

AD-A127 369

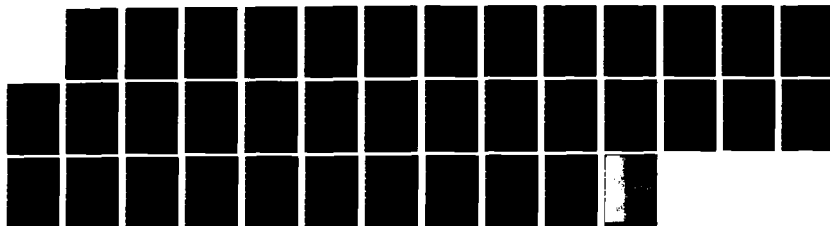
SUMMARIES OF PAPERS CONTAINED IN MACHINE INTELLIGENCE
10 CONFERENCE PROCE. (U) CASE-WESTERN RESERVE UNIV
CLEVELAND OH V PAO 1982 AFOSR-TR-83-0286 AFOSR-82-0015

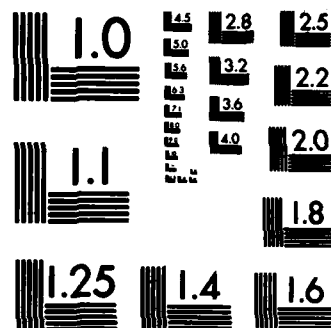
1/1

UNCLASSIFIED

F/G 5/2

NL





MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

AFOSR-TR- 83 - 0286

(3)

Final

SUMMARIES OF PAPERS CONTAINED
IN MACHINE INTELLIGENCE 10, CONFERENCE PROCEEDINGS FOR
THE MI 10 INTERNATIONAL CONFERENCE ON KNOWLEDGE ENGINEERING.

HELD AT CASE WESTERN RESERVE UNIVERSITY

NOVEMBER 11 - 13, 1982

CLEVELAND, OHIO 44106

AFOSR-82-0015

DTIC
JUL 1983
A

Approved for public release;
distribution unlimited.

83 04 28 069

AD A127359

DTIC FILE COPY

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER AFOSR-TR- 83 - 0286	2. GOVT ACCESSION NO. <i>AD-A127369</i>	3. RECIPIENT'S CAT. LOG NUMBER
4. TITLE (and Subtitle) SUMMARIES OF PAPERS CONTAINED IN MACHINE INTELLIGENCE 10, CONFERENCE PROCEEDINGS FOR THE MI 10 INTERNATIONAL CONFERENCE ON KNOWLEDGE ENGINEERING		5. TYPE OF REPORT & PERIOD COVERED FINAL, 15 OCT 81-14 APR 82
7. AUTHOR(s) Yoh-Han Pao		6. PERFORMING ORG. REPORT NUMBER
9. PERFORMING ORGANIZATION NAME AND ADDRESS Department of Computer Science & Engineering Case Western Reserve University Cleveland OH 44106		8. CONTRACT OR GRANT NUMBER(s) AFOSR-82-0015
11. CONTROLLING OFFICE NAME AND ADDRESS Mathematical & Information Sciences Directorate Air Force Office of Scientific Research Bolling AFB DC 20332		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS PE61102F; 2304/A2
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)		12. REPORT DATE
		13. NUMBER OF PAGES 33
		15. SECURITY CLASS. (of this report) UNCLASSIFIED
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE
16. DISTRIBUTION STATEMENT (of this Report) Approved for public release; distribution unlimited.		
17. DISTRIBUTION STATEMENT (of this abstract entered in Block 20, if different from Report)		
18. SUPPLEMENTARY NOTES PROCEEDINGS FOR THE MI 10 INTERNATIONAL CONFERENCE ON KNOWLEDGE ENGINEERING, CASE WESTERN RESERVE UNIVERSITY, Nov 11-13, 82, Cleveland, Ohio		
19. KEY WORDS (Continue on reverse side if necessary and identify by block number)		
20. ABSTRACT (Continue on reverse side if necessary and identify by block number) This paper contains summaries of papers presented at the conference on the following topics: <u>Mechanized Reasoning</u> ; - (1) On Generating and Using Examples in Proof Discovery; (2) A First-Order Formalisation of Knowledge and Action and Action for a Multi-Agent Planning System; (3) Knowledge-Based Problem-Solving in AL3; (4) A Provably Correct Advice Strategy for the End-Game of King and Pawn Versus King; <u>Reasoning About Computations</u> - (5) Mechanical Theorem-Proving in the Case Verifier; (6) Computational Frames and Structural Synthesis of Programs; <u>Acquisition and Matching of Patterns</u> - (7) Semi-Autonomous Acquisition of Pattern-Based Knowledge; (8) Revealing Conceptual Structure in Data by (CONT.)		

DD FORM 1473
1 JAN 73

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

88 04 28 069

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE(When Data Entered)

ITEM #20, CONTINUED: Inductive Inference; (9) Fast Memory Access by Similarity Measure; (10) A Robot Vision Lab Concept; (11) Interpreting Line-Drawings as 3-Dimensional Surfaces; Problems of Robotics - (12) Practical Machine Intelligence; (13) Programmable Assembly System Research and its Applications; Knowledge-Based Systems - (14) New Research on Expert Systems; (15) Application of the Prospector System to Geological Exploration Problems; (16) XSEL: A Computer Sales Person's Assistant; (17) Knowledge-Based Programming Self-Applied; The Roles of Knowledge and Deduction in Algorithm Design; (19) Natural Language Dialogue Systems: A Pragmatic Approach; Logic Programming - (20) LOGLISP: An Alternative to PROLOG; (21) Programming With Full First-Order Logic; (22) Higher-Order Extensions to PROLOG: Are They Needed?; (23) PROLOG: A Language for Implementing Expert Systems; (24) Chess End-Game Advice: A Case Study in Computer Utilisation of Knowledge; Programs of the Brain - (25) RANA Computatrix: An Evolving Model of Visuo-Coordination in Frog and Toad; (26) The Computational Problem of Motor Control; (27) Brains, Machines and Crystals of Knowledge; Philosophies of Man and Machine - (28) Artificial Intelligence, Philosophy and Existence Proofs; and (29) Ethical Machines.

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE(When Data Entered)

MECHANIZED REASONING

1

ON GENERATING AND USING EXAMPLES IN PROOF DISCOVERY

A. M. Ballantyne and W. W. Bledsoe
Departments of Mathematics and Computer Science
University of Texas at Austin, Austin, Texas



This paper describes some work on automatically generating finite counterexamples in topology, and the use of counterexamples to speed up proof discovery in intermediate analysis, and gives some example theorems where human provers are aided in proof discovery by the use of examples.

This paper is divided into two parts.

The first part is on counterexamples, their automatic generation and use as filters to speed up proofs. The results of two different experiments dealing with counterexamples are presented in the first part. The first experiment deals with the construction of finite counterexamples to false conjectures in elementary point set topology. A program was constructed that incrementally tried to build an interpretation for a finite collection of sets. Although this experiment required that the program discover interpretations for sets and families of sets (topologies), it nevertheless succeeded in this difficult task by

- (1) considering only interpretations consisting of finite families of finite sets;
- (2) using considerable knowledge about elementary topology to organize the search.

The second experiment deals primarily with using counterexamples as a

AIR FORCE OFFICE OF SCIENTIFIC RESEARCH (AFSC)
NOTICE OF TRANSMITTAL TO DTIC
This technical report has been reviewed and is
approved for distribution under AFM 190-12.
1 Distributable unlimited.
MATTHEW J. KEMPER
Chief, Technical Information Division

subgoal filter in analysis. In this project, interpretations for universally quantified function and constant symbols were supplied manually. The program then extended that interpretation to include the skolem functions and constants that arose as a result of quantifier elimination.

The second part of this paper is on using examples as a positive force in theorem proving whereby the examples suggest lines of attack rather than act as filters to prune bad subgoals.

2

A FIRST-ORDER FORMALISATION OF KNOWLEDGE AND ACTION AND ACTION
FOR A MULTI-AGENT PLANNING SYSTEM

K. Konolige
Artificial Intelligence Center
SRI International, USA

The author is interested in constructing a computer agent whose behaviour will be intelligent enough to perform cooperative tasks involving other agents like itself. The construction of such agents has been a major goal of artificial intelligence research. One of the key tasks such an agent must perform is to form plans to carry out its intentions in a complex world in which other planning agents also exist. To construct such agents, it will be necessary to address a number of issues that concern the interaction of knowledge, actions, and planning. Briefly stated, an agent at planning time must take into account what his future states of knowledge will be if he is to form plans that he can execute; and if he must incorporate the plans of other agents into his own, then he must also be able to reason about the knowledge and plans of other agents in an appropriate way.

In this paper the author developed a formalism for reasoning about knowledge, belief, and action; showed how this formalism can be used to deal with several well-known problems and then described how it could be used by a plan constructing system.

To summarize the contributions of this paper: The author has defined a syntactic approach to the representation of knowledge and belief in which the key element is the identification of beliefs with provable expressions in a theory of the object language. The technique of *semantic attachment* to the intended interpretation of the metalanguage provability predicate has been advanced as a method of simplifying proofs by directly modelling an agent's inference procedure, rather than simulating it.

To unify a formalization of knowledge and action, the author showed how to formalize the interaction of knowledge and action within the syntactic framework. The benchmark example was a presentation of a *test* in which an agent uses his knowledge of observable properties of the world and the way actions affect the world to discover the state of an unobservable property. Finally, the author pointed out how the formalization could be used in a planning system.

3

KNOWLEDGE-BASED PROBLEM-SOLVING IN AL3

I. Bratko
Institute Jozef Stefan and Faculty of Electrical Engineering
University of Ljubljana

AL3 (Advice Language 3) is a problem-solving system whose structure facilitates the implementation of knowledge for a chosen problem-domain in

terms of plans for solving problems, 'pieces-of-advice', patterns, motifs, etc. AL3 is a successor of AL1 and AL1.5. Experiments in which AL1 was applied to chess endgames established that it is a powerful tool for representing search heuristics and problem-solving strategies. The power of AL1 lies mainly in the use of a fundamental concept of AL1: piece-of-advice. A piece-of-advice suggests what goal should be achieved next while preserving some other condition. If this goal can be achieved in a given problem-situation (e.g., a given chess position) then we say that the piece-of-advice is 'satisfiable' in that position. In this way AL1 makes it possible to break the whole problem of achieving an ultimate goal into a sequence of subproblems, each of them consisting of achievement of a subgoal prescribed by some piece-of-advice. The control structure which chooses what piece-of-advice to apply next consists of a set of 'advice-tables', each of them being specialized in a certain problem-subdomain. Each advice-table is a set of rules of the form

if precondition then advice-list

If more than one rule-precondition is satisfied then simply the first rule is chosen.

This comparatively simple control structure has several advantages: simplicity, neatness of solutions, susceptibility to formal proofs of correctness of strategies. However, its disadvantage is that it is difficult to implement problem-solving strategies which make extensive use of higher-order concepts, such as plans, and which also do 'meta-level' reasoning about plans and pieces-of-advice themselves.

AL3 is an attempt at facilitating the use of higher-order concepts by providing a more flexible control structure over the basic mechanisms of AL1.

The main modules of the AL3 system are:

- (1) a knowledge-base which contains *methods* that 'know' how to solve particular problems, and *lemmas* (or theorems) about the problem-domain that, hopefully, can be applied during the problem-solving process;
- (2) a current-knowledge-pool (CKP) containing the already known *facts* and *hypotheses* about the problem being solved, and other objects that are relevant to the problem;
- (3) the *control module* which decides what method, or lemma, to activate next.

There are several ways of looking at the AL3 system. One possible view is that AL3 is a theorem-prover which accepts a problem in the form of a formula to be proved or disproved. If the formula is neither a theorem nor a contradiction then AL3 tries to find new facts about the problem. The new facts are then used as additional axioms for proving or disproving the initial formula. In this sense each executorial cycle aims at producing the most useful new axioms, such that they bring the formula as close as possible towards a theorem or a contradiction.

Another view of AL3 is that AL3 is a problem-solver which uses a special formalism for problem representation. This formalism can be thought of as a generalization of two known schemas for problem-representation: the state-space representation and the AND/OR graph representation.

The experiments with AL3, described in this paper, were implemented in PROLOG. The problem-domain used in these experiments is a chess ending: king and pawn *vs.* king and pawn with passed pawns.

These experiments demonstrate that AL3 can be used for knowledge-based problem-solving using higher-order concepts.

4

A PROVABLY CORRECT ADVICE STRATEGY FOR THE END-GAME
OF KING AND PAWN VERSUS KING

T. B. Niblett
Machine Intelligence Research Unit
University of Edinburgh, UK

This paper describes a strategy for the endgame king and pawn *vs.* king (KPK), and its implementation. Consideration is given to the knowledge representation issues involved, and to methods by which the strategy can be demonstrated correct.

The advice strategy for KPK was produced within the framework of chess textbook knowledge and was developed within the framework of AL1, the main idea of which is "piece-of-advice". The strategy itself is written as a set of logical rules. A broadly effective measure of the search required by a rule is the maximum search depth required by that strategy. To determine the maximum search depth for the various methods in the advice text a database method is used. A new mechanism for search using sets of squares instead of individual squares were used.

The advice strategy has been partially implemented in PROLOG on the Edinburgh/ICF DEC 10. A straight comparison between the square sets algorithm and a normal alpha-beta search over a restricted sample of positions shows an improvement of 50-100 fold in nodes searched.

The technique of search using square sets demonstrates that with an appropriate search representation the task of supplying the tactical information necessary for efficient play can be made much easier. The separation of the tactical and strategic components of the endgame allows for a straightforward proof of correctness for the advice language strategy. Even for an endgame as simple as KPK any more detailed proof involving a fully determined search strategy is unrealistic.

REASONING ABOUT COMPUTATIONS

5

MECHANICAL THEOREM-PROVING IN THE CASE VERIFIER

G. W. Ernst and R. J. Hookway
Case Western Reserve University
Cleveland, Ohio USA

The Case verifier is a man-machine system for constructing a mathematical proof that a module of a computer system meets its specifications. This paper views the Case verifier as an expert intelligent system. After a brief overview of the entire system, the remainder of the paper focuses on the theorem proving component which is the main source of intelligence in the verifier.

The main contribution of this research is as a case study in the use of mechanical theorem proving in a non-trivial application. The inference rules have a tendency to produce duplicate subgoals because often the application of two rules commute.

The mechanical search strategy in the theorem prover avoids many of these duplicates. It only applies case analysis once to a given subgoal.

The order in which inference rules are applied can always be overridden by the user. Most of the other inference rules generate the potential for duplicate subgoals, but this is not a problem for the theorem prover because they are used exclusively under user control.

The input programming language is a large subset of MODULA including abstract data types, and the verifier should be capable of proving, with

reasonable human assistance, components of 'realistic' computer systems. Non-trivial examples are used to motivate the discussion.

6

COMPUTATIONAL FRAMES AND STRUCTURAL SYNTHESIS OF PROGRAMS

E. H. Tyugu
Institute of Cybernetics
Estonian Academy of Sciences
Tallinn, USSR

Computational frames represent the meaning of concepts in terms of computability. They can be used for describing situations and expressing problem conditions as any kind of frames. On the other hand - the computational frames can be converted into special theories in which structural synthesis of programs is applicable.

The following two general ideas lie behind the structural synthesis of programs:

- A special theory is built for every particular problem. The theory can be tailored to fit the problem and to facilitate the proof of an existence theorem needed for program synthesis.
- Only structural properties of computations are used almost everywhere in the proof. The correctness of primitive steps of computations is not proved at all. It is assumed that if something can be computed, then it is computed correctly. This may be justified by the consideration that to describe the correctness conditions for primitive steps of computations is not more reliable than to describe the steps of computations themselves.

The starting point for the program synthesis was in the artificial intelligence field where problem solvers were built. Now the program synthesis

is growing out of the limits of AI and is becoming a topic for system programmers. As a consequence of this there are higher requirements for the efficiency and the reliability of a program synthesizer.

ACQUISITION AND MATCHING OF PATTERNS

7

SEMI-AUTONOMOUS ACQUISITION OF PATTERN-BASED KNOWLEDGE

J. R. Quinlan
Baser Department of Computer Science
University of Sydney

This paper contains a brief survey of some existing inductive inference systems namely Meta-Dendral, INDUCE and THOTH-P, and a more detailed examination of the particular system used for these experiments.

This paper discussed three themes:

- (1) The task of acquiring and organizing the knowledge on which to base an expert system is difficult.
- (2) Inductive inference systems can be used to extract this knowledge from data.
- (3) The knowledge so obtained is powerful enough to enable systems using it to compete handily with more conventional algorithm-based systems.

These themes are explored in the context of attempts to construct high-performance programs relevant to the chess endgame king-rook versus king-knight.

A description of the general induction system ID3 that was used for the experiments reported in this paper is given. The knowledge discovered is in the form of a decision tree for differentiating objects of one class from another.

The last 2- and 3-ply experiments discussed in this paper demonstrate

that expert systems built on knowledge inferred from data by an inductive system can match more conventional programs. These experiments were conducted over complete databases. Other experiments indicate that a decision tree formed from only a small part of a collection of instances is accurate for a large proportion of the remainder. However, in the latter case, the decision tree produced from incomplete databases will most likely be inexact and will have to be modified as exceptions are discovered.

8

REVEALING CONCEPTUAL STRUCTURE IN DATA BY INDUCTIVE INFERENCE

R. S. Michalski and R. Stepp
University of Illinois
Urbana, USA

In many applied sciences there is often a problem of revealing a structure underlying a given collection of objects (situations, measurements, observations, etc.). A specific problem of this type is that of determining a hierarchy of meaningful subcategories in such a collection. This problem has been studied intensively in the area of cluster analysis. The methods developed there, however, formulate subcategories ('clusters') solely on the basis of pairwise 'similarity' (or 'proximity') of objects, and ignore the issue of the 'meaning' of the clusters obtained. The methods do not provide any description of the clusters obtained. This paper presents a conceptual clustering method which constructs a hierarchy of subcategories, such that an appropriately generalized description of each subcategory is a single conjunctive statement involving attributes

of objects and has a simple conceptual interpretation. The attributes may be many-valued nominal variables or relations on numerical variables. The hierarchy is constructed in such a way that a flexibly defined 'cost' of the collection of descriptions which branch from any node is minimized.

Experiments with the implemented program, CLUSTER/paf, have shown that for some quite simple problems the traditional methods are unable to produce a structuring of objects most 'natural' for people, while the method presented here was able to produce such a solution.

9

FAST MEMORY ACCESS BY SIMILARITY MEASURE

Y-H. Pao and G. P. Hartoch
Case Western Reserve University
Cleveland, USA

In artificial intelligence research, procedures for storing away a large number of individual facts and for retrieving them efficiently are of great interest and importance. At present only two basic techniques are used, one being hashing and the other that of matching the query pattern. The authors describe in this paper a specific information storage and retrieval procedure which is rapid and economical in use of storage space, and is especially appropriate if retrieval is on the basis of similarity rather than exact match. The new development of the associative memory for content addressable storage described in this paper is a technique for reducing the 'noise' otherwise present in such structures. The relationship between this structure and those proposed previously by others is also discussed.

10

A ROBOT VISION LAB CONCEPT

T. Vamos and M. Bathor
Computer and Automation Institute
Hungarian Academy of Sciences
Budapest, Hungary

An experimental laboratory is described, with interactive graphics facilities and a wide range of programming tools for generation of application systems. The philosophy of the system and relevant components, especially for man-machine communication and system design, is reported with application examples.

The general methodology is illustrated by some examples.

- (1) Bus-body cover sheet identification task: This task is the control of a pointing robot in a plant producing about 14,000 buses/year. The problem is typically two-dimensional and content-sensitive.
- (2) Crop sorting in canning factories: A shape and color check is carried out both for regular selection and for detecting irregularities or aberrations.
- (3) Assembly-line part-selector: In most cases the task can be solved as a reduced 3D one.
- (4) Gamma-camera scintillograms: The task is the processing of shape-changes, and evaluation of relevant diagnostic characteristics.
- (5) Fan-assembly for buses: This is a sequence of simple assembly tasks performed partly by the robot and partly by special-purpose devices such as that used, for example, for driving of screws.
- (6) Check of tool-edges: This is a very important problem in a workshop environment and is crucial in an unmanned production cell.

11

INTERPRETING LINE-DRAWINGS AS 3-DIMENSIONAL SURFACES

H. G. Barrow and J. M. Tenenbaum
Artificial Intelligence Center
SRI International, USA

This paper's authors propose a computational model for interpreting line drawings as three-dimensional surfaces, based on constraints on local surface orientation along extremal and discontinuity boundaries. Specific techniques are described for two key processes: recovering the three-dimensional conformation of a space curve (e.g., a surface boundary) from its two-dimensional projection in an image, and interpolating smooth surfaces from orientation constraints along extremal boundaries.

Some important unsolved problems remain. The proposed technique for interpreting a three-dimensional space curve is slow and ineffective on noisy image curves. Also the surface interpolation technique must be extended to handle partially constrained orientations along discontinuity boundaries.

PROBLEMS OF ROBOTICS

12

PRACTICAL MACHINE INTELLIGENCE

E. D. Sacerdoti
Machine Intelligence Corporation
Palo Alto, California, USA

This paper will discuss three specific areas of work in machine intelligence that MIC feels are ripe for commercial application: machine vision, natural-language access to computers, and expert systems.

In the machine vision area, MIC's initial product, the VS-100, was developed to provide a broad range of image processing capabilities, rather than to perform optimally on a specific limited task. It recognizes and inspects images of complex objects against a contrasting background in 1/2 second to several seconds, depending on the complexity of the image. It accepts grey-scale data which it thresholds to produce a binary image. Object recognition of each region in the scene can be performed using a nearest neighbor classifier operating on a user-selectable subset of the features.

In the natural language processing area, MIC has developed the core of a microcomputer-based language processing system, which can provide most of the capabilities of the large systems in a much more cost-effective fashion.

The market for professional-quality expert systems is highly dependent on the particular subject, and is very intensive in its use of technical

experts. Thus, MIC expects this business to grow very slowly in the short term.

13

PROGRAMMABLE ASSEMBLY SYSTEM RESEARCH AND ITS APPLICATIONS

J. L. Nevins, D. E. Whitney
The Charles Stark Draper Laboratory, Inc.
Cambridge, Massachusetts

S. C. Graves
Massachusetts Institute of Technology

The long range goal of the Draper Laboratory programmable assembly system research is better understanding of design and operation problems in both conventional and novel assembly systems. The novel systems are based on new technology such as robots, smart robot wrists, computer controlled groups of machines, and model mix assembly.

The authors note that it is not unusual for Draper to have close working relationships with industry on problems with both an evolving knowledge base and a developing technology.

Part of this report is developed to describing the techniques developed, the status of these relationships, and the initial results. The remainder of the report gives a brief status report on the programmable assembly system research and its future aims and directions.

This method of coupling on-going research activities to specific applications has some positive aspects to it. It clearly identifies the

limitations to the present-but constantly evolving-knowledge base. Furthermore, it helps to both identify, as well as focus attention on, new fruitful research areas.

It appears that consideration of multi-level systems in present day factories will pose new options for organizing factories - perhaps the precursor to the type of organization research needed for the consideration of unmanned manufacturing systems.

KNOWLEDGE-BASED SYSTEMS

14

NEW RESEARCH ON EXPERT SYSTEMS

B. G. Buchanan
Computer Science Department
Stanford University, USA

In this paper the author gives a brief survey of the current state of research on expert systems. Then the directions of future work are discussed.

The state of the art of expert systems technology is advancing, but it is necessary to look at existing limitations as well as the potential power of this approach. The following table lists many characteristics of what can currently be done.

Expert Systems: State of the art

- o Narrow domain of expertise
- o Limited language for expressing facts and relations
- o Limiting assumptions about problem and solution methods
- o Stylized I/O languages
- o Stylized explanations of line of reasoning
- o Little knowledge of own scope and limitations
- o Knowledge base extensible but little help available for initial design decisions
- o Single expert as "knowledge czar"

As for the future work on expert systems much has to be done on the extensions of problem solving, controlling search and inference, representing facts and relations about the world, understanding language and visual scenes, etc.

15

APPLICATION OF THE PROSPECTOR SYSTEM TO GEOLOGICAL EXPLORATION PROBLEMS

J. Gaschnig
Artificial Intelligence Center
SRI International, USA

In this paper the author describes several applications of the PROSPECTOR consultation system to mineral exploration tasks. One was a pilot study conducted for the National Uranium Resource Estimate program of the U.S. Department of Energy. This application estimated the favorability of several test regions for occurrence of sandstone uranium deposits. For credibility, the study was preceded by a performance evaluation of the relevant portion of PROSPECTOR's knowledge base, which showed that PROSPECTOR's conclusions agreed very closely with those of the model designer over a broad range of conditions and levels of detail. A similar uranium favorability evaluation of an area in Alaska was performed for the U.S. Geological Survey. Another application involved measuring the value of a geological map. The author comments on characteristics of the PROSPECTOR system that are relevant to the issue of inducing geologists to use the system.

This work illustrates that expert system intended for actual practical use must accommodate the special characteristics of the domain of

expertise. In the case of economic geology, it is not rare for field geologists to disagree to some extent about their field observations at a given site. Accordingly, the use of various sorts of sensitivity analysis is stressed in PROSPECTOR to bound the impact of such disagreements and to isolate their sources.

16

XSEL: A COMPUTER SALES PERSON'S ASSISTANT

J. McDermott
Carnegie-Mellon University
Pittsburgh, USA

R1, a knowledge-based configurer of VAX-11 computer systems, began to be used over a year ago by Digital Equipment Corporation's manufacturing organization. The success of this program and the existence at DEC of a newly formed group capable of supporting the development of programs that can be used in conjunction with R1. This paper describes XSEL, a program being developed at Carnegie-Mellon University that will assist salespeople in tailoring computer systems to fit the needs of customers. XSEL will have two kinds of expertise: it will know how to select hardware and software components that fulfil the requirements of particular sets of applications, and it will know how to provide satisfying explanations in the computer system sales domain.

Though the XSEL program is more ambitious than R1 was, it will become useful long before it reaches maturity. XSEL is sufficiently developed so that within a few months it will be available for use, in conjunction with

R1, to aid salespeople in entering orders and in determining the precise set of components that a customer needs. As knowledge is extracted from configuration design experts, it will be given to XSEL.

17

KNOWLEDGE-BASED PROGRAMMING SELF-APPLIED

C. Green and S. J. Westfold
Systems Control Inc.
Palo Alto, California

A knowledge-based programming system can utilize a very-high-level self description to rewrite and improve itself. This paper presents a specification, in the very-high-level language V, of the rule compiler component of the CHI knowledge-based programming system. From this specification of part of itself, CHI produces an efficient program satisfying the specification. This represents a modest application of a machine intelligence system to a real programming problem, namely improving one of the programming environment's tools - the rule compiler. The high-level description and the use of a programming knowledge base provide potential for system performance to improve with added knowledge.

Most of the programming knowledge used by the system is expressed explicitly, usually in some rule form, and may be manipulated in that form. This collection of programming rules is used by the system to help in selecting implementation techniques, and to help in other programming activities such as editing and debugging. The programming knowledge base includes stepwise refinement or transformation rules for optimizing, sim-

plifying and refining data structures and control structures.

The language V is used not only to specify programs but also to express the knowledge base of synthesis rules and meta-rules.

The authors have succeeded in creating a very-high-level description of RC, a rule compiler written in LISP that compiles the production rule subset of the V language into efficient LISP code, in the V language. The adequacy of the description has been tested by having RC compile itself, i.e., the original version of RC in LSIP compiled the V description of RC into LISP. This newly-compiled LISP program was then tested by having it compile its V description.

18

THE ROLES OF KNOWLEDGE AND DEDUCTION IN ALGORITHM DESIGN

D. R. Barstow
Computer Science Department
Yale University, USA

In this paper two basic approaches in applying AI techniques to the problem of automatic program synthesis are discussed. In the earliest attempts deduction (that is, the use of a general purpose mechanism such as a theorem prover) played a central role. The basic idea was to rephrase the program specification as a theorem to be proved or a problem to be solved. If a proof or solution was found, it could be transformed into a program satisfying the original specification.

The second approach, an application of the knowledge engineering

paradigm, is based on the assumption that the ability of human programmers to write programs comes more from access to large amounts of knowledge about specific aspects of programming than from the application of a few general deductive principles.

In the rest of this paper, a way to combine these two approaches is suggested.

19

NATURAL LANGUAGE DIALOGUE SYSTEMS: A PRAGMATIC APPROACH

L. I. Mikulich
Institute of Control Sciences
USSR Academy of Sciences
Moscow, USSR

This paper describes an approach for designing a natural language interface with data bases, that can be useful in real applications on relatively small computers. Pragmatically-oriented analysis of utterances made by casual users is a key concept of the approach. Using the pragmatic approach the author designed several systems intended for practical applications. A family of systems called DISPUT has been developed for application in management information systems for the merchant fleet. Design of DISPUT dialogue systems illustrates the structure of such systems in general; interaction through an interpreter in a natural language is discussed in detail.

LOGIC PROGRAMMING

20

LOGLISP: AN ALTERNATIVE TO PROLOG

J. A. Robinson and E. E. Sibert
School of Computer and Information Science
Syracuse University, USA

In this paper the authors give a summary description of the system resulting from the effort to create within LISP a faithful implementation of the logic programming idea, which the authors call LOGLISP. LOGLISP can be thought of as LISP + LOGIC, where LISP is what it always has been and LOGIC is the collection of new primitives which are added to LISP. In LOGIC assertions, queries, and all other logic-programming constructs are represented as LISP data-objects. The basic process of the LOGIC system is the use of a deduction cycle to compute the answer to a given query. Examination of the deduction cycle reveals that it consists of growing a tree (the deduction tree) whose nodes are implicit constraints.

In testing LOGIC with a variety of examples it has been found that the deduction tree grows at rates of 50 or 60 nodes per second for most examples. An extended example shows some of the main features of LOGLISP at work.

21

PROGRAMMING WITH FULL FIRST-ORDER LOGIC

K. A. Bowen
School of Computer and Information Science
Syracuse University, USA

An automatic deduction system based on a modification of Gentzen's sequent system LJ is presented, and its use as the basis for a logic programming system is described. The system is a natural extension of Horn clause logic programming systems in that when all of the formulas in the input sequent are atomic, the behavior of the system mimics that of LUSH resolution systems. The use of such systems in program development systems and in database management systems is discussed.

The examples indicate that a system based on the principles described in this paper will provide a valuable tool for both program development and database query. Efficient implementation of the basic system along the lines of Warren et al. (1977) and Robinson and Sibert (1980), will make it practical and friendly to the user. One can even envisage a compiler built on principles similar to those of the Edinburgh Horn clause compiler. These are directions for further research. Another line of investigation would be to push the basic system still further by extensions to more powerful cut-free systems.

22

HIGHER-ORDER EXTENSIONS TO PROLOG: ARE THEY NEEDED?

D. H. D. Warren
Department of Artificial Intelligence
University of Edinburgh

PROLOG is a simple and powerful programming language based on first-order logic. This paper examines two possible extensions to the language which would generally be considered "higher-order".* The first extension

introduces lambda expressions and predicate variables so that functions and relations can be treated as 'first class' data objects. This extension does not add anything to the real power of the language. The other extension concerns the introduction of set expressions to denote the set of all (provable) solutions to some goal. This extension does indeed fill a real gap in the language, but must be defined with care. There are many problems which cannot be expressed in pure PROLOG without such an extension. It is important for set expressions to be "backtrackable," so that they can be used freely in any context.

* Throughout this paper, 'higher-order' is used in the informal (computing) sense of 'pertaining to functions, sets, relations, etc.'. The use of this term should not be taken to imply any particular connection with higher-order logic.

23

PROLOG: A LANGUAGE FOR IMPLEMENTING EXPERT SYSTEMS

K. L. Clark and F. G. McCABE
Department of Computing
Imperial College, London

In this paper the logic programming language PROLOG is briefly described, concentrating on those aspects of the language that make it suitable for implementing expert systems. It is shown that features of expert systems such as:

- (1) Inference generated requests for data,
- (2) Probabilistic reasoning,
- (3) Explanation of behavior

can be easily programmed in PROLOG. Each of these features is illustrated by showing how a fault finder expert could be programmed in PROLOG.

24

CHESS END-GAME ADVICE: A CASE STUDY IN COMPUTER UTILISATION OF KNOWLEDGE

M. H. van Emden
Department of Computer Science
University of Waterloo, Canada

The intended contribution of this paper is the development of tools, in the form of a computer language, for the automatic utilization of knowledge. The field of study, having as objective to make such a utilization possible, is called 'knowledge engineering'. One proposed application is the development of a new breed of textbooks: such a textbook would have the form of an interactively usable computer program and database. The source code would be readable as a textbook, although more systematic, explicit, and precise than the ones we are used to. In interactive use the computerized textbook would solve problems in the application domain of the embedded knowledge. In this mode the user can be ignorant of everything except perhaps the broad outlines of the textbook's contents.

The purpose of the work reported in this paper is to investigate whether logic as a programming language (Kowalski, 1974) can take the place of the advice language proposed by Michie and for expressing knowledge for purposes of knowledge engineering. This strategy followed in this paper is to introduce basic notations of game playing and logic programming by means of the game of Nim. These introductory sections conclude

with a logic program for Nim. Subsequently the basic notions are refined and extended to be applicable to subgames of the rook endgame of chess.

PROGRAMS OF THE BRAIN

25

RANA COMPUTATRIX: AN EVOLVING MODEL OF VISUO-COORDINATION IN FROG AND TOAD

M. A. Arbib
Department of Computer and Information Science
University of Massachusetts at Amherst

Frogs and toads provide interesting parallels to the way in which humans can see the world about them, and use what they see in determining their actions. What they lack in subtlety of visually-guided behavior, they make up for in the amenability of their behavior and the underlying neural circuitry to experimental analysis. This paper presents three specific models of neural circuitry underlying visually-guided behavior in frog and toad. They form an 'evolutionary sequence' in that each model incorporates its predecessor as a subsystem in such a way as to explain a wider range of behavior data in a manner consistent with current neurophysiology and anatomy. The models thus form stages in the evolution of Rana computatrix, an increasingly sophisticated model of neural circuitry underlying the behavior of the frog.

Three important features of the style of modeling developed in this paper were noted:

(1) New phenomena are addressed not by the creation of ad hoc models but by the orderly refinement and expansion of models already created. (2) Each model presented in this paper is in fact a model-family. (3) The choices are only loosely constrained by the experimental data at present available.

THE COMPUTATIONAL PROBLEM OF MOTOR CONTROL

T. Poggio, B. L. Rosser
Max-Planck-Institut für Biologische Kybernetik
Tübingen, FRG

In this paper the authors suggest that the motor control systems, as complex systems that process information, must be analyzed and understood at several different levels. At the lowest level there is the analysis of basic components and circuits, the neurons, their synapses, etc. At the other extreme, there is the study of the computations performed by the system - the problems it solves and the ways that it solves them - and the analysis of its logical organization in terms of its primary modules. Each of these levels of description, and those in-between, has its place in the eventual understanding of motor control by the nervous system.

The problem of trajectory control in multi-jointed limbs with several degrees of freedom is a central problem to any motor system, biological or artificial, at the computational level. In this paper the most fundamental characteristics of the problem of trajectory control is outlined and one of the classical approaches to it from the field of robotics is considered. Finally, the implications of some recent findings concerning the way in which antagonistic muscle pairs may be used in the control of the equilibrium position are discussed in detail.

27

BRAINS, MACHINES AND CRYSTALS OF KNOWLEDGE

D. N. Spinelli and F. E. Jensen
Department of Computer and Information Science
University of Massachusetts, Amherst, USA

The neocortex of the brain is organized in areas dedicated to specific processes. These cortical areas seem essential in the performance of tasks which are deemed important to intelligence, such as pattern recognition, decision making, learning and goal seeking. Natural brains still hold unchallenged superiority to AI machines in such tasks. Recently it has been discovered that early experience in developing animals strongly augments the size of a cortical representation responsible for specific behavioral process, if reinforcement makes it important to the animal. This reallocation of neural elements, caused by experience during development, to form an aggregate of neurones dedicated to a specific process can be viewed as a Crystal of Knowledge which, in concert with other analogous systems provided by the genome, form the structural foundations for adaptive behavior. The authors present here a working hypothesis for a mechanism capable of explaining such an aggregate. The data presented in this paper, combined with the rich collection of ideas generated by neuroscience and AI research, provided inspiration to guide this work on the simulation of neuron-like elements.

PHILOSOPHIES OF MAN AND MACHINE

28

ARTIFICIAL INTELLIGENCE, PHILOSOPHY AND EXISTENCE PROOFS

R. J. Nelson
Case Western Reserve University
Cleveland, USA

The connection between AI and mechanism is discussed in this paper from a psychological and philosophical point of view. Mechanism is the philosophy that the human mind is an information processing system. The author also discussed

- (1) the general nature of adequacy conditions in relation to the now classic treatment of them in formal logic,
- (2) conditions that should be satisfied by programs or Turing machine constructions,
- (3) how to demonstrate that expectation is analyzable in automata theoretic terms.

29

ETHICAL MACHINES

I. J. Good
Virginia Polytechnic and State University
Blacksburg, USA

The notion of an ethical machine can be interpreted in more than one way. Perhaps the most important interpretation is a machine that can generalize from existing literature to infer one or more consistent ethical systems and can work out their consequences. An ultra-intelligent machine should be able to do this, and that is one reason for not fearing it.

END

FILMED

5-83

DTIC