the study of the linkage between the two representations (initial, informal, and internal, formal, problem representations). During APEX's problem solving, the problem is represented in the form of a data connection network, which is progressively augmented by these models in the form of additional network elements. In order to explicate the data conversion scheme between the model instances and the features of objects in the initial problem, the notion of *view* is presented as an object-level representational framework for connecting the two representations. The view framework also supports multiple representations (i.e., viewing many objects as a single canonical physical object, and one object as many canonical physical objects), handling of incompletely specified problems, and the *invertibility* of the views (the facility for transferring the results obtained from the internal representations back to the original representations). APEX facilitates the selection of models in tw independent modes: a human-guided mode in which models are selected by a human through a user-friendly interface, and a machine-inference mode in which models are selected by a rule-based inference system. This computational framework provides a powerful representational mechanism that allows a *finite* set of physical principles to be applied to a potentially *infinite* variety of problems.

AI89-102 **Understanding Coreference in a System for Solving Physics Word Problems**, William Charles Bulko, May 1989 (Ph.D. Dissertation).

In this thesis, a computer program (BEATRIX) is presented which takes as input an English statement of a physics problem and a figure associated with it, understands the two kinds of input in combination, and produces a data structure containing a model of the physical objects described and the relationships between them. BEATRIX provides a mouse-based graphic interface with which the user sketches a picture and enters English sentences; meanwhile, BEATRIX creates a neutral internal representation of the picture similar to that which might be produced as the output of a vision system. It then parses the text and the picture representation, resolves the references between objects common to the two data sources, and produces a unified model of the problem world. The correctness and completeness of this model has been validated by applying it as input to a physics problem-solving program currently under development.

Two descriptions of a world are said to be *coreferent* when they contain references to overlapping sets of objects. Resolving coreferences to produce a correct world model is a common task in scientific and industrial problem-solving: because English is typically not a good language for expressing spatial relationships, people in these fields frequently use diagrams to supplement textual descriptions. Elementary physics problems from college-level textbooks provide a useful and convenient domain for exploring the mechanisms of coreference.

Because flexible, opportunistic control is necessary in order to recognize coreference and to act upon it, the understanding module of BEATRIX uses a blackboard control structure. The blackboard knowledge sources serve to identify physical objects in the picture, parse the English text, and resolve coreferences between the two. We believe that BEATRIX demonstrates a control structure and collection of knowledge that successfully implements understanding of text and picture by computer. We also believe that this organization can be applied successfully to similar understanding tasks in domains other than physics problem-solving, where data such as the output from vision systems and speech understanders can be used in place of text and pictures.

AI89-101 **Representation and Reuse of Explanations in Case-Based Reasoning**, L. Karl Branting, March 1989 (Dissertation Proposal).

Many forms of expertise depend heavily on the representation and reuse of explanations of past cases This paper proposes a computational model of explanation representation and reuse in domains having a complex but incomplete theory. The model is based upon a form of explanation, termed exemplar-based explanation (EBE), in which conclusions about an exemplar are imputed to a new case on the basis of relevant similarities between the exemplar and the new case. Relevant similarities can be explained by direct featural correspondence, rule-based explanations, or subordinate EBE's. The performance of an implementation of the EBE model, GREBE, is illustrated with examples from the domain of worker's compensation law.

AI89-100 **Toward an Integrated Expert Database System**, Daniel P. Miranker and David A. Brant, March 1989.

Incremental match algorithms such as RETE and TREAT have an exponential worst-case space complexity. Thus, it is impractical, using currently known methods, for an expert-system written in production system form to directly absorb the quantity of data present in even small databases. Therefore, expert-system applications with a database component internally maintain a distinct copy of a small subset of the database and poll the database when new information is needed. We describe a "lazy" match algorithm that has linear worst case space complexity in the size of the database and will prune much of the search for instantiated rules. We consider the lazy match to be the prerequisite result for developing integrated rule-based expert database systems.

AI89-99 **Performance Models for Fine-Grained AI Machines**, Daniel P. Miranker and Archie D. Andrews, March 1989, revised and updated July 1990.

This paper describes a model that supports parallel inferencing on fine-grained networks of cooperating processors and then analyses the expected performance resulting from implementing those networks onto currently realizable multiprocessor architectures. A serial simulation engine of the marker passing paradigm for semantic networks is built. The model is then extended by introducing indexing techniques and parallelism. The parallel version, when taken to the extreme, closely resembles the NETL machine, a massively parallel processor that directly supports a realization of semantic networks and the marker passing paradigm. Although the discussion focuses on the NETL paradigm the model and the interpretation of the results are applicable to the neural network models and extendible to the general network knowledge representation systems such as frame and object-oriented systems.

Analysis of the variant models reveals: 1) the performance of sequential special purpose hardware with indexing performs favorably when compared to the predicted performance of the NETL machine; 2) as increasing amounts of parallelism are introduced into fine-grained parallel semantic net systems, contention alone does not limit usable parallelism; but, 3) the performance of such fine-grained parallel systems in a synchronous, unbuffered architectural model are very sensitive to the transit time through each stage of the interconnection network.

AI89-98 **Automatic Evaluation of Third-Order Constraints in Qualitative Simulation**, David T. Dalle Molle, February 1989.

A common source of spurious behaviors in qualitative simulation is ambiguity in the derivative of complex expressions or variables such as the highest-order derivative in a model. An ambiguous derivative may "chatter" endlessly between increasing, decreasing, and steady (but is still constrained by continuity). Curvature (second-order) constraints provide a means for possibly eliminating otherwise intractable branching that occurs when the derivative of an expression is under constrained. These constraints can often be derived automatically from the constraints of a model in terms of variable value and derivative information already present in the model. Previous work with curvature constraints on high-order systems demonstrated that in many cases, the value of a curvature constraint expression can be zero, leaving the actual direction of change determined by the next-highest nonzero derivative. While third-order constraints can be derived manually, finding an expression that can be evaluated from information already present in the model is nontrivial. In this work, a method is presented for automatically evaluating third-order constraint expressions in some cases from variable value, derivative, and curvature information already present in the model without actually deriving a new expression. Several examples are provided that show the need for third-order constraints and demonstrate that the proposed method can eliminate otherwise intractable branching without any additional input from the modeler.

• an order form is located at the end of the listings •

AI89-97 **Gaining Autonomy During Knowledge Acquisition**, E. Ray Bareiss, Bruce W. Porter, Kenneth S. Murray, February 1989.

A knowledge-acquisition tool must gain autonomy throughout the development of a knowledge base A tool that fails to exploit the evolving knowledge base, and thus improve independent problem solving and learning, is limited to a narrow phase of development. We review the current methods for knowledge acquisition and characterize the developmental phase that each supports. We describe our learning system, Protos, which supports multiple phases of development by shifting the training it expects and the assistance it provides as the knowledge base grows. Finally, we describe our continuing machine learning research into tools for the advanced stages of development, that use extensive domain knowledge to integrate new information.

AI89-96 **Concept Learning and Heuristic Classification in Weak-Theory Domains**, Bruce W. Porter, E. Ray Bareiss and Robert C. Holte, February 1989, revised and updated March 1990.

This paper describes a successful approach to knowledge acquisition for heuristic classification. Almost all current programs for this task create or use explicit, abstract generalizations. These programs are largely ineffective for domains with weak or intractable theories. An exemplar-based approach is suitable for domains with inadequate theories, but raises two additional problems: determining similarity and indexing exemplars. Our approach extends the exemplar-based approach with solutions to these problems. An implementation of our approach, called Protos, has been applied to the domain of clinical audiology. After reasonable training, Protos achieved a competence level equaling that of human experts and far surpassing that of similar programs. Additionally, an "ablation study" has identified the aspects of Protos that are primarily responsible for its success.

AI89-95 **Spatial Unification: Qualitative Spatial Reasoning about Steady State Mechanisms**, David R. Throop, February 1989.

Qualitative simulation is one method for reasoning with incomplete knowledge about mechanisms which change in time. This work adapts that method to reasoning about steady state mechanisms—which change continuously in *space*.

Because flow-paths in mechanisms can loop and self-intersect, spatial behaviors show complexities that temporal behaviors do not. This paper features two extended examples of counter-current heat exchange. Both examples involve such loops—places where a later flow-path segment interacts with an earlier segment.

QSIM has been extended to simulate the behavior of these mechanisms. The extension rules out inconsistent predictions by first assuming them, then following them to a contradiction. The contradictions are detected as inconsistencies between two separate views of the behavior—one view from each side of the interaction.

The two views are cast as the restrictions in a two point boundary value problem. The solutions are generated by instantiating one of the boundaries with all possible qualitative values. The inconsistent predictions manifest as instantiations that have no qualitative solutions at the second boundary.

The extension has several novel aspects which support spatial reasoning, including:

- Spatially unifying multiple views of a QSIM state;
- Allowing a state in one region to transition to multiple states (representing multiple possible behaviors) in an adjoining region; and

- Allowing two parameters to share a single quantity space.

These extensions are expected to have direct application back to temporal Qualitative Simulation.

AI89-94 **On the Applicability of the Smoothness Assumption to Qualitative Simulation,** David R. Throop, February 1989.

The *smoothness assumption* is a useful tool for filtering spurious behaviors from qualitative simulations. However, some common monotonic functions violate it. Violations, which are not always obvious, cause real behaviors to be ignored. This leads to qualitatively incorrect and possibly dangerous conclusions.

This report describes three simple systems which violate the smoothness assumption: two cascaded tanks, a single tank, and a U-Tube. The report discusses errors which could follow this violation: prediction of future behavior becomes faulty, and correct explanations of observed behavior are rejected.

AI88-93 **Representation of Models for Solving Real-World Physics Problems,** Hyung Joon Kook and Gordon S. Novak, December 1988.

Expertise in solving real-world physics problems is characterized by the ability to adequately *set up* (or *represent*) a problem. An expert's representation is distinguished from a novice's by employing idealized, formal models (such as a point mass or the principle of uniform circular motion) for modeling objects and relationships in the initial problem. A computer program, APEX (*A Physics Expert*), has been developed to investigate the nature and contents of such models. As the representational primitives of the domain, two types of models are defined: *canonical physical objects* and *physical models*. During problem solving, the problem is represented in the form of a data connection network, which is progressively augmented by these models in the form of additional network elements. APEX's representational framework supports multiple representations (e.g., viewing many objects as a single canonical physical object) and handling of incompletely specified problems. The proposed representational methodology may be applied to similar domains in which representations of domain knowledge are based mainly on models.

AI88-92 **Bottom-up, Text-driven Construction of Discourse Structures for Understanding Descriptive Texts,** Jungyun Seo, December 1988.

This work proposes a bottom-up processing model of understanding descriptive texts—especially disease descriptions. Early research on *top-down, expectation-driven* text understanding using pre-built discourse structures such as scripts, schemas, or plan-trees demonstrated psychologically plausible ways of explaining narrative texts. These approaches, however, have not been developed for understanding descriptive texts. Since descriptive texts usually explain topics to a reader who is not familiar with them, it is unlikely that the reader will have a pre-determined content expectation when he reads such texts. We propose a *bottom-up, text-driven* understanding system for descriptive texts. The proposed research suggests three focusing strategies to handle the huge search space in typical discourse analysis. First, the proposed system focuses on the relevant part of the available knowledge base for a text, as the system classifies the concept of each word and determines the semantic relations among the words in a sentence. Second, only focal words in each sentence, called the *local focus*, are considered when the system finds the relation between two sentences. Finally, the system considers only the focal sentence in the context, called the *global focus*, when it relates the current sentence to the context. The system maintains a *focus history stack* to keep prospective global foci in a recency order.

• an order form is located at the end of the listings •

AI88-91 **Text Outlining System**, Hae-Chang Rim, November 1988.

This paper proposes a text outlining system which can automatically translate sequences of input sentences into an outline structure of the text. The proposed system accomplishes the text outlining task by performing three subtasks: information classification, focus selection and information organization The system classifies the information, chooses the focus for each sentence, and then constructs a hierarchical outline structure by relating each sentence to the context through the classification labels or through the relationships between foci. Each subtask is explained with a sample text "Hepatitis". Possible extensions to a different domain are also discussed.

AI88-90 **KI: An Experiment in Automating Knowledge Integration**, Kenneth S. Murray, October 1988.

Knowledge integration is the incorporation of new information into existing knowledge. It is an active learning process involving three steps. recognition, elaboration, and adaptation. Recognition identifies known concepts that relate to the new information. Elaboration embellishes new information to fill in missing details and identify conflicts with existing knowledge. Adaptation modifies the learner's knowledge to accommodate the elaborated information. Although it has received little attention in machine learning, knowledge integration is central to psychological models of learning. To study knowledge integration, we propose to develop and evaluate a knowledge acquisition tool that helps with the arduous task of extending a large-scale knowledge base. An initial prototype of this tool has been implemented and demonstrated on a complex extension to a large knowledge base.

AI88-89 **Varying the Degree of Generalization in Concept Learning: An Empirical Study**, Liane Acker, November 1988.

In the CN2 and ID3 automatic rule induction systems, classification rules that *cover* (explain or match) few training instances have higher error rates than rules that cover more training instances.

Both ID3 and CN2 are *aggressive generalizers:* all rules formed are maximally generalized. Perhaps the error rate of these systems could be reduced by reducing the error rates of rules with small coverage. One way to do this might be to vary the degree of generalization so that the generality of a rule produced is proportional to the number of training examples covered by the rule.

In this study varying the degree of generality of rules did not result in a significantly lower overall error rate, even though the error rates of rules with low coverage were reduced. The reason is that many of the test cases that would be correctly classified by a maximally general rule are not covered by a more specific rule. They must be classified either by another of the induced rules or by a default rule. In this study the majority of these test cases matched only the default rule, which had a very high error rate (around 50%.) As a result, the system's overall error rate was usually worse when the degree of generality was varied than when maximally general rules were formed.

AI88-88 **AI Research in the Context of a Multifunctional Knowledge Base: The Botany Knowledge Base Project**, Bruce W. Porter et. al., October 1988.

Our research applies a large-scale, multifunctional knowledge base to significant AI problems. The knowledge base encodes commonsense and expert knowledge of botany, in particular, anatomy, physiology and development. To convey the scope of the knowledge base, we present the top-level representations of objects and processes as well as the skeletal diagramming of the knowledge base. The first applications of the Botany Knowledge Base are machine learning and tutoring. The goal of our learning research is to develop and evaluate a computational model for a new learning task—knowledge integration. This task, which might be termed "learning at the fringes of a knowledge base," involves the incorporation of new information into existing knowledge. The goal of our intelligent tutoring research is to develop knowledge representations and processes for dynamically revising teaching plans and tailoring explanations to the individual needs of a student. Our research is predicated on the knowledge principle that emphasizes the crucial role of extensive, task-independent knowledge in intelligence.

AI88-87 **Guide to CL-Protos: An Exemplar-based Learning Apprentice**, Daniel L. Dvorak, September 1988.

CL-Protos is a machine learning program which acquires knowledge for performing expert heuristic classification. The program learns as a byproduct of performing classification under the guidance of a human teacher. When presented with a case to be classified, CL-Protos classifies the case by recalling an appropriate previous case and explaining its similarity to the new case. If classification fails, the teacher is asked to supply an explanation of the correct classification. The program learns by selectively retaining cases and their explanations. By learning from explanations in addition to examples, CL-Protos is able to learn to classify from little training and to justify its conclusions. CL-Protos is a Common Lisp reconstruction of the Protos "exemplar-based learning apprentice". Protos was originally conceived by E. Ray Bareiss and Bruce W. Porter, implemented in Prolog, and applied to the domain of clinical audiology.

This document is in two parts—a user's guide and a software guide. The first part overviews the Protos paradigm and explains how to use the CL-Protos program. The second part is a guide to the software; it describes the high-level design of CL-Protos in terms of the major structures and functions.

AI88-86 **Replaying Episodes of an Automated Metaphoric Interface Designer**, Brad Blumenthal, August 1988.

This paper is an edited version of a dissertation proposal which outlines a computer system to partially automate the design of metaphoric human interfaces for computer applications. This project is motivated by a desire to improve the derivational analogy technique proposed by Carbonell [Carbonell 86], and by a desire to help computer interface designers more easily create effective interfaces. The system consists of two components. The first is a design component which uses an agenda-based control mechanism combined with a large number of human interface design heuristics to produce a detailed, metaphoric interface specification from a generic application description. The second component uses a derivational analogy based design replay algorithm which takes the history produced by the first component and uses it to guide subsequent design episodes using different metaphors. This paper also describes a mechanism for fleshing out the details of a new metaphor given experience with a similar metaphor and knowledge of the similarities between the two real world sources of the metaphors. A working prototype implementation is described which elaborates some of the details of this proposed system. Finally, techniques for validating the work proposed are described.

AI88-85 **Roles, Events and Saying events in Expository Discourse**, Olivier J. Winghart, August 1988.

In this paper, we describe a computational representation of expository discourse, separating the elements of the described situation (Roles and Events) from those of the exposition itself (Saying events). The situation elements do not depend on the writer's intentions; their interrelations are only due to world concept coherence. In particular, we propose a new notion of rhetorical relation as mapping of expansions between situation elements, rather than one single relation between linguistic units. This notion is clearer, and easier to recognize than the typical notion. On the other hand, it is less informative and doesn't induce a structuring on the text.

• an order form is located at the end of the listings •

AI88-84 **A Parallel Implementation of Logic Programs**, Yow-Jian Lin, August 1988. (Ph.D. Dissertation)

Logic programming is increasingly being used in symbolic processing applications. As the complexity of symbolic computation increases, executing logic programs in parallel is perhaps the only way to meet the computational demands of the next generation computing. Due to implicit representation of parallelism and separation of specification and control, logic programming also offers a possible solution to the problem of utilizing highly parallel architectures. Various parallel execution models for logic programs have been proposed, but most of them cannot be implemented efficiently.

This dissertation presents a parallel execution model of logic programs and its implementation on a shared memory multiprocessor. The execution model preserves the don't-know nondeterminism, and follows the generator-consumer approach to exploit AND-parallelism. Unlike many other execution models, this model constructs the data-dependency information dynamically, requires no information from the user, and is able to exploit all the AND-parallelism available in the framework of the generator-consumer approach. Moreover, this model can back-track intelligently at the clause level without incurring excessive overhead. Our implementation of this execution model on the Sequent Balance multiprocessor obtains linear speedup on many programs containing AND-parallelism. Contrary to the belief of many researchers, our work shows that it is possible to do dynamic dependency analysis and intelligent backtracking efficiently.

AI88-83 **Protos: A Unified Approach to Concept Representation, Classification, and Learning**, Ellis Raymond Bareiss Jr., August 1988. (Ph.D. Dissertation)

The primary contribution of this research is a unified approach to concept representation, classification, and concept learning. This approach has been implemented as a computer program, Protos, which learns concepts as it performs classification under the guidance of a teacher. The soundness of the approach has been demonstrated by successfully applying Protos to the task of acquiring knowledge for performing heuristic classification at an expert level of proficiency.

The Protos approach addresses the complexities of representing, using, and learning natural concepts. These concepts are polymorphic and ill-defined. Most machine learning research is based on inductive learning and deductive classification, which are more suitable for artificial domains (*e.g.*, mathematics) than natural domains (*e.g.*, medicine). In contrast, Protos takes an exemplar-based approach. It represents concepts extensionally as sets of retained exemplars, classifies a new instance by recalling a similar exemplar and explaining its similarity to the instance, and learns when a classification failure indicates that knowledge is missing. Because Protos learns as a byproduct of classification, its performance continually improves.

Protos has been experimentally evaluated by training it to diagnose hearing disorders. An expert audiologist trained Protos with 200 cases of hearing disorder. Through this small amount of training, Protos evolved into an expert system whose classification performance was comparable to that of experienced human clinicians.

AI88-82 **Concept Learning with Incomplete Data Sets**, Rita Terese Duran, August 1988. (Master's Thesis)

This report presents a case study of the effects of missing data on the performance of ID3, a representative inductive learning system. Although several mechanisms have been proposed to enable ID3 to cope with missing data, these mechanisms perform poorly when tested on data from the domain of audiology, a typical diagnostic domain. To test the conjecture that ID3's poor performance is due to the high percentage of missing features in the audiology data, these mechanisms were tested on data from another domain that had been artificially corrupted with various percentages of missing features. The results of this experiment show that ID3 is indeed sensitive to missing data. These findings are significant because missing data is a fundamental problem in diagnostic domains which must be addressed by inductive learning systems.

AI88-81 **Understanding Complex Dynamic Scenes: Part I—Tracking Feature Points in Time-Varying Images using an Opportunistic Selection Approach**, Vincent S. S. Hwang, August 1988.

Understanding complex dynamic scene is an exciting and difficult task since many objects may be present in the scene. These objects may each be moving. Occlusions can also arise. It is necessary to identify and group those features that are associated with the same object in order to compute the 3-D structure and to recover the motion parameters of each object. One way to group "related" features together is to compute the trajectory of each feature and reason about the trajectories in order to obtain proper "grouping." In this report, we present our approach toward the computing of trajectories of feature points based on the tracking of feature points across an extensive number of frames.

In tracking of feature points in time-varying images, one major issue is to identify the feature points at different times that represent the same physical object point. This process is often called the *correspondence problem* and is an important research issue in both motion analysis and stereopsis. This paper describes a new approach toward the tracking of feature points in time-varying images by reasoning about the trajectories of feature points. The advantage of this approach is that the solution can be computed efficiently and the algorithm can potentially be supported by multiple processor computers. This approach is tested extensively on the synthesized data. The statistics on the experiments strongly suggests the generality and the robustness of the approach.

AI88-80 **Superlinear Speedup in Parallel State-Space Search**, V. Nageshwara Rao and Vipin Kumar, August 1988.

When N processors perform depth-first search on disjoint parts of a state space tree to find a solution, then the speedup can be superlinear (i.e., >N) or sublinear (i.e., <N) depending upon when a solution is first encountered in the space by one of the processors. It may appear that on the average the speedup would be either linear or sublinear. Using an analytical model, we show that if the search space has more than one solution and if these solutions are randomly distributed in a relatively small region of the search space, then the average speedup in parallel depth-first search can be superlinear. If all the solutions (one or more) are uniformly distributed over the whole search space, then the average speedup is linear. This model is validated by our experiments on synthetic state-space trees and the 15-puzzle problem. The same model predicts average superlinear speedup in parallel best-first branch-and-bound algorithms on suitable problems.

AI88-79 **Exploratory Robot Learning of Sensory-Motor Integration**, W. Scott Spangler, May 1988. (Master's thesis)

This thesis described an algorithm for mobile robot learning of sensory-motor coordination. The goal is to learn how to characterize the effect that taking a particular movement action will have on a robot's sense input, given the initial sensory situation. The concepts are learned from the sensory results of taking random actions from within a training environment (Learning from Examples), but these concepts are general enough to apply to unseen environments. The learning algorithm is implemented and tested for a particular type of simulated sonar sensing robot.

AI88-78 **The TOUR Model: A Theoretical Definition**, Benjamin J. Kuipers, April 1988.

The TOUR model is a representation for knowledge of large-scale space as exhibited by the human "cognitive map." The representation is capable of expressing a variety of types of incomplete knowledge of spatial properties of an environment, in the form of procedural descriptions, topological descriptions, and metrical descriptions of the environment. The inference mechanism is oriented toward the problem of assimilating observations gathered during exploration and travel into a cognitive map capable of supporting spatial problem-solving using well-known methods. In this paper, the elements of the TOUR model representation are stated in a more formal, predicate-calculus-like form than in previous publications.

AI88-77 **The Role of Explanation in Reasoning from Legal Precedents**, L. Karl Branting, April, 1988.

A computational model for the process of reasoning from legal precedents to new cases is outlined in which legal concepts consist of facts of precedent cases together with explanations for the satisfaction of the concept by each case. Classification of new cases is accomplished by coercing the explanation structure of a precedent classification onto the new case. Techniques for representing and mapping explanation structures are presented.

AI88-76 **Constrained Marking Passing**, Yeong-Ho Yu and Robert F. Simmons, April 1988.

Marker passing is an inference mechanism used in semantic network representations to find all paths between any pair of network nodes. Theoretically, each node in the network can be assumed to be a simple processor that activates each node/processor to which it connects, and the procedure can be seen to be accomplished as a parallel search that succeeds each time the initial pair of nodes are connected by some path of activations. Unfortunately, most such paths are irrelevant and meaningless, so a filtering process is required to select those paths that make relevant and potential value of parallel marker-passing, the filtering has usually been a serial process, and the resulting system if implemented in a parallel machine would find filtering a bottleneck that would sharply limit the parallel processing gains. This paper presents a method for accomplishing the filtering as an integral aspect of the parallel marker passing thus removing the expensive bottleneck of filtering.

AI88-75 **A Computerized Prototype Natural Language Tour Guide**, Hwee Tou Ng, April, 1988.

This paper describes a computerized prototype natural language tour guide. The computer system serves as a tour guide and provides the user with travel information through an interactive question answering session. Based on the travel information stored in its database, the system processes the user's query, accesses the database and returns an answer. The system is capable of providing cooperative response when appropriate, and it can also handle simple sentence fragments (ellipsis).

AI88-74 **AND-parallel Execution of Logic Programs on a Shared Memory Multiprocessor**, Vipin Kumar, April 1988.

This paper presents the implementation of an AND-parallel execution model of logic programs on a shared -memory multiprocessor. The major features of the implementation are (i) dependency analysis between literals of a clause is done dynamically without incurring excessive run-time overhead. (ii) backtracking is done intelligently at the clause level without incurring any extra cost for the determination of the backtrack literal, iii) the implementation is based upon the Warren Abstract Machine (WAM), hence retains most of the efficiency of the WAM for sequential segments of logic programs. Performance results on Sequent Balance 21000 show that our parallel implementation can achieve reasonable speedup on dozens of processors.

AI88-73 **A Robust Qualitative Method for Robot Exploration and Map-Learning**, Benjamin J. Kuipers and Yung-Tai Byun, April 1988.

We present a robust qualitative method for a mobile robot to explore an unknown environment and learn a map. Procedural knowledge for the movement, topological model for the structure of the environment, and metrical information for geometrical accuracy are separately represented in our method, whereas traditional methods describe the environment mainly by metrical information. The topological model consists of nodes and arcs corresponding to distinctive places and local travel edges linking nearby distinctive places. A distinctive place is defined as the local maximum of some measure of distinctiveness appropriate to its immediate neighborhood, and is found by a hill-climbing search Local travel edges are defined in terms of local control strategies required for travel. How to find distinctive places and follow edges is the procedural knowledge which the robot learns dynamically during exploration stage and guides the robot in the navigation stage. An accurate topological model is created by linking places and edges, and allows metrical information to be accumulated with reduced vulnerability to metrical errors. The method, inspired by the study of cognitive maps which humans use, can be robust in the face of various possible errors in the real world, and the map description can be more useful than traditional approaches to the robot, as well as people. We describe a working simulation in which a robot, NX, with range sensors and two tractor-type chains explores a variety of static 2-D environments and we give its successful results under varying levels of random sensor error.

AI88-72 **An Analytical Framework for Learning Systems,** Robert Craig Holte, February 1988.

The problem addresses in this thesis is that of defining a set of concepts and techniques that facilitate the comparison and analysis of learning systems. Systems are modelled in terms of certain abstract processes and bodies of information. Different types of systems correspond to different ways of representing the model. Systems of different types are compared using behavior-preserving transformations. Formal definitions are given for "representation" and "generative structure" of a system. These and related concepts, such as "bias" and "implicit knowledge", facilitate the analysis of a system's efficiency and its use of task-specific knowledge.

AI88-71 **A High Level Language Approach to the Fault Tolerant Execution of AI Expert Systems,** Daniel P. Miranker, March 1988.

To perform load balancing certain algorithms for the parallel execution of expert systems written in production rule form require the replication of both the program and the data. Thus an intrinsic requirement of fault tolerant systems, the replication of hardware and software components, is automatically satisfied by the parallel implementation of these systems. This paper describes how these algorithms may be extended to create a fault tolerant expert system interpreter.

The fault-tolerant model includes graceful degradation. Certain properties of forward chaining rule systems allow redundant processors to be exploited simultaneously as spares and as additional processors contributing to performance. Further, increased processor utilization may permit software based fault-tolerance into a parallel production system interpreter may be free.

AI88-70 **Parallel Heuristic Search of State-Space Graphs: A Summary of Results,** Vipin Kumar, K. Ramesh and V. Nageshwara Rao, March 1988.

This paper presents many different parallel formulation of the A*/Branch-and-Bound search algorithm. The parallel formulations primarily differ in the data structures used. Some formulations are suited only for shared-memory architectures, whereas others are suited for distributed-memory architectures as well. These parallel formulations have been implemented to solve the vertex cover problem and the TSP problem on the BBN Butterfly parallel processor. Using appropriate data structures, we are able to obtain fairly linear speedups for as many as 100 processors. We also discovered problem characteristics that make certain formulations more (or less) suitable for some search problems. Since the best-first search paradigm of A*/Branch-and-Bound is very commonly used, we expect these parallel formulations to be effective for a variety of problems. Concurrent and distributed priority queues used in these parallel formulations can be used in many parallel algorithms other than parallel A*/branch and bound.

• an order form is located at the end of the listings •

AI88-69 **Understanding Text with an Accompanying Diagram,** William C. Bulko, March 1988.

Because English is typically not a good language for expressing spatial relationships, people frequently use diagrams to supplement textual descriptions in engineering and scientific applications. BEATRIX is a computer program which takes as input an English statement of a physics problem and a figure associated with it, understands the physical objects described and the relationships between them, and produces a data structure containing a model of the problem. Using a simple graphic interface provided by BEATRIX, the user sketches a picture using a mouse-based menu system similar to other familiar drawing programs. The user may also enter normal English sentences through the keyboard, using the mouse to help in editing. Meanwhile, BEATRIX transparently creates an internal representation of the picture similar to that which might be produced as the output of a vision system. It then translates the text and the picture representation into a semantic network model, resolves the references between objects common to the two sources of knowledge, and produces a unified model of the problem world. The correctness and completeness of this model can be validated when it is supplied as input to a physics problem-solving program which is currently under development.

A major problem in the field of knowledge representation is that of *coreference*: recognizing when two descriptions contain references to the same object or set of objects so that information is associated with the correct object in the internal model. Flexible, opportunistic control is necessary in order to recognize coreference and to act upon it; accordingly, the understanding module of BEATRIX uses a blackboard control structure. The knowledge sources for the blackboard contain information for identifying and matching common objects in physics problems. The blackboard also handles the semantics of parsing the English text and the identification and association of picture elements.
It is natural and often necessary for people to communicate using a combination of text and pictures. There are many applications in science and engineering where the ability to communicate in this way with a computer could be used to a great advantage. We believe that BEATRIX illustrates a control structure and collection of knowledge that can successfully implement text and picture understanding by computer. We also believe that this organization can be applied successfully to similar understanding tasks in domains other than physics problem-solving, where sources of knowledge such as the output from vision systems and speech understanders can be used in place of text and pictures.

AI88-68 **Parallel Depth First Search on the Ring Architecture,** Vipin Kumar, V. Nageshwara Rao and K. Ramesh, January 1988.

This paper presents the implementation and analysis of parallel depth-first search on the ring architecture. At the heart of the parallel formulation of depth-first search is a dynamic work distribution scheme that divides the work between different processors. The effectiveness of the parallel formulation is strongly influenced by the choice of the work distribution scheme. In particular, a commonly used work distribution scheme is found to give very poor performance on large rings(> 32 processors). We present a new work distribution scheme that is better than the work distribution scheme used by other researchers, and gives good performance even on large rings (128 processors). We introduce the concept of iso-efficiency function to characterize the effectiveness of different work distribution schemes.

AI88-67 **Omnirec: A Character Recognition System,** Timothy J. Buchowski and Vincent S. S. Hwang, February 1988.

Most of the commercially available machine printed character recognition systems require input text printed in a single font and style. Omnirec is a character recognition system designed to recognize the ten numerals 0-9 printed in a variety of fonts, styles and sizes. Its three principal components are a preprocessor, a feature detector, and a classifier.

Classification is performed with a decision tree. A preliminary decision tree was constructed from a training set of 160 images of numerals printed in four fonts. Experiments were performed with a database of 1560 numerals printed in ten fonts. The preliminary decision tree achieves greater than 97 percent correct recognition. with little effort, a final decision tree was generated from the preliminary decision tree. the final decision tree achieves 100 percent correct recognition with the 1560-image database.

AI88-66 **A General Explanation-Based Learning Mechanism and Its Application to Narrative Understanding,** Raymond Mooney, January 1988. (PhD dissertation)

This report consists of the author's dissertation from the University of Illinois at Urbana-Champaign.

AI88-65 **Higher Order Derivative Constraints and a QSIM-based Total Simulation Scheme,** Charles Chiu, January 1988.

In a recent paper in collaboration with Kuipers, we reported two distinct methods in combatting the branching problems occurred in qualitative simulations. The first method changes the level of the behavioral description to aggregate an exponentially exploding behaviors into a few distinct possibilities. The second method identifies the approximation in the simulation algorithm, replacing the direction of change of a parameter by its slope, to be a source of the branching problems. With additional assumptions about the smoothness of partially known monotonic functions, we showed that "curvature constraints" can be derived without introducing new parameters in the models. These curvature constraints, in many instances, have corrected the deficiency of this approximation and have substantially subdued the branching problems. In this work we expand on our second approach. First we present the details of our curvature constraint algorithm and its implementations. Second we extend our applications of higher order derivative constraints to include models which involve third order derivative constraints. Third using the two tank model as an example, we present a total simulation scheme, which among other things, includes the construction of QSIM "behavior-families" and an "asymptotic analysis". It leads to the prediction of the qualitative global behavior pattern of the system.

AI87-64 **Syntactic Graphs: A Representation for the Union of All Ambiguous Parse Trees,** Jungyun Seo and Robert F. Simmons, November 1987. (Revised, October 1988)

In this paper we present a new method of representing the surface syntactic structure of a sentence. Trees have usually been used in linguistics and natural-language processing to represent syntactic structures of a sentence . A tree structure shows only one possible syntactic parse of a sentence, but, in order to choose a correct parse, we need t examine all possible tree structures one by one. Syntactic-graph representation makes it possible to represent all possible surface syntactic relations in one directed graph. Since a syntactic graph is expressed in terms of a set of triples, higher-level semantic processes can access any part of the graph directly without navigating the whole structure. Furthermore, since a syntactic graph represents the union of all possible syntactic readings of a sentence, it is fairly easy to focus on the syntactically ambiguous points. In this paper, we introduce the basic idea of syntactic-graph representation and discuss its various properties. We claim that a syntactic graph carries complete syntactic information provided by a parse forest—the set of all possible parse trees.

AI87-63 **Developing Expert Systems from Examples and Explanations,** Joseph C. Ross, August 1987.

During the past two decades many expert systems have been developed which perform classification very well in narrow domains. This thesis examines this work, summarizing some useful ideas which have emerged from it, but also identifying the following problems: (1) the construction of these systems has been a very cumbersome process, and (2) the resulting knowledge bases have not been broadly applicable. This thesis proposes than an important reason for these problems has been the failure to retain all knowledge available from an expert during knowledge acquisition. The primary tasks performed by most experts are identified as consisting of (1) classification and (2) explanation.

Therefore, a representation is proposed in which classified examples and explanations of the classifications are retained explicitly. This approach greatly simplifies knowledge-base construction, since very little transformation must be performed by the knowledge engineer, and examples provide a natural way to systematically acquire and structure the knowledge. Furthermore, the two types of knowledge complement each other very well, and the representation is shown to be effective in supporting a variety of expert tasks. Finally, this thesis describes an actual implementation, in which these ideas were applied to the development of an expert system for the domain of clinical audiology.

AI87-62 **Bisecting the Version Space**, Anish K. Arora, Daniel L. Dvorak, Thomas C. Vinson, August 1987.

A class of problem in machine learning is concept learning where an experimenter poses questions to an oracle. The ideal experimenter learns the target concept with the fewest questions. For learners using Mitchell's version space algorithm (a.k.a. candidate elimination algorithm) for concept formation, Mitchell had conjectured that the best questions would be those that bisect the version space. To date, no published research has attempted to verify the conjecture (or even define what it means to "bisect" the version space). This paper reports the results of an effort to devise an ideal experimenter that has no a priori knowledge about the target concept. Several strategies were programmed and tested on single- and multiple-attribute learning on concept hierarchy trees and directed acyclic graphs. We obtained the best results with an experimenter that:

- selects the training instance based on a concept midway on a path between elements of the G and S sets bounding the version space, and
- for multiple-attribute concepts it modifies only one attribute at a time (divide-and-conquer).

For the special case where training instances are allowed to be non-leaf nodes in the concept hierarchy, an even better result can be obtained with the addition of a new rule for updating the version space.

AI87-61 **Critical Decisions under Uncertainty: Representation and Structure**, Benjamin Kuipers, Alan Moskowitz and Jerome P. Kassirer, August, 1987.

How do people make difficult decisions in situations involving substantial risk and uncertainty? In this study, we presented a difficult medical decision to three expert physicians in a combined "thinking aloud" and "cross examination" experiment. Verbatim transcripts were analyzed using script analysis to observe the process of constructing and making the decision, and using referring phrase analysis to determine the representation of knowledge of likelihoods. These analyses are compared with a formal decision analysis of the same problem to highlight similarities and differences. The process of making the decision resembles an incremental, sequential-refinement planning algorithm, where a complex decision is broken into a sequence of choices to be made with a simplified description of the alternatives. This strategy results in certain kinds of relevant information being under-weighted in the final decision. Knowledge of likelihood appears to be represented as symbolic descriptions capturing categorical and ordinal relations with "landmark" likelihoods, only some of which are described numerically. Numerical probabilities, capable of being combined and compared arithmetically, were not observed. These observations suggest an explanation for the heuristics and biases in human decision-making under uncertainty in terms of the processes that manipulate symbolic descriptions of likelihoods and construct plans of action for situations involving risk and uncertainty. *(Appears in Cognitive Science.)*

AI87-60 **Using Dominator-Modifier Relations to Disambiguate A Sentence**, Takanore Tsukada, August 1987. (Master's thesis)

This work introduces a system to parse and disambiguate sentences efficiently. The system consists of three parts: a compiler, a parser, and a disambiguation program. Dominator-modifier relations, which are edges between tow words, are produced to represent an ambiguous sentence. The relations are calculated using an efficient parsing method. They represent the ambiguities of a sentence clearly and facilitate ambiguity detection. These pairs and tree structures are produced by the parser and are used later in a disambiguation program. The parser uses a bottom-up parsing algorithm, the LRT(0) algorithm with dynamic programming. A compiler that translates Definite Clause Grammar rules into subprograms of the parser is also described. A disambiguation program is described to show how dominator-modifier relations are used for disambiguation by heuristics. This experiment shows that dominator-modifier relations make it easier to solve lexical and structural ambiguities.

AI87-59 **CALCLAB: An Interactive Environment for Scientific Numerical Problem Solving,** Yusuf N. Mauladad, August 1987. (Master's thesis)

CalcLab is an interactive environment for solving numerical problems in such areas as physics. Objects or situations are described by sets of variables. Physical principles and constraints are expressed as mathematical equations. Variables and equations can be connected to form models to solve numerical problems.

A primary goal of this research has been to design and implement a visual environment that is easy and intuitive to use. The user interface requires no programming language expertise. Several features have been incorporated to aid the user by automatically performing simple tasks. Another goal has been to enable the user to build a library of solution models that can be reused or modified. Such models can also be combined with others to solve more complex problems.

CalcLab is implemented in GLISP and runs on Xerox 1108 AI Workstations. It has been used to solve physics problems of the kind typically encountered by freshman physics students.

AI87-58 **TREAT: A Better Match Algorithm for AI Production Systems; Long Version,** Daniel P. Miranker, July 1987.

This paper contains a more detailed presentation of the TREAT match algorithm for AI Production systems than appears in the proceedings of AAAI-87 as well as an additional section sketching the compilation of the algorithm for sequential machines. The TREAT algorithm introduces a new method of state saving in production system interpreters called conflict-set support. Also presented are the results of an empirical study comparing the performance of the TREAT match with the commonly assumed best algorithm for this problem, the RETE match. On five different OPS5 production system programs TREAT outperformed RETE, often by more than fifty percent. This supports an unsubstantiated conjecture made by McDermott, Newell and Moore, that the state saving mechanism employed in the RETE match, condition-element support, may not be worthwhile.

AI87-57 **Recognition of Two Dimensional Objects Using Hypothesis Integration Technique,** Vincent S. S. Hwang, July 1987.

This paper describes an algorithm for recognition of two dimensional solid objects using a hypothesis integration technique. Matchings between model and image features are used to generate correspondence hypotheses that map the object into image coordinates. Consistent correspondence hypotheses are clustered and selected for verification. A refined correspondence is computed using all the matches participating in the selected cluster. The resulting correspondence is either verified or refuted by comparing the object directly against the image. To demonstrate the algorithm, we present some examples of recognizing some two dimensional rigid objects (objects can have arbitrary 2D translation, rotation and small amounts of foreshortening). Orthographic projection is assumed. Partial occlusion is allowed. This approach is robust and general. *(Appears in the Proceedings of the IEEE Workshop on Computer Vision, November, 1987.)*

• an order form is located at the end of the listings •

AI87-56 **Multiple Convergence: An Experiment in Disjunctive Concept Acquisition**, Kenneth S. Murray, July 1987.

Multiple convergence is proposed as a method for acquiring disjunctive concept descriptions. Disjunctive descriptions are necessary when the concept representation language is insufficiently expressive to satisfy the completeness and consistency requirements of inductive learning with a single conjunction of generalized features. Multiple convergence overcomes this insufficiency by allowing the disjuncts of a complex concept to be acquired independently. By summarizing correlations among features in the training data, disjunctive concepts can provide rich extensions to the representation language which may enhance subsequent learning. This paper presents the benefits of disjunctive concept descriptions and advocates multiple convergence as an approach to their acquisition. Multiple convergence has been implemented in an experimental learning system named HYDRA, and a detailed example of its execution is presented. *(Appears in the Proceedings of the Tenth IJCAI)*

AI87-55 Expert Systems for Monitoring and Control, D. Dvorak, May 1987.

Many large-scale industrial processes and services are centrally monitored and controlled under the supervision of trained operators. Common examples are electrical power plants, chemical refineries, air-traffic control, and telephone networks --- all impressively complex systems that are chal'enging to understand and operate correctly. The task of the operator is one of continuous, real-time monitoring and control, with feedback. The job can be difficult when the physical system is complex (tight coupling and complex interactions). Also, there may be faults not only in the system but also in its sensors and controls. Deciding the correct control action during a crisis can be difficult; a bad decision can be disastrous.

This paper surveys existing work in the field of knowledge-based systems that assist plant/process operators in the task of monitoring and control. The goal here is to better define the information processing problems and identify key requirements for an automated operator assistant. A high-level design of an ``expert system for operators" is presented. The design relies on a functional/causal model of the physical system as a source of deep knowledge (in addition to several sources of shallow knowledge). The major processes described in this design are *focusing, model-building, tracking, envisioning and advising.*

AI87-54 Coda: An Extended Debugger for PROLOG, D. Plummer, April 1987.

In this paper I describe Coda, an extension of the de facto standard debugger which presents more information about the execution of the program to the user as the program is debugged. Coda extends the standard debugger in a number of ways. First, Coda allows the user to interact with the pattern matching computation step. Thus the reason for the failure of a particular goal may be more precisely determined by the programmer. Second, Coda displays the program trace in terms of the clauses of the program rather than the goals that are executed. Thus, the program trace is directly related to the program that was written, and is at a level more appropriate to the programmer than that of the standard debugger. Finally, Coda allows finer control over the information that is displayed by the debugger, by an extended command set and a more powerful language for describing ``spy points".

AI87-53 Protos: An Exemplar-Based Le**ning Apprentice, E. R. Bareiss, B. W. Porter and C. C. Weir, April 1987.

Building Protos, a learning apprentice system for heuristic classification, has forced us to scrutinize the usefulness of inductive learning and deductive problem solving. While these inference methods have been widely studied in machine learning, their seductive elegance in artificial domains (e.g., mathematics) does not carry-over to natural domains (e.g., medicir :). This paper briefly describes our rationale in the Protos system for relegating inductive learning and deductive problem solving to minor roles in support of retaining, indexing, and matching exemplars. The problems that arise from "lazy generalization" are described along with their solutions in Protos. Finally, an example of Protos in the domain of clinical audiology is discussed. *(Appears in the Proceedings of the Fourth International Workshop on Machine Learning, University of California at Irvine, Morgan Kaufmann Publishers, 1987.)*

AI87-52 **TMYCIN Expert System Tool**, G. Novak, April 1987. (Revised, August 1988)

TMYCIN ("Tiny EMYCIN") is a simple expert system tool patterned after the EMYCIN tool developed at Stanford. TMYCIN does not attempt to provide all of the features of EMYCIN; it is intended to provide the most commonly used features in a package that is small and simple. The internal implementation of TMYCIN has been written from scratch and is therefore different from EMYCIN.

AI87-51 **The UTIPS Image Processing System**, V. Hwang, April 1987.

This document describes the UTIPS image processing software for the SYMBOLICS 3600 series Lisp machine with optional 8-bit color system running Genera 7.0 operating system. UTIPS contains more than 40 image manipulation and image processing functions and is implemented using Common Lisp

AI87-50 **A Survey of Psychological Models of Concept Representation**, E. R. Bareiss and B. W. Porter, March 1987.

This paper surveys the psychological literature on human representation of object concepts. There are three major views of how people represent concepts: the Classical, Probabilistic, and Exemplar Views. The Classical View holds that a concept is defined by a collection of features which are singly necessary and jointly sufficient for classifying an object as an instance of the concept. The Probabilistic View relaxes the notion of a definition to make classification a probabilistic function of the subset of concept members' possible features which are present in an object. The Exemplar View holds that a concept is defined extensionally; past instances of a concept are retained for use in classification. Each of these views is discussed in turn. Its strengths and weaknesses are pointed out, and major research in the spirit of that view is identified. The paper also provides an appendix which defines some of the terms used in this area of the psychological literature and discusses some important supporting ideas.

AI87-49 **A Prototype Natural Language Understanding Legal Consultant**, L. K. Branting, February 1987.

Because of the intimate connection between legal reasoning and natural language understanding, a robust natural language interface is important for a useful automated legal reasoning system. This project attempts to demonstrate how existing techniques of natural language understanding and discourse analysis could be used to construct a system able to interpret and resolve legal problems and express their solutions. The program uses frame instantiation both in text understanding and in legal analysis.

AI87-48 **A Program for Translating English Sentences into Lisp Programs**, A. Flatau, February 1987.

AI87-47 **Interpreting Narratives with Evaluative Statements**, A. Bhattacharjee, January 1987. (Master's Thesis)

Discourse structure is one of the foremost areas of study in natural language. Discourse is concluded to be governed by structures that go beyond the sentence or the relation between sentences. These structures capture a global purpose, which is the motivation for discourse. A form of such a global structure is an evaluative statement in discourse. This thesis is a case study of narratives and their associated evaluative statements. It is a computational attempt at reflecting the process by which an evaluative statement is used in interpreting narratives.

• an order form is located at the end of the listings •

AI87-46 **A Parallel Implementation of Iterative-Deepening-A*,** V. Nageshwara Rao, Vipin Kumar and K. Ramesh, February 1987.

This paper presents a parallel version of the Iterative-deepening A* (IDA*) algorithm. Iterative-Deepening-A* is an important admissible a' ithm for state-space search wh... has been shown to be optimal both in time and space for a wide v... ety of state-space search problems. Our parallel version retains all the nice properties of the sequential IDA* and yet does not appear to be limited in the amount of parallelism. To test its effectiveness, we have implemented this algorithm on Sequent Balance 8000 parallel processor to solve the 15-puzzle problem, and have been able to obtain almost linear speedups on the 9 processors that are available on our machine. On machines with larger number of processors, we expect that the speedup will still grow linearly. The parallel version seems suitable even for loosely coupled architectures such as the Hypercube.

AI87-45 **REVISED. Parallel Heuristic Search on a Shared Memory Multiprocessor,** V. Nageshwara Rao, V. Kumar, and K. Ramesh, June 1987.

In this paper we discuss two different ways of parallelizing the A* state space search algorithm for shared memory multiprocessors. We implement these parallel algorithms to solve the 15-puzzle problem and the TSP problem on the Sequent Balance 21000 and the Butterfly parallel processor. Our preliminary results are very encouraging, as we are able to obtain linear speedups upto 100 processors. Since the best-first search paradigm of A* is very commonly used, we expect these parallel versions to be effective for a variety of problems.

AI87-44 **Developments Towards Constraining Qualitative Simulation,** W. W. Lee, C. Chiu and B. J. Kuipers, January 1987.

Qualitative simulation is a useful and powerful tool for causal reasoning with physical mechanisms. However, qualitative simulation algorithms can predict impossible behaviors. In working with Kuipers' QSIM algorithm, an approach to identify and eliminate spurious predictions was developed. This approach addresses underlying shortcomings of qualitative simulation, which are due to a combination of locality and qualitative description. The differential equation describing a mechanism is analyzed to produce constraints which eliminate spurious predictions. This paper describes the application of this approach to the simulation of a damped spring. One local and two global constraints eliminated all spurious behaviors in simulating one period of oscillation. Some implications are discussed.

AI87-43 **A Survey of Specialized Parallel Architectures Designed to Support Knowledge Representation,** Daniel P. Miranker, January 1987.

While modern data processing systems have evolved from arithmetic calculating machines, knowledge representation schemes, developed for artificial intelligence programs, have evolved from cognitive models of intelligent behavior. Instead of the basic arithmetic operations used in data processing, knowledge representation schemes require extensive set manipulation, pattern matching and graph theoretic abilities. Thus, the implementation of artificial intelligence programs on conventional computers has suffered from a mismatch of basic abilities. This paper surveys computer architectures that attempt to overcome this mismatch by developing computer organizations that intrinsically support the structure and basic operations required to manipulate artificial intelligence knowledge bases. Common to all these proposals is the introduction of large scale parallelism in the form of many simple processing elements.

• an order form is located at the end of the listings •

AI86-42 **Extracting Data Base Knowledge from Medical Text**, Hae-Chang Rim and Robert F. Simmons, December 1986.

Relational database knowledge can be extracted automatically from appropriate sublanguage texts. This paper shows the feasibility of translating a particular sublanguage text concerning cold symptoms and treatments into the semantically uniform propositional forms, and thus a possible method for reducing the number of inference rules needed to relate a text to a question.

AI86-41 **REVISED. A Data-dependency Based Intelligent Backtracking Scheme for Prolog**, Vipin Kumar and Yow-Jian Lin, August 1987.

This paper presents a scheme for intelligent backtracking in Prolog programs. Rather than doing the analysis of unification failures, this scheme chooses backtrack points by doing the analysis of data dependency between literals. The other data-dependency based methods previously developed can not be easily incorporated in the Warren's abstract machine, and are not able to perform across-the-clause backtracking intelligently. Out scheme overcomes all these problems. For many problems this scheme is just as effective as intelligent backtracking schemes based upon (more accurate) analysis of unification failure, and yet incurs small space and time overhead. To demonstrate the usefulness of our scheme, we have modified a Warren's abstract machine simulator to incorporate our intelligent backtracking scheme, and have evaluated its performance on a number of problems.

AI86-40 **Question Answering with Rhetorical Relations**, Wanying Jin and Robert F. Simmons, December 1986.

A method is presented for computing discourse relations among text sentences and using the resulting data to generate new sentences to enrich a text knowledge base for the purpose of improving question answering. The coherence relations defined by Hobbs and Lockman and Klappholz, and the event coherence relations developed by Alterman are employed as a theoretical basis. Rhetorical sentences, which enrich the knowledge base, are derived from the discourse relations. The resulting text knowledge base is able to answer questions that would otherwise fail to match the tc

AI86-39 **An Intelligent Remote File Server**, Kim Korner, December 1986. (PhD Dissertation)

Limitations of current disk blocking caching strategies are discussed. A new model for providing remote file service using knowledge based caching algorithms is proposed. The knowledge based algorithms generate expectations of user process behavior which are used to provide hints to the file server. Surplus resources of the remote file server permit incorporation of these hints into caching algorithms. The research involves gathering trace data from a modified Unix kernel and driven simulation of remote file server models. Comparisons are made between conventional, knowledge based and optimal models. Further applications of knowledge based strategies in operating systems are discussed.

AI86-38 **Intuitionistic Modelling and Simulation of Mechanical Devices**, Chin Yang Chee, December 1986. (Master's Thesis)

The objective of this thesis is to explore and develop representational schemes to represent both the textual and pictorial description of a mechanical device (e.g. an engine) to produce an explanation of the device's operation by generating an animation of the device while highlighting the part of the text being simulated. For this purpose we have built a prototype which includes a highly interactive and hierarchical 2-D graphics editor to construct/edit devices and a semantic parser which parses the text into events of the device operation which is then used as a "script" for animating the device.

• an order form is located at the end of the listings •

AI86-37 **A Review of the First International Meeting on Advances in Learning**, Bruce W. Porter, November 1986.

This report is a review of the First International Meeting on Advances in Learning (IMAL) held the week of July 28, 1986 in Les Arcs, France. *(Appears in Machine Learning, Kluwer Academic Publishers, 1987)*

AI86-36 **A Framework for Intelligent Backtracking in Logic Programs**, Vipin Kumar and Yow-Jian Lin, November 1986.

This paper presents a scheme for intelligent backtracking in Horn-clause logic programs. The scheme is simple a. d y ?t effective for backtracking within a clause. We also present a framework for using extra analysis to make within-clause-backtracking even more intelligent and also to perform across-the-clause backtracking intelligently. The primary strength of our scheme over other schemes is that it incurs very small overhead and yet can eliminate a lot of redundant backtracking. Our backtracking scheme can also be used when AND-parallelism is exploited in logic programs (i.e., when multiple literals of a clause are executed simultaneously).

AI86-35 **PROTOS: An Experiment in Knowledge Acquisition for Heuristic Classification Tasks**, Bruce W. Porter and E. Ray Bareiss, August 1986.

PROTOS is an exp iment in acquiring, applying and revising expert knowledge for heuristic classification. Learned knowledge includes exemplar-based categories and domain knowledge to support explanation and pattern matching. Research on PROTOS has revealed fundamental issues in concept formation and classification concerning representation of ill-defined, "fuzzy" categories, multiple applications of learned knowledge, and the context in which learning takes place. This paper highlights these issues, describes the PROTOS approach to learning heuristic classification, and surveys related research.

AI86-34 **An Execution Model for Exploiting AND-Parallelism in Logic Programs**, Yow-Jian Lin and Vipin Kumar, September 1986.

This paper presents a parallel execution model for exploiting AND-parallelism in pure Horn Clause logic programs. The model is based upon the execution model of Conery and Kibler, and compares favorably with various related execution schemes (including Conery and Kibler's). We also present two implementation schemes to realize our model: one which has a coordinator to control the activities of processes solving different literals; and the other one which achieves synchronization by letting processes pass messages to each other in a distributed fashion. Trade-offs between these two schemes are then discussed.

AI86-33 **New Algorithms for Dependency-Directed Backtracking**, Charles J. Petrie, September, 1986. (Master's thesis)

The problem of providing dependency-directed backtracking in a Doyle-style Truth Maintenance System (TMS) is solved with three algorithms. The first modifies Doyle's basic algorithm by eliminating the necessity for conditional proof justifications. Unlike other attempts to do so, this method maintains the capability of the TMS to provide complete explanations from justifications. The second algorithm, which eliminates the generation of a maximal assumption set, provides a novel description of backtracking as a simple search. Finally, an extension of this algorithm allows contradiction resolution to be extended as an inference technique for expert systems. The dependency-directed backtracking algorithms presented are also the first to correctly ensure that an assertion is not justified unnecessarily and that unsatisfiable circularities are not created.

• an order form is located at the end of the listings •

AI86-32 **Talus: Automatic Program Debugging for Intelligent Tutoring Systems**, William R. Murray, August 1986.

This report summarizes the author's dissertation (AI86-27).

AI86-31 **A Rule Language for the GLISP Programming System**, Christopher A. Rath, August 1986. (Master's thesis)

GLISP is a programming language written by Dr. Gordon Novak at the University of Texas at Austin. It is an object oriented language based on LISP, a list processing language used in artificial intelligence programming. GLISP is often used in the university environment for the construction of expert systems programs, yet it has no formal rule language. The thesis is the design and implementation of a general rule language incorporated into GLISP. The project includes an investigation into currently implemented rule languages, an implementation of the rule language compiler in InterLisp, and sample applications. The resulting system is expected to be efficient, general purpose, and easy to learn for those programmers already familiar with GLISP.

AI86-30 **Metaphorical Shift and The Induction of Similarities**, Phillipe M. Alcouffe, July 1986. (Master's thesis)

This thesis presents a view of metaphor as the syntactic realization of an underlying cognitive process: analogical reasoning. The semantic changes resulting from the 'anomalous' juxtaposition of the metaphor's referents can be represented in a conceptual semantic space by rules of construal. These rules describe the semantic changes induced by a comparison component part of analogical reasoning. The meaning of metaphor is then a set of valid transferred similarities. Some similarities are pre-encoded as part of our general knowledge and are `discovered' through the transfer process, whereas other similarities are `created'. A taxonomy of conventional metaphorical themes is presented that supports the creativity of metaphor by the means of similarities induced by metaphorical shifts and inherited through this taxonomy.

AI86-29 **A Theory of Argument Coherence**, Wing-Kwong C. Wong, July 1986.

Previous research has suggested different approaches to analyze the coherence of discourses. But these approaches are not suitable to represent the structure of logical arguments. We develop a notion of reasoning relations and argue that it is more appropriate for characterizing argument coherence. Reasoning relations are naturally classified into two types, deductive and rhetorical, based on theories of classical logics and argumentation rhetorics. A set of commonly used relations is described. Ways of employing these relations to restructure and evaluate arguments are discussed. Possible connections between discourse coherence and Discourse Representation Theory are also explored.

AI86-28 **The Role of Inversion, Clecting and PP-Fronting in Relating Discourse Elements**, Mark V. Clefting, July 1986.

Languages use various types of syntactic constructions to convey more information than what is expressed by the sum of words in a sentence. This paper will explore and discuss the less obvious ways syntactic structure is used to convey information and how this information could be used by a natural language database system to organize and search a discourse space.

AI86-27 **Automatic Program Debugging for Intelligent Tutoring Systems**, William R. Murray, June, 1986. (PhD dissertation)

Program debugging is an important part of the domain expertise required for intelligent tutoring systems that teach programming languages. This dissertation explores the process by which student programs can be automatically debugged in order to increase the instructional capabilities of these systems. This research presents a methodology and implementation for the diagnosis and correction of nontrivial recursive programs. In this approach, recursive programs are debugged by repairing induction proofs in the Boyer-Moore Logic. The potential of a program debugger to automatically debug widely varying novice programs in a nontrivial domain is proportional to its capabilities to reason about computational semantics. By increasing these reasoning capabilities a more powerful and robust system can result. This thesis supports these claims by examining related work in automated program debugging and by discussing the design, implementation, and evaluation of Talus, an automatic debugger for LISP programs. Talus relies on its abilities to reason about computational semantics to perform algorithm recognition, infer code teleology and to automatically detect and correct nonsyntactic errors in student programs written in a restricted, but nontrivial, subset of LISP. Solutions can vary significantly in algorithm, functional decomposition, role of variables, data flow, control flow, values returned by functions, LISP primitives used, and identifiers used. Solutions can consist of multiple functions, each containing multiple bugs. Empirical evaluation demonstrates that Talus achieves high performance in debugging widely varying student solutions to challenging tasks.

AI86-26 **Symmetric Rules for Translation of English and Chinese**, Wanying Jin and Robert F. Simmons, May 1986.

A system of grammars using symmetric phrase structure and translation rules in a Lisp version of Prolog is shown to provide symmetric bidirectional translation between English and Chinese for a fragment of the two languages. It is argued that symmetric grammars and translation rules significantly reduce the total grammar writing requirement for translation systems, and that research on symmetric translation systems deserves further study.

AI86-25 **Fault Diagnosis Using Qualitative Simulation**, Ray Bareiss and Adam Farquhar, April 1986.

This paper presents a new approach to using qualitative simulation to diagnose faults in mechanisms. A mechanism is described by three models. A component model associates a set of measurement parameters with aspects of its physical structure. A constraint model defines relationships which must hold between parameters during the mechanism's normal behavior. A fault model embodies relevant aspects of a domain theory for a class of mechanisms. The Diagnose algorithm evaluates an observed state of a mechanism in terms of its constraint model. If violated constraints exist, Findfaults uses the component model to localize the problem and then uses the fault model to interpret it. These algorithms have been implemented in Prolog and have successfully identified faults in a variety of simple mechanisms.

AI86-24 **Qualitative Simulation as Causal Explanation**, Benjamin J. Kuipers, April 1986.

To give a causal explanation of an event means to deduce a statement which describes it, using as premises of the deduction one or more universal laws, together with certain singular statements, the initial conditions." (Karl Popper) Traditional diagnostic systems do not explain their findings in this sense. Using the QSIM qualitative simulation representation and algorithm, we demonstrate that we can construct models of physiological mechanisms, capable of explaining the observed behavior of the healthy mechanism, of the disease state, and of the response to therapy. We also present a new type of abstraction relation for linking simple equilibrium mechanisms into a hierarchical description of a complex mechanism such as human physiology.

AI86-23 **A Parallel Execution Scheme for Exploiting AND-parallelism of Logic Programs**, Yow-Jian Lin and Vipin Kumar, March 1986.

In this paper we present a parallel execution scheme for exploiting AND-parallelism of logic programs. This scheme follows the generator-consumer approach of the AND/OR process model. Using problem-specific knowledge, literals of a clause are linearly ordered at compile time. This ordering and run-time binding conditions are then used to dynamically select generators and consumers for different variables at run time. The scheme can exploit all the AND-parallelism available in the framework of generator-consumer approach. Compared with other execution schemes based on the same approach, our scheme is simpler, and potentially more efficient.

AI86-21 **GT: A Conjecture Generator for Graph Theory**, Wing-Kwong Wong, January 1986. (Master's thesis)

The process of knowledge acquisition can be automated with programs that learn from observation and discovery. A better understanding of this learning strategy would facilitate the construction of expert systems and the exploration of scientific domains. GT, a frame-based program written for this thesis, learns relationships among classes of graphs by observing the examples and non-examples of these classes. The relationships considered are subclass, superclass, exclusion, and complement exclusion. GT's knowledge is partly represented as frames and partly as formulas of predicate calculus. The learned relationships are stored hierarchically. GT has found non-trivial, well-known theorems of graph theory.

AI86-22 **An Intelligent Backtracking Algorithm for Parallel Execution of Logic Programs**, Yow-Jian Lin, Vipin Kumar and Clement Leung, March 1986.

In this paper we present a simple but efficient backtracking scheme which works when AND-parallelism is exploited in a logic program. The scheme is well suited for implementation on a parallel hardware. We show that the backtracking scheme presented by Conery and Kibler in the context of AND/OR process model is incorrect, i.e., in some cases it may miss solutions while performing backtracking. Even if no AND-parallelism is exploited (i.e., all literals are solved sequentially), our scheme is more efficient than the "naive" depth-first backtracking strategy used by Prolog because our scheme makes use of the dependencies between literals in a clause. Chang and Despain have recently presented a backtracking scheme which also makes use of the dependencies between literals. We show that our scheme is more efficient than their scheme in the sense that our scheme does less backtracking.

AI86-20 **Experimental Goal Regression: A Method for Learning Problem Solving Heuristics**, Bruce W. Porter and Dennis Kibler, January 1986.

This research examines the process of learning problem solving with minimal requirements for a priori knowledge and teacher involvement. Experience indicates that knowledge about the problem solving task can be used to improve problem solving performance. This research addresses the issue of what knowledge is useful, how it is applied during problem solving, and how it can be acquired. For each operator used in the problem solving domain, knowledge is incrementally learned concerning why it is useful, when it is applicable, and what transformation it performs. The method of experimental goal regression is introduced for improving the learning rate by approximating the results of analytic learning. The ideas are formalized in an algorithm for learning and problem solving and demonstrated with examples from the domains of simultaneous linear equations and symbolic integration.

AI85-19 **Explanation of Mechanical Systems Through Qualitative Simulation**, Stuart Laughton, December 1985. (Master's thesis)

This thesis documents an investigation into the feasibility of Qualitative Simulation as the basis of a computer program that can represent and explain the function and structure of mechanical systems. First, the general principles of qualitative reasoning and the theories of several researchers in this field are reviewed. Next, the process of modelling a specific simple mechanical device, a piston driven pump, and the resulting model's simulation output are discussed in detail. Finally, some techniques are presented for generating explanatory animation and natural language from a qualitative simulation model and its output.

• an order form is located at the end of the listings •

AI85-18 **Menu-Based Creation of Procedures for Display of Data**, Man-Lee Wan, December 1985. (Master's thesis)

We have investigated methods of creation of interface procedures that extract data needed by a library display procedure from user data of arbitrary structure and present it to the display procedure for generation of a display for presentation to the user. A procedure is created by the GLISP compiler from the following pieces of information:

1. the knowledge about how to produce the display, which is stored as a library procedure.
2. description of the user's data in the GLISP structure description language.
3. selection of the data from the user object that are to be displayed.
4. specification of the data needed by the display procedure.

A system called LINK has been written that creates and runs interface procedures automatically using the information outlined above. The LINK system has been integrated with the GEV data inspector of the GLISP system.

AI85-17 **The Map-Learning Critter**, Benjamin J. Kuipers, December 1985.

The Critter is an artificial creature which learns, not only the structure of its (simulated) environment, but also the interpretation of the actions and senses that give it access to that environment. The Map-Learning Critter embodies a strong a priori hypothesis: that its environment is, at least in part, structured as a large-scale space consisting of places connected by paths. The Critter's learning strategy begins by diagnosing its actions and senses. By performing experiments and examining the periodicity of its sense-impressions, it classifies actions as ``turn-like,'' ``travel-like,'' and ``others.'' After the actions are classified, it becomes possible to aggregate sets of sense-impressions to define places; then places are linked into paths. An exploration strategy assures that the entire environment will be explored and assimilated into this model. The Map-Learning Critter has been implemented and has experienced a variety of reasonable and unreasonable environments. Some implications of the results are discussed, and directions for future research are outlined.

AI85-16 **Negotiated Interfaces for Software Reusability**, Rick Hill, December 1985. (Master's thesis)

This thesis presents an algorithm which constructs an interface between a user's structure and a generic program, allowing the GLISP compiler to specialize the generic program to work with the user's data structure. The interface between the user's data and the data structure expected by the generic program is "negotiated" by the user through menu selection.

AI85-15 **The Knower's Paradox and the Logics of Attitudes**, Nicholas Asher and Hans Kamp, August 1985.

As Montague and Kaplan (Kaplan and Montaque, 1960, Montaque, 1963) pointed out long ago, a syntactic treatment of propositional attitudes has a fundamental weakness; virtually all of epistemic logic must be sacrificed, if it, like ordinary logic, is to be applicable to arbitrary subject matter. Thomason (Thomason, 1980) extended these results to include a syntactic treatment of the logic of belief, and showed how they also apply to analyses of the attitudes that, while not strictly "syntactic", use representations or structured propositions as the objects of the attitudes. In what follows, we shall call all such treatments "representational". The Hangman's Paradox as represented by Montague and Kaplan is another example of how even a minimal amount of epistemic logic in conjunction with a self referential attitude can lead to disaster. A more abstract formulation of the same self-referential attitude is exemplified in the "Knower's Paradox" and is perhaps best expressed in colloquial English by 'my negation is known' (we shall call this sentence the "Knower Sentence"). The usual moral drawn from these results is that a formally serious analysis of the attitudes should treat expressions of propositional attitudes, like modality, as predicates of propositions qua intensions, not as predicates of sentences or sentence-like entities. It is well-known how to provide a coherent logic of propositional attitudes with this approach. One of our main points, however, is that this "solution" is bought at far too dear a price: it leads neither to a satisfactory analysis of attitude reports nor to a general theory of attitudes. Since we think that these are desiderata of any theory of attitudes, we think that the standard "solution" is no solution at all, but rather refusal to take a theory of the attitudes seriously. A real solution to the Knower Paradox, one developed within the framework that leads to the desiderata, demands a thorough reanalysis of the logic of attitudes. We have both been developing such frameworks in other papers using discourse representation theory (Asher, 1984a, Asher, 1984b, Kamp, 1985a, Kamp 1985b). In this paper, however, we shall only make use of very general features of those frameworks that are relevant to the discussion of the Knower Paradox.

AI85-14 **Technologies for Machine Translation**, Robert F. Simmons, August 1985.

Advances in hardware have made available micro-coded LISP and PROLOG workstations, supported by text editing and formatting software. Some of these have been augmented with linguistic technology including large bilingual dictionaries, parsers, generators, and translators to make them powerful tools for research and development of automated translation. Some techniques of linguistic engineering for accomplishing translation are described, and it is suggested that the present barely satisfactory approach involving sentence-by-sentence translation will eventually be improved by incorporating the results of research on analyzing discourse.

AI85-13 **Computer Science and Medical Information Retrieval**, Robert F. Simmons, August 1985.

Presented at the *Conference on the Medical Information Sciences*, University of Health Science Center, San Antonio, Texas, July 1, 1985.

AI85-12 **Computational Treatment of Metaphor in Text Understanding: A First Approach**, Olivier Winghart, August 1985. (Master's thesis)

This thesis starts with a general presentation of the phenomenon of metaphor, necessary to define our object of study. It also contains an ordered presentation of the current research approaches to deal computationally with metaphors in text analysis. As an illustration, a case study system that analyses a mundane text (containing metaphors) is included, with a description of its functioning. Finally, more general issues are raised, that require elucidation to enlarge the field of application of such systems.

AI85-11 **Branch-And-Bound Search**, Vipin Kumar, August 1985.

AI85-10 **Parallel Processing for Artificial Intelligence**, Vipin Kumar, August 1985.

AI85-09 **A General Paradigm for AND/OR Graph and Game Tree Search**, Vipin Kumar, Dana S. Nau and Laveen N. Kanal, August, 1985.

This paper represents a general procedure fo ding an optimal solution tree of an acyclic AND/OR graph with monotone cost functions. Due to the relationship between AND/OR graphs and game trees, it can also be used as a game tree search procedure. Seemingly disparate procedures like AO*, SSS*, alpha-beta, B* are instantiations of this general procedure. This sheds new light on their interrelationships and nature, and simplifies their correctness proofs. Furthermore, the procedure is applicable to a very large class of problems, and thus provides a way of synthesizing algorithms for new applications. The procedure searches an AND/OR graph in a top-down manner (by selectively developing various potential solutions) and can be viewed as a general branch-and-bound (B&B) procedure.

• an order form is located at the end of the listings •

AI85-08 **A General Heuristic Bottom-up Procedure for Searching AND/OR Graphs,** Vipin Kumar, August 1985.

This paper presents a general heuristic bottom-up procedure for finding a least-cost solution tree of an AND/OR graph when the cost functions associated with the arcs are monotone. Since monotone cost functions are very general, the procedure is applicable to a very large number of problems. The procedure works for both cyclic and acyclic AND/OR graphs, and subsumes most of the known bottom-up procedures for searching AND/OR graphs. Many state-space search procedures and dynamic procedures are also special cases of our procedure.

AI85-07 **Heuristic and Formal Methods in Automatic Program Debugging,** William R. Murray, June 1985. (Appears in IJCAI-85 proceedings.)

TALUS is an automatic program debugging system that both detects and corrects nonsyntactic bugs in student programs written to solve small but nontrivial tasks in pure Lisp. TALUS permits significant variability in student solutions by using heuristic methods to recognize different algorithms and formal methods to reason about computational equivalence of program fragments. A theorem prover intentionally represents an infinite database of rewrite rules, thus allowing for unanticipated implementations. TALUS detects bugs using formal methods in both symbolic evaluation and case analysis. Heuristic methods conjecture the exact location of bugs and alterations necessary to correct the bugs. Finally, formal methods establish or disprove these heuristic conjectures, reflecting a generate and test methodology.

AI85-06 **Lisp Programming Lecture Notes,** Gordon S. Novak, July 1985.

This tutorial is intended as an introduction to Lisp programming for persons who already have experience programming in some language, e.g. FORTRAN. This course presents a set of basic system functions that are frequently used and are present in virtually every Lisp implementation.The material follows the conventions of Common Lisp. Five programming assignments are included.

AI85-05 **A Self Organizing Retrieval System for Graphs,** Robert A. Levinson, May 1985. (PhD dissertation)

This report describes the theory, design and implementation of a graph-based, self-organizing database retrieval system. The system is designed to support the expert problem solving tasks of recall, design and recovery. The fundamental design principle is the production of a partial ordering by the relation subgraph-of. This relation is considered to be equivalent to more-general-than. This document discusses this design from three different levels: an abstract level in which the nodes in the partial ordering are concepts, the implementation level described above (the nodes are graphs), and an application level in which the nodes are domain specific objects such as molecules or reactions. The primary problem domain explored is organic chemistry. A large database of organic reactions and starting materials can be queried to extract reactions or molecules that match, either exactly or approximately, desired structures. The system may also suggest precursors to a desired target molecule. The queries are answered by exploiting a set of concepts that are commonly subgraphs of molecule or reaction graphs. Concepts serve multiple purposes: They constrain the search involved in the matching process so that the time required to answer a query grows sub-linearly in the size of the database. Concepts define the notion of "similarity" that is crucial if approximate match is desired. They also may be useful generalizations of reactions or molecular structures. The concepts can be "discovered" (i.e., constructed) by the system itself using largely syntactic criteria based on the topology of the database. A variety of performance tests are performed, including a comparison of the system's precursor recommendation capability with graduate students in organic chemistry. The system is also applied to the retrieval and generalization of chess positions.

AI85-04 **Using and Revising Learned Concept Models: A Research Proposal,** Bruce W. Porter, May 1985.

People improve their learning performance with experience yet most machine learning systems do not The premise of this research is that a learning system can use learn knowledge to guide the acquisition of further knowledge. Furthermore, learned knowledge can guide the interpretation of training examples that are not strictly prototypical of the (potentially fuzzy) concept being learned. The learning system can revise this learned bias when new training violates expectations. This research combines results from learning from examples, learning with background knowledge, and learning with a domain model to study the progression from knowledge-poor to knowledge-rich learning. This research is directed toward development of a computational model of concept acquisition which learns, uses, and revises domain knowledge.

AI85-03 **Learning Problem Solving: A Proposal for Continued Research,** Bruce W. Porter, March 1985.

Many of the tasks which people perform involve problem solving: applying a sequence of operators to solve a problem. This research explores how efficient problem solutions are discovered from the myriad of less efficient alternatives. Results in machine learning are applied both to explain findings from psychological experimentation and to expand the utility of computers. Learning to problem solve requires acquiring multiple forms of knowledge. Problem solving is viewed as a search of a state-space formulation of a problem. With this formalism, operators are applied to states to transit from the initial state to the goal state. The learning task is to acquire knowledge of the state-space to guide search. In particular, three forms of knowledge are required: why each operator is useful, when to apply each operator, and what each operator does. This research builds on an existing PROLOG system that learns problem solving in the domains of simultaneous linear equations and symbolic integration. First the current learning system is described. Then new research directions are proposed. These include:

- critically comparing machine learning techniques demonstrated in a variety of problem solving domains.
- using learned knowledge to guide the acquisition of further learning.
- dynamically re-defining the concept description language by discovering useful descriptors from the training.

AI85-02 **Knowledge Based Contextual Reference Resolution for Text Understanding,** Michael Kavanaugh Smith, January 1985. (PhD dissertation)

This report extends the concept of reference resolution in a discourse context to cover a broad range of connective inference required for text understanding. Access to all conceptual relations is restricted or facilitated by the context established by preceding text. This contextual filter greatly simplifies the establishment of connections between the surface text and previously instantiated discourse representation. The reference procedure requires a taxonomically organized knowledge base of structured concepts, in the sense of frames and scripts. The procedure selects lexical senses and generates reference candidates, which may be either explicit or implicit in the discourse context. These are matched against constraints imposed by the surface text and a conceptual representation is constructed and integrated with the accumulated discourse structure.

AI84-05 **A Text Knowledge Base for the AI Handbook,** Robert F. Simmons, December 1983.

This research aims at defining a consistent set of text representation conventions for organizing fifty pages of the AI handbook as an inferential knowledge base founded on a procedural logic system of general inference schemas for answering questions from it. After a year of research on the AI handbook project, we have developed a prototype, natural-language, text knowledge system that includes a data base manager to compile the text knowledge and to make it available to navigational commands. The text is represented as logical propositions which form a set of text axioms to model its content. English questions and commands are translated to corresponding logical formulae and treated as theorems to be proved with respect to the text model. The logical form is that of semantic relations (SR) -- logical Predicates with varying numbers and orders of arguments. To compute effectively with such a free form, a relaxed unification procedure was defined as the basis of the SR theorem prover. The use of procedural logic augmented with fast, compiled Lisp functions has shown that questions can be answered in times ranging from a few tenths of a second to minutes of CPU time on a DEC2060 system The navigational capabilities of the data base manager make available larger contexts surrounding the text and offer the user complete freedom to explore the text and to extract any desired information from it.

AI84-04 **From Menus to Intentions in Man-Machine Dialogue,** Robert F. Simmons, November 1984.

Operating systems are designed to achieve goals, not to recognize user intentions. But the use of Help systems and Menu-selection make them more useful and friendly. Natural Language interfaces must go farther by guessing what the user wants and what is meant but not specified. Natural language programming systems -- still in infancy -- promise explicit capability for a user to define his/her intentions explicitly. Text Knowledge systems -- at the research frontier -- bewilder us with the complexity of intentions expressed and implied. Current techniques for recognizing intentions and computing appropriate representations and responses are discussed in this paper.

AI84-03 **Translating Horn Clauses From English,** Yeong-Ho Yu, August 1984. (Master's thesis)

This work introduces a small grammar which translates a subset of English into Horn Clauses. The parallel between the syntactic structures of sentences and corresponding literals of their Procedural Logic representation facilitates the translation. An interpreter is also described which accepts English descriptions of problem solving methods and facts about an example domain, the Blocks World. The interpreter translates then into Horn Clauses by using the grammar, and interprets them as programs and data. It also carries out commands on data by using the programs, and answers queries about data. In addition, it provides a mechanism called "Consistency Rules" which maintains integrity of the database. This experiment shows that Procedural Logic provides a unified system to accomplish a wide variety of computations, and that a small grammar without any semantic or domain-specific knowledge can translate a small subset of English sentences into Horn Clauses. Further research on several other domains is suggested to evaluate the usefulness of this translation.

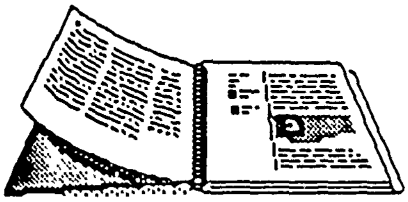
AI84-02 **Computing Discourse Conceptual Coherence: A Means to Contextual Reference Resolution,** Ezat Karimi, August 1984. (Master's thesis)

This thesis discusses the problem central to the interpretation of the discourse of a text: contextual reference resolution. The problem itself devolves to problems about the nature of the inferencing mechanism, knowledge base organization and discourse representation. A framework for an inferencing mechanism based on a theory of discourse coherence and focusing is discussed. A framework is described for a knowledge base which is composed of the knowledge about entities (through frame structures) and the knowledge about the coherence relations between different event/states. A model for discourse representation based on a small set of intersentential (coherence) relations and the relations between the conceptual structures for the entities discussed in the text is also presented.

AI84-01 **Artificial Intelligence Project at The University of Texas at Austin,** Gordon S. Novak and Robert L. Causey, et al., 1984.

This report is the technical part of a proposal to the U.S. Army Research Office (Electronics Division, Dr. Jimmie R. Suttle, Director) in late 1983 in response to their call for proposals on "Artificial Intelligence Research and Education." The University of Texas at Austin (along with The University of

Pennsylvania) was selected for substantial funding over a five-year period out of a total of thirty-five proposals submitted. This report also contains a subsequent proposal by Dr. Bruce Porter. The individual project proposals in this document illustrate the breadth of research in Artificial Intelligence at the University of Texas, though not all of the proposed projects were selected for funding. The research reports that are currently supported by the Army grant are those of Novak, Simmons, Kumar, and Porter.



Technical Report Order Form

University of Texas at Austin
Artificial Intelligence Laboratory
Taylor Hall 4.140
Austin, Texas 78712
Attention: Marlene Bateman

(512) 471-9567
(marlene@cs.utexas.edu)

Please enclose a check made out to the Artificial Intelligence Lab with your order.

NO CHARGE

☐ AI90-140	$ 4.00	☐ AI89-116	$ 4.00	☐ AI88-92	$ 4.00
☐ AI90-139	$ 6.00	☐ AI89-115	$ 3.00	☐ AI88-91	$ 5.00
☐ AI90-138	$ 3.00	☐ AI89-114	$ 4.00	☐ AI88-90	$10.00
☐ AI90-137	$ 3.00	☐ AI89-113	$ 3.00	☐ AI88-89	$ 3.00
☐ AI90-136	$ 3.00	☐ AI89-112	$ 4.00	☐ AI88-88	$[illegible]0.00
☐ AI90-135	$ 3.00	☐ AI89-111	$ 6.00	☐ AI88-87	$10.00
☐ AI90-134	$ 3.00	☐ AI89-110	$ 3.00	☐ AI88-86	$ 5.00
☐ AI90-133	$10.00	☐ AI89-109	$ 3.00	☐ AI88-85	$ 4.00
☐ AI90-132	$ 4.00	☐ AI89-108	$ 4.00	☐ AI88-84	$15.00
☐ AI90-131	$ 3.00	☐ AI89-107	$20.00	☐ AI88-83	$20.00
☐ AI90-130	$ 3.00	☐ AI89-106	$ 3.00	☐ AI88-82	$10.00
☐ AI90-129	$ 3.00	☐ AI89-105	$15.00	☐ AI88-81	$ 3.00
☐ AI90-128	$ 3.00	☐ AI89-104	$ 3.00	☐ AI88-80	$ 3.00
☐ AI90-127	$ 3.00	☐ AI89-103	$20.00	☐ AI88-79	$10.00
☐ AI90-126	$ 4.00	☐ AI89-102	$20.00	☐ AI88-78	$ 3.00
☐ AI90-125	$ 4.00	☐ AI89-101	$ 4.00	☐ AI88-77	$ 3.00
☐ AI90-124	$10.00	☐ AI89-100	$ 3.00	☐ AI88-76	$ 3.00
☐ AI90-123	$ 6.00	☐ AI89-99	$ 6.00	☐ AI88-75	$ 4.00
☐ AI90-122	$ 6.00	☐ AI89-98	$ 3.00	☐ AI88-74	$ 3.00
☐ AI90-121	$20.00	☐ AI89-97	$ 3.00	☐ AI88-73	$ 3.00
☐ AI90-120	$ 3.00	☐ AI89-96	$ 4.00	☐ AI88-72	$20.00
☐ AI90-119	$ 4.00	☐ AI89-95	$10.00	☐ AI88-71	$ 3.00
☐ AI89-118	$ 3.00	☐ AI89-94	$ 3.00	☐ AI88-70	$ 3.00
☐ AI89-117	$ 5.00	☐ AI88-93	$ 3.00	☐ AI88-69	$ 3.00

NO CHARGE

NO CHARGE

☐	AI88-68	$ 3.00	☐	AI86-	$ 4.00
☐	AI88-67	$10.00	☐	AI86-[illegible]	$15.00
☐	AI88-66	$20.00	☐	AI86-30	$15.00
☐	AI88-65	$ 6.00	☐	AI86-29	$ 3.00
☐	AI87-64	$ 4.00	☐	AI86-28	$ 3.00
☐	AI87-63	$10.00	☐	AI86-27	$20.00
☐	AI87-62	$ 3.00	☐	AI86-26	$ 4.00
☐	AI87-61	$ 5.00	☐	AI86-25	$ 6.00
☐	AI87-60	$15.00	☐	AI86-24	$ 4.00
☐	AI87-59	$15.00	☐	AI86-23	$ 3.00
☐	AI87-58	$ 3.00	☐	AI86-22	$ 3.00
☐	AI87-57	$ 3.00	☐	AI86-21	$10.00
☐	AI87-56	$ 6.00	☐	AI86-20	$ 5.00
☐	AI87-55	$ 3.00	☐	AI85-19	$10.00
☐	AI87-54	$ 3.00	☐	AI85-18	$ 6.00
☐	AI87-53	$ 3.00	☐	AI85-17	$ 3.00
☐	AI87-52	$ 4.00	☐	AI85-16	$15.00
☐	AI87-51	$ 3.00	☐	AI85-15	$ 3.00
☐	AI87-50	$ 3.00	☐	AI85-14	$ 3.00
☐	AI87-49	$ 6.00	☐	AI85-13	$ 3.00
☐	AI87-48	$ 3.00	☐	AI85-12	$10.00
☐	AI87-47	$ 6.00	☐	AI85-11	$ 3.00
☐	AI87-46	$ 3.00	☐	AI85-10	$ 3.00
☐	AI87-45	$ 3.00	☐	AI85-09	$ 5.00
☐	AI87-44	$ 4.00	☐	AI85-08	$ 3.00
☐	AI87-43	$ 5.00	☐	AI85-07	$ 3.00
☐	AI86-42	$ 3.00	☐	AI85-06	$15.00
☐	AI86-41	$ 3.00	☐	AI85-05	$10.00
☐	AI86-40	$ 5.00	☐	AI85-04	$ 3.00
☐	AI86-39	$20.00	☐	AI85-03	$ 3.00
☐	AI86-38	$10.00	☐	AI85-02	$10.00
☐	AI86-37	$ 3.00	☐	AI84-05	$ 3.00
☐	AI86-36	$ 3.00	☐	AI84-04	$ 3.00
☐	AI86-35	$ 3.00	☐	AI84-03	$10.00
☐	AI86-34	$ 5.00	☐	AI84-02	$10.00
☐	AI86-33	$ 6.00	☐	AI84-01	$ 3.00

NO CHARGE

NO CHARGE

NO CHARGE

END 3-8-91