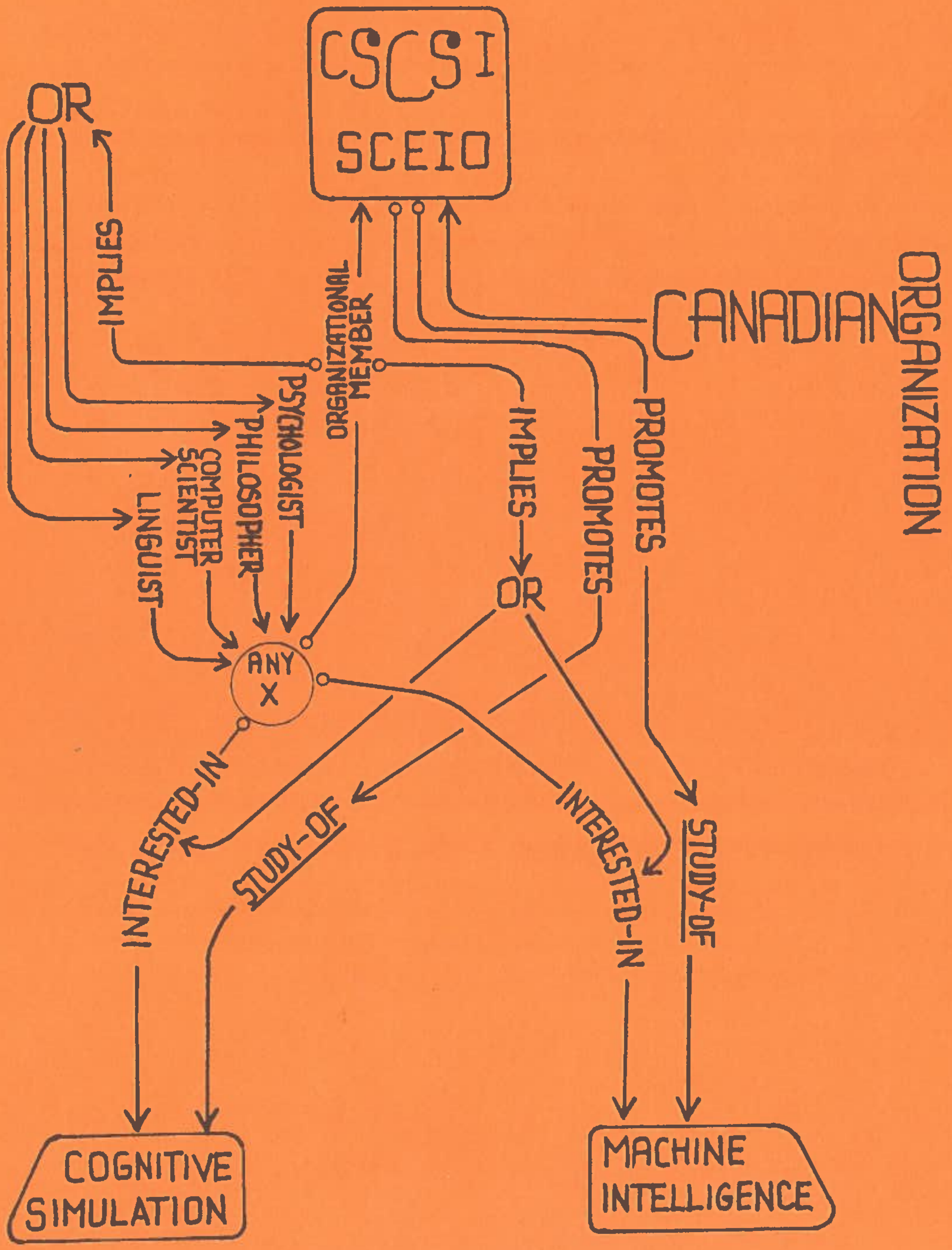
vol.1/no.4

December 1977

cscsi/sceio newsletter

an occasional publication of the Canadian Society for Computational Studies of Intelligence // Société Canadienne des Etudes d'Intelligence par Ordinateur



CONTENTS

Edited at the University of Alberta,

Department of Computing Science

EDITORIAL REMARKS

Your editorial crew once again received a large volume of material summarizing research and teaching activities on the Canadian and international AI scenes. We are most grateful to all the contributors, and regret that space considerations required the abbreviation of some of the longer submissions. This issue contains several special features, including an announcement and Call for Papers for the next CSCSI/SCEIO conference, in Toronto next July. There is also a special section on game playing, which includes a report on the recent computer chess tournament at the ACM National Conference. This issue concludes with a selection of puzzles which offer considerable AI content (and fun).

The big Canadian AI event since the appearance of the last newsletter was, of course, the CSCSI/SCEIO conference at UBC in August of 1976. The printed proceedings contain some two dozen papers in areas ranging from vision, knowledge representation and natural language to the social consequences of AI. What the proceedings cannot convey is the aura of enthusiasm and producitivty that pervaded both the technical sessions and the well-attended social functions. We are all looking forward to more of the same in Toronto next summer.

Acknowledgements. Your editorial staff was greatly assisted by Mrs. Sandra Wilkins of the U. of Alberta Computing Science Department, and by the text-entry staff in the Computing Services Department. Special thanks to our graduate student, Anne Brindle, for her superb Double-Crostic puzzle.

Len Schubert
Jeff Sampson

ANNOUNCEMENT

CSCSI/SCEIO Conference

As indicated in the Call for Papers which follows, the Second National Conference of CSCSI/SCEIO will be held in Toronto on July 19-21, 1978. Dave Waltz, SIGART Chairman, has agreed in principle that SIGART would officially cooperate with us, but we still need ACM approval; this has been requested. Our meeting will be held during the week preceding the proposed ACL/TINLAP conference in Urbana-Champaign, July 26-27, and we hope that many will attend both gatherings. The success of the Conference depends on your papers, your attendance, and your participation, as well as those of your friends. See you in Toronto!

C. Raymond Perrault
General Chairman

CALL FOR PAPERS

Second National Conference
of
Canadian Society for Computational Studies of Intelligence/Societe Canadienne pour Etudes d'Intelligence par Ordinateur

Toronto, Canada
19-21 July, 1978

Some suggested, although not exclusive, topics of interest are: natural language understanding, heuristic problem solving and game playing, automatic programming and debugging, computer perception, psychological aspects of A.I., automatic theorem proving, representation of knowledge, applications of A.I., and social consequences of A.I.

You are invited to submit six copies of a detailed summary to the program chairman

Professor E. W. Elcock
Dept. of Computer Science
University of Western Ontario
London, Ontario, Canada M6A 5B7

All submissions will be read by several members of the program committee and evaluated on the basis of significance, originality, and overall quality. To make refereeing possible it is important that the summary (1) convey the novel ideas; (2) contain enough information about the scope of the work; and (3) include comparisons to the relevant literature. We suggest a total of between 5 and 10 double spaced pages.

Program Committee: E.W. Elcock (Chairman), John S. Brown, Gordon McCalla, Jerry Hobbs, Ray Reiter, Steven Zucker.

The deadline for submissions of summaries is January 31, 1978. Authors will be notified of acceptance or rejection by March 15, 1978. The accepted papers, typed on special forms, will be due on May 15, 1978, and should be sent to the general chairman:

Professor C. Raymond Perrault
Dept. of Computer Science
University of Toronto
Toronto, Ontario, Canada M5S 1A7

Information about local arrangements will be available from the general chairman. Proceedings will be distributed at the conference, and will be subsequently available for purchase from CSCSI/SCEIO.

COURSES AND PROGRAMS OF STUDY

University of British Columbia
Department of Computer Science
Vancouver, B.C.

Computer Science 502: Artificial Intelligence I

1 1/2 Units
Fall 1977
Instructor: A. K. Mackworth

As C.Sc. 512 "LISP-based Symbolic Computation" is not offered this year, the first half of this course is an intensive introduction to LISP and its applications. Application areas in artificial intelligence such as problem-solving, graph searching, pattern-matching and elementary theorem proving are considered. These areas introduce some of the early artificial intelligence paradigms.

In the second half of the course, further developments in artificial intelligence are studied in the context of computational vision. The central question is, "How can we represent and use knowledge for perception?" Starting with the most primitive pattern recognition proposals we pursue the answers to that question through a variety of representational schemes up to current schemata-based systmes. Issues of segmentation and interpretation are tied to control structure problems. The variety of proposed solutions is related to the development of AI programming languages such as PLANNER, CONNIVER, KRL and MAYA.

Students are be required to design and implement solutions to four or five assignments.

Text: Winston, P.H. Artificial Intelligence, Addison-Wesley, 1977
Documentation and notes on LISP, PLANNER,...will be made available.

Other Programs at UBC

At the University of British Columbia we are launching a new Interdisciplinary Graduate Program in Remote Sensing. This brings together researchers in Computer Science, Forestry, Soil Sciences, Geography, and Electrical and Civil Engineering. In support of the program we will be acquiring a digital colour image processing and display system in the Department of Computer Science. Our work on the interpretation of LANDSAT images will benefit enormously from the system.

Other vision research, on sketch map interpretation, 3D scene analysis and languages for perception, continues.

Simon Fraser University
Computing Science Program
Burnaby, B.C.

Precariously balanced atop the precipice know as Burnaby Mountain one finds the pristine (yet precocious) University dedicated to the memory of Simon Fraser, explorer. Nestled somewhat inconspicuously within the University's bosom, the Computing Science Program (re department) has become active like a long dormant volcano suddenly sprung to life by an irritating world around it.

While many diverse activities occur in the SFU Computing Science Program (due to its interdisciplinary nature) there are some undertakings which should be of particular interest to CSCSI/SCEIO readers.

Equipment Grant

A modest Simon Fraser University President's Research Grant, for the purchase of equipment, has been awarded to Nick Cercone of Computing Science for the construction of a special purpose research tool for Artificial Intelligence research. The proposal for the grant is in the Research Summaries Section. Any suggestions and/or comments from CSCI/SCEIO members would be appreciated.

Graduate Studies in Computing Science

Presently a proposal for a new graduate studies program in Computing Science at Simon Fraser University is traversing its way through various university committees, paving the way for possible implementation by Fall, 1978. While it is still incomplete, it is possible to give the flavour of the proposed program as it stands now.

The Computing Science Program has developed rapidly at Simon Fraser University over the past several years. The swift growth and maturation which has occurred at the undergraudate level should now be continued at the graduate level. The reasons for this further formal development within the Computing Science Program are many and varied. They include: community need for such a program (undergraduate enrolment in Computing Science has climbed from 0 to 800 in the past 5 years, and demand for graduate studies is so persistent that 3 students have been admitted into Master's programs under Special Arrangements); the availability of qualified faculty and grant money; and the desirability of extending our unique inderdisciplinary undergraduate program to the graduate level.

The following list of areas (drawn up for comprehensive examination purposes) gives an indication of the proposed structure of the graduate program. The choice of areas was influenced by categories in use at other universities, by faculty interests and also by the desirability of identifying areas of comparable scope. The underlying principle for chosing these particular areas is that well-educated computing scientists ought to have some knowledge of each area. The six areas are: (1) Theoretical Computing Science; (2) Artificial Intelligence; (3) Programming Languages; (4) Programming Systems; (5) Computer Design and Organisation; (6) Advanced Applications (e.g., graphics, cartography, medical applications, operations research).

Research Facilities

Equipment available for research and instruction includes a Varian V75 with 64K of core memory, assorted peripheral equipment; and a data link to the Computing Centre's 370/155 mainframe. The department is also equiped with an Evans and Sutherland Picture System I, a DEC PPDP 11/34 and a microprocessor laboratory containing six Intel 8080A based mini-micro designers, other microprocessors, a 16 channel logic anlayzer, and a micro ultraviolet erasable PROM programmer, among other hardware. Also available for research use is a Hewlett Packard 2116 System operated by the Psychology Department and a DEC GT40 graphics computer operated by the Kinesiology Department.

Computing Science 280-3 (3-1-0) & 380-3 (3-1-0):

Computation in the Humanities I & II

These courses are designed to present the student with an overview of the techniques, methodology, and use of computers in humanistic work. Students will be introduced to the concepts of several programming languages with emphasis on the uses to which they have been put for various types of humanistic research.

Students will be expected to work on individual projects. In exceptional cases a group project may be sanctioned. Topics for projects can be picked by the student (after consultation) or may (by default) be assigned. In addition, one exam (take home, open notes, essay-type) will be given sometime toward the end of the term, and several homeworks will be assigned.

1. Artificial Intelligence and its relationship to the humanities.
   What is Artificial Intelligence [AI]? AI research methods. Future?

2. Introduction to programming languages for humanistic endeavours.
   Basic constructs, examples, structures, and uses of PL/1, LISP, SNOBOL (SPITBOL), ALGOL (ALGOLW, PASCAL).

3. Text Processing.
   Classical methods of text processing (data representation, various kinds of indicies, concordances and frequency counts, etc.); editing; concordances; cryptography; style analysis.

4. Machine Dictionaries.
   Goals and functions in linguistic research; compiling, look-up; updating; survey of existing dictionaries; word morphology.

5. Linguistic descriptions of text for humanistic research.

   Phonological: transcriptions; rhymes; metrics; models of phonetic change; other phonological features.

   Morpho-syntactic: word segmentation: simple parsing without complete sentence analysis; word class tagging; phrase structure and deep structure oriented parsing.

   Semantical: semantic features; semantic structure, representation, and handling; content analysis; discourse analysis.

6. Statistical description of verbal data.
   Methods for pattern distinguishing; stylistic description; authorship attribution; psycholinguistic and sociolinguistic researches; language acquistion and learning.

7. Data Base concepts.
   Data base storage and structure; data retrieval; bibliographies; catalogues; thesauruses; archives; interactive use of data bases; information retrieval systems; organisation.

8. Natural Language Processing and Understanding by Machine.

   Knowledge Representation: declarative and procedural knowledge; semantic networks; frames; scripts; plans; schemata; models.

   Reasoning: cognitive simulation; formal reasoning; deduction; inference; heuristic methods.

   Descriptions of working Natural Language Understanding systems: early systems; Winograd, Wilks, Schank, etc.

   Speech: recognition and understanding; special requirements; review of actual projects.

9. Workshops and Discussions.

   Text processing in Law; Computer produced braille; The computer in art and visual display; Computer analysis, synthesis, and composition in music; Computer generated stories; Quantative methods in non-verbal oriented research including history, archeology, fine arts, and music;

Computing Science 410-3 (3-1-0): Artificial Intelligence

This course is designed to present the student with an introduction and overview of Artificial Intelligence. Students will be introduced to LISP programming and will be required to write several programs in LISP.

Students will be expected to work on individual projects. In exceptional cases a group project may be sanctioned. Topics for projects can be picked by the student (after consultation) or may (by default) be assigned. In addition, one midterm exam will be given, a final exam, and several homeworks will be assigned. A topic outline follows:

1. Meaning, Goals, and Methods of Artificial Intelligence.
2. LISP programming and LISP techniques.
3. State Space Representations and Search Methods.
4. Game Playing and Heuristic Programming.
5. Pattern Recognition and Classification.
6. Theorem Proving using the Resolution Principle.
7. Question-Answering Systems.
8. Natural Language Representation and Understanding.
9. Computer Perception and Vision.

Faculty at SFU

The following is a partial list of faculty with interests related to Artificial Intelligence.
T. Peucker - Computer Cartography and Mapping
D. Kirkpatrick - Complexity of Algebraic and Combinatorial Algorithms
N. Cercone - Natural Language Representation, Understanding and Processing
T. Calvert - Neural Models and Computer Assisted Movement Analysis.

While we are only beginning to develop programs and tools for advanced research at Simon Fraser the future looks bright. An accidental side effect resulting from the initial implementation of Computing Science at Simon Fraser promises to have profound effects concerning artificial intelligence research in the program. Initially, Computing Science was established with a plethora of joint-appointed faculty with existing departments such as geography, kinesiology, commerce, chemistry, mathematics, and the fine arts program. In recent years additional faculty, equipped with computing science degrees, have been hired. This interdisciplinary meld promises to enhance the possibility of applying artificial intelligence techniques to current problems in other disciplines as well as to advance the techniques for new problems.

THE SECOND COMPUTER GO-MOKU TOURNAMENT

Lawrence J. Mazlack
Computing and Information Science
University of Guelph
Guelph, Ontario

On November 27 and 28, 1976, the second computer GO-MOKU tournament was held. Eleven programs competed (up from four in 1975). The programs displayed a considerably higher standard of play than in 1975. (The 1975 competition, along with rules for computer play, was described on pp. 9-12 of Vol 1, No. 3 of this Newsletter).

The format of the tournament was verbal relay of moves from one program to another, using Guelph's facilities as a clearing house. Several non-Guelph programs were run on Guelph equipment or equipment available to Guelph through a local computer resource pool. The programs were run either by local volunteers or by their programmers. The standings might have been somewhat different if various problems had not occurred. Several competitors had significant software and/or hardware problems.

One item of theoretical interest was the relative non-success of SHEIN. Its programmer is the third ranked Canadian GO player, and an accomplished FORTRAN programmer.

The final standings and a sample game are displayed on the following two pages.

The final standings were as follows:

Standing	Program	Programmer(s)	Computer	Language	Plies	Core (K)
1	PLUNC	E. JOHNSON A. COSTON Chapel Hill,NC	PDP 11/45	ELEVEN -TRAN	3	18
2	ARTHUR	M. COMPTON Montreal, Que.	IBM 370/158	PL/I	2-7	VS
3	GOMOKU	D. WALDEN Skokie, Ill.	XEROX 530	XSYMBOL	1-7	7
4	SHIFTY	J. DAY Cupertino, CA.	IBM 370/155	FORTRAN	1-4?	330
4	PHANTOM	T. HEAVEN London, Ont.	PDP 10	POP-10	4	33
6	WINR	M. OUYE Acampo, CA.	MICRODATA 1800	ASSEMBLER	5	10
7	FIVE-IN -A-ROW	H. BAIRD Princeton,N.J.	PDP-8	ASSEMBLER	1	4
7	GOMOKU	H. SAAL Palo Alto, CA.	IBM 370/158	APL	1-5	34
9	SHEIN	S. WANG Guelph, Ont.	IBM 370/155	FORTRAN	1-3	45
10	DEMENTA	P. FLEISCHER Guelph, Ont.	IBM 370/155	PL/I	1	150
11	FRED	J. STURDY Toronto, Ont.	IBM 370/155	PL/I	1	200

A Sample Game

The final position in one of the games between the winning program (PLUNC) and the second ranked program (ARTHUR) is shown below. PLUNC (moving first) is marked X on the diagram and ARTHUR is marked O. ARTHUR resigned after move 51 as a loss was imminent. After move 51,

(52) N-12 (53) N-12 (forced), then

(53) produces a forced win in four by some variation of

(54) P-16 (55) M-15 (56) K-15 (57) P-15 win.

```
    A B C D E F G H J K L M N O P Q R S T
19  . . . . . . . . . . . . . . . . . . .  19
18  . . . . . . . . . . . . . . . . . . .  18
17  . . . . . . . . . . . . . . . . . . .  17
16  . . . . . . . . . . . . . . . . . . .  16
15  . . . . . . . . . . X . X . . . . . .  15
14  . . . . . . . . . . O . X . X . . . .  14
13  . . . . . . . . . O X X X X O . . . .  13
12  . . . . . . . . . . X O   O O X . . .  12
11  . . . . . . . . . O X O X . O . . . .  11
10  . . . . . . . O . X O X O O O O X . .  10
 9  . . . . . . . . X O X X O . X . O . .   9
 8  . . . . . . . . X X O O X . . . . X .   8
 7  . . . . . . . . O . X X O . . . . . .   7
 6  . . . . . . . . . . . . O O . . . . .   6
 5  . . . . . . . . . . . . . . O . . . .   5
 4  . . . . . . . . . . . . . . . X . . .   4
 3  . . . . . . . . . . . . . . . . . . .   3
 2  . . . . . . . . . . . . . . . . . . .   2
 1  . . . . . . . . . . . . . . . . . . .   1
    A B C D E F G H J K L M N O P Q R S T
```

The final board position was reached by:

```
 (1) L-11   (2) M-11   (3) M-10   (4) N-9
 (5) L-9    (6) N-10   (7) L-12   (8) L-8
 (9) N-11  (10) O-12  (11) L-13  (12) L-10
(13) L-15  (14) L-14  (15) M-9   (16) O-10
(17) O-13  (18) M-8   (19) K-8   (20) J-7
(21) L-7   (22) K-9   (23) J-8   (24) N-6
(25) M-7   (26) N-7   (27) N-8   (28) P-11
(29) Q-12  (30) Q-10  (31) N-13  (32) R-9
(33) S-8   (34) P-13  (35) P-9   (36) O-6
(37) J-9   (38) P-10  (39) R-10  (40) P-5
(41) Q-4   (42) H-10  (43) M-13  (44) K-13
(45) N-14  (46) P-12  (47) P-14  (48) K-11
(49) K-10  (50) M-12  (51) N-15
```

COMPUTER CHESS REPORT

T.A. Marsland
University of Alberta

There has been much activity in the computer-chess field since the last CSCSI newsletter. At the second C-C workshop in Edmonton during June 1976, for instance, five different programs (CHAOS, CHUTE, COKO, TREEFROG, and WITA) were for the first time installed on a single machine (an Amdahl) and various tests performed in an attempt to measure the computational efficiency of the algorithms used. Unfortunately an air traffic control strike prevented the participation of some of the authors. Three additional programs (BELLE, E. SHRDLU and OSTRICH) were implemented on other equipment, and something akin to a normal tournament took place. Where possible the programs were constrained to complete their moves with a specified amount of the CPU resource (rather than the customary elapsed time clock), in an attempt to compensate for the intrinsic speed differences between the machines. In addition, instruction traces were taken for the five programs on the Amdahl, and a condensed report of the data gathered is available from the author of this note.

A Third computer-chess workshop is planned in conjunction with the CIPS Conference in Edmonton, 23-25 May 1978. Local arrangements are being handled by Dr. Steven Soule at the Univ. of Calgary. By that time it is expected that DATAPAC will be offering communication service to most major Canadian Universities.

In early August the second World Computer Chess Championship was held in Toronto, at the IFIP77 Conference. Despite airline disruptions, sixteen programs participated, including KAISSA the previous champion from Russia, and the current North American champion, CHESS, which recently won a place in the Minnesota closed chess tournament and holds its own against Masters in 5-minute games. Perhaps as interesting as the actual games was the audience of more than 300 people for each round. Among the chess notables were M. Botvinnik, Edward Lasker, and author D. Levy who directed. Our local entry, WITA, finished equal sixth, and was the highest placed Canadian program. The calibre of the games was varied, but significant was that BELLE demonstrated that it could hold a draw on the "rook side" of the "won" K&Q v K&R ending, even against Masters.

Crosstable World Computer Chess Championship.

1. CHESS	USA	+13	+6	+2	+4	4.0
2. DUCHESS	USA	+3	+12	-1	+6	3.0
3. KAISSA	USSR	-2	+16	+5	+7	3.0
4. BELLE	USA	=7	+11	+9	-1	2.5
5. CHAOS	USA	+15	=9	-3	+11	2.5
6. MASTER	UK	+14	-1	+10	-2	2.0
7. BLACK KN.	USA	=4	=10	+12	-3	2.0

8.	WITA	CDN	-12	+13	=15	-9	2.0
9.	DARK HORSE	SWE	+16	=5	-4	=8	2.0
10.	ELSA	GER	=1	=7	-6	+15	2.0
11.	BLITZ	USA	=10	-4	+14	-5	1.5
12.	OSTRICH	CDN	+8	-2	-7	-13	1.5
13.	BCP	ENG	-1	-8	+16	-12	1.5
14.	CHUTE	CDN	-6	=15	-11	+16	1.0
15.	BS'66'76	HOL	-5	=14	=8	-10	1.0
16.	TELL	SWIT	-9	-3	-13	-14	0.0

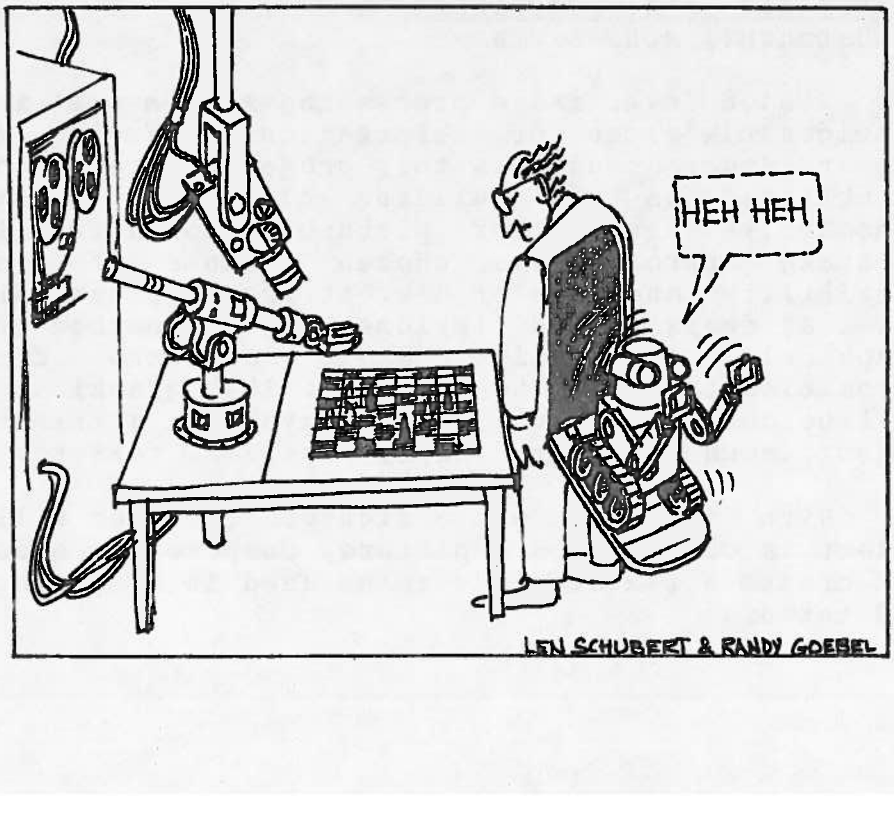
Even more recently the 9th North American championship took place in Seattle. Again Canadian programs (OSTRICH, CHUTE, WITA, and BRUTE FORCE) were well represented, but not too sucessfull. The results are summarized in the cross table, but these cannot show the heartbreaking failures which arise in public exhibitions of this type. For example, OSTRICH lost its game to 8080 when it 'died' in a print loop trying to spell MATE! Similarly, WITA lost two won games when its new Pawn-promotion-sacrifice mechanism collapsed.

The climax of the tournament was a fine drawn game between CHESS and DUCHESS, who consequently tied for first place. At the conference itself three papers on computer chess were delivered, and were followed by a panel discussion. Unhappily none of the questions was related to the AI aspects of the work. Information such as "what is the rating improvement of your 'ply 6' program over your 'ply 5' program" was sought, but there was little enquiry about the relative effectiveness or generality of the methods used, or analysis of the theoretical problems posed. In fact there was little to suggest that computer chess has anything to do with AI. All the sucessful programs unashamedly use brute force methods, and almost without exception their authors are professional systems programmers. Those programs with avowed AI content, TYRO with its table driven pattern matching scheme, and WITA with its highly selective search (average of 270 positions per move), continue to be plagued by design errors. next year.

Crosstable: 8th North American Computer Chess Championship.

1.	CHESS	CDC176	+8	+4	+3	=2	3.5	Northwestern
2.	DUCHESS	370/168	+5	+6	+7	=1	3.5	Duke
3.	CHAOS	Amdahl	+9	+12	-1	+5	3.0	Michigan
4.	XENARBOR	370/158	+10	-1	+6	=8	2.5	CDC
5.	BLITZ	Sigma 9	-2	+8	+12	-3	2.0	Mississippi
6.	BLACK KN.	U1110	+11	-2	-4	+10	2.0	Univac
7.	OSTRICH	Nova	-12	+11	-2	+9	2.0	McGill
8.	CHUTE	Amdahl	-1	-5	+9	=4	1.5	Toronto
9.	TYRO	PDP10	-3	+10	-8	-7	1.0	U.S.C.
10.	WITA	Amdahl	-4	-9	+11	-6	1.0	Alberta
11.	BRUTE	370/168	-6	-7	-10	+12	1.0	Manitoba
12.	8080	Intel	+7	-3	-5	-11	1.0	Proc. Tech.

As a help to improved communication regarding computer chess matters, a data base containing information on this subject is being built here at the University of Alberta. A technical report entitled "A comprehensive list of computer chess literature", TR77-4 by T.A. Marsland is available. A selected annotated bibliography including non-English language papers should be ready during next year.



RESEARCH SUMMARIES

McGill University
Montreal, P.Q.

Computer Graphics and Image Processing Group
Department of Electrical Engineering

Image Interpretation Using a World Model
S. Haboucha, D. Kashtan, M.D. Levine, S. Shaheen

A relational database package has been developed and implemented on the PDP/15 system. The package uses a relational algebra sublanguage for either interactive interrogation of the database by a casual user or to be incorporated by other high level programs for the general purpose image understanding system.

The relational database is designed to be used as an associative memory for scene analysis or synthesis. It represents the world model or long term memory of the image understanding system.

Research is presently underway to develop a high level image analyzer in the form of a production system which will be able to direct and control the low level processing stages, make use of the already known facts (results) about the scene to derive new facts, and finally, to produce a symbolic description of the input pictorial data.

An Associative Database for Creating World Models for Image Processing
S. Haboucha, M.D. Levine

A high level image processing system must incorporate a considerable amount of information regarding the problem domain under study. In this project, a relational database is utilized as the building block for constructing an associative memory for picture information. A relational database approach was chosen because of its inherent flexibility and ease of use. At present, the main effort is aimed at devising and implementing a method of entering graphical information about objects, directly and automatically, into the database. This graphical information will be composed mainly of the physical attributes of the object, such as color, shape, size, and texture.

With the aid of a lightpen the user will be able to select an object from a picture, compute its size and color, and create a feature file to be used in computing the shape and texture.

Non-Purposive Picture Segmentation
J. Leemet and M.D. Levine

This research is concerned with the low level processing stage of a computer vision system. At this level, sensory information is employed to partition the picture into regions containing pixels exhibiting similar properties. The input measurements are the red, green, and blue color components obtained by three successive scans of the image. Provision is also made for the incorporation of texture information. Each of these primary features is employed to create a pyramid structure which is then used as input to a shared nearest neighbor clustering processs. Edge data is used to provide good global starting points for the grouping of the pixels, as well as to select prospective candidates for labelling. The method may be looked upon as a parallel generation of skeletons followed by a propagation process expanding the skeletons in all directions.

Experiments With Low Level Picture Segmentation
M.D. Levine and A. Nazif

Experiments are being conducted on the Low level processing stage of an image understanding system. The objectives of these experiments are to:

1. Select suitable features for color information representation. Three color coordinate representations are compared in gradient and edge calculation, as well as the resulting picture segmentation. The representations are the Red, Green, and Blue color scans of a picture; the intensity, hue, and saturation color coordinate system; and the TV color coordinate system (YIQ).

2. Find suitable weights (or weighting functions) for the features used in the gradient calculation and the picture segmentation.

3. Optimize various systems parameters. This includes setting the threshold for the edge calculations and the clustering parameters for picture segmentation.

Intermediate Level Processing for an Image Interpretation System
M.D. Levine and D. Ting

Research is being done in the area of integrated scene analysis systems. Specifically, the design of a knowledge based image recognition system is being implemented. One level of processing which is of concern is the Intermediate Level which interfaces the knowledge concerning the scene with the data in the actual image being analyzed. The function of this level is to perform an interpretation of the image, given certain a priori knowledge about it. This is accomplished by a combination of techniques involving both heuristic graph search and dynamic programming.

Monocular Depth Perception
M.D. Levine and D. Rosenberg

This project deals with monocular depth perception by computer, utilizing contextual depth cues. A segmented picture with features calculated for each region is used as the input data. Depth labels for each region are proposed based on an occlusion analysis of the neighboring regions. The possible labels are then resolved using a relaxation algorithm which provides a partial region ordering according to depth. In the future other depth cues will be used to complete the ordering. These will include ground plane support and texture/color gradient.

Cooperative and Competitive Computation in Computer Vision Systems
S.W. Zucker

This research in low level vision has primarily been centered around the study of networks of cooperating and competing computational processes. These processes, called relaxation labelling processes, are particularly useful in reducing local ambiguities which arise during the processing of visual information. For example, when an original intensity array (i.e. picture) is interpreted into a lowlevel, symbolic vocabulary, local feature detectors do not respond only to the selected pattern feature; they also respond to various noise configurations. The relaxation labelling processes reduce these kinds of local ambiguities by making use of constraint or compatibility relationships between pairs of neighbouring response interpretations (or labels). They work in a parallel and iterative manner, thus allowing local certainties to exert a global influence.

Formally, relaxation labelling is a class of computational processes which manipulate labels on graphs. The underlying graph structure denotes both the picture parts or abstract objects to be labelled and the neighbour relations over these objects. If the relaxation process operates discretely, then it discards labels which are inconsistent with the label sets attached to neighboring nodes. If it operates probabilistically, then it updates a measure of likelihood attached to each label. The initial probabilities are obtained, e.g., on the basis of the feature dector responses. Mathematical models to study the convergence, speed, and stability of these processes have been developed. Also, their empirical behavior has been studied through a number of applications, including the enhancing of both lines and curves in LANDSAT and SKYLAB satellite imagery, and the description of dot cluster patterns.

Labelling and Grouping Dot Patterns

Y. Leclerc and S.W. Zucker

After low-level structures have been labelled explicitly, they must be grouped into more global patterns. This project is designed to study the grouping processes at work cn dot patterns that make their cluster structure so apparent. It offers a new approach to non-parametric cluster analysis that is based on the shape of dot clusters. It also provides examples of the complex feedback paths necessary within more powerful vision systems so that they can cope with ambiguity and noise.

Classification of Osteogenesis Imperfecta Tissue Cultures Through Interactive Texture Analysis

G. Brighten and M.D. Levine

Automated Study of Blood Cell Geometry from Photomicrographs

M.D. Levine and J. Russ

Describing the Line and Edge Structure in Medical Imagery

J. Mohammed and S.W. Zucker

An Automatic Picture Processing Method for Tracking and Quantifying the Dynamics of Blood Cell Movement

M.D. Levine and Y. Youssef

Animation of Computer Generated Color Images

M.W. Koo and A.S. Malowany

Computer Generation of Realistic Pictures

M. Aoki and M.D. Levine

THE McGill Image Processing System (MIPS)

M.D. Levine

The McGill Image Processing System, MIPS, is a modular, interactive set of image processing program modules, each of which is totally independent. These modules are linked by the MIPS monitor, an interactive program which activates modules as they are selected by the user. Interaction with the image processing system is almost entirely graphical. Lists of options are displayed in the form of a menu. A joystick, with crosshairs, is used to select the desired options.

MIPS runs under a real-time multi-tasking operating system written at McGill for its PDP-15 processor. This system supports multiple user program development as well as multiple user access to the image processing system. The operating system supports "sub-tasking", a feature which allows a running program (also called a task) to request the execution of another task on its behalf and under its control. The requesting task may then continue executing in parallel with the "sub-task" or may, at any time, wait for the "sub-task" to complete. Using this facility, the MIPS monitor assesses which modules are in the system (including those belonging to the user), displays a menu of the modules on a graphics terminal, reads the joystick selection and instructs the operating system to begin execution of the selected module on behalf of the monitor. Each user may include his own image processing programs in his menu by merely having the program name (task name) begin with the letter "*". Thus he may have a selection of modules suited towards his particular application.

The image processing system also includes a standard filing system for storing pictures on disk. Picture names are 5 characters long with up to 40 characters appended to the basic name to describe various picture attributes (e.g. whether the picture contains intensity, red, green, blue, hue or saturation information). The first record of the picture contains a complete description of picture parameters and is followed by the actual pixel values for the picture.

MIPS contains modules for digitization, display, compression, filtering, histogram analysis, texture and feature analysis, and other general functions. The various levels in the computer vision system under design are also modules on their own. The system is continuously being augmented; current development is concerned with two-dimensional image filtering and a simulated array processing subsystem.

Universite de Montreal
Montreal, P.Q.

A New Image Processing System

Bill Armstrong and Jan Gecsei
Department d'informatique et de recherche operationnelle
Universite de Montreal

An interactive image processing system is currently under development at the Universite de Montreal. It will make use of two graphic display units located in the Pattern Recognition Laboratory: a Norpak 400B for generating a 256 x 256-pixel color display, and a Princeton 801 for generating gray-scale images and line drawings. The user will control the whole system using the Princeton 801 keyboard and cursor as inputs.

For high-speed image creation, these terminals will be connected via high-speed links to an Interdata 4 mini computer in the Applied Computer Science Laboratory. Tapes with image data, for example from LANDSAT, can be consulted and their processing directed from the terminals. The Princeton 801 and the Interdata 4 are also linked to the two Cyber 173 computers in the U. de M. computing centre.

There will be two levels of processing: low-level extraction of primitives in the Interdata 4 with the aid of a special purpose hardware processor, and higher-level processing on the two Cyber 173's. Low-level operations include classifications of pixels or groups of neighboring pixels. Although fairly simple, these operations would be quite time-consuming if done for a large image on one of the Cyber 173's. The objective of the special processor is to perform at high speed (and low cost) certain operations which are expected to be particularly useful for general pattern classification. The nature of these operations will now be briefly outlined.

A pixel usually contains information about the intensity I(i) of reflected light in each of k spectral bands. Sometimes processing requires computation of other values V(j) such as logarithms or quotients of intensities. Each such value will usually be discretized to 256 levels or less. A class, e.g. water in a LANDSAT image, is characterized by a class-conditional probability function defined on the multidimensional space of values. The resulting decision for the various classes can, however, also be determined in a straightforward manner by examining a set of thresholds on the numerical values. More generally the above union of two rectangles becomes a set formed by unions and intersecticns starting from half-spaces. In general this is equivalent to evaluating some boolean function of binary features derived from the numerical values I(i),V(j). The simplicity of the shape of a decision region is reflected to some degree in the simplicity of the boolean function above, which ws why the special processor has been designed to compute simple boolean functions very quickly.

The processor can compute in one cycle (roughly 100 ns) a binary feature, such as I(i)>t, or a simple boolean function of 32 variables based on data in the 512 x 32 bit RAM data memory. In some cases, it is more efficient to compute several boolean features of the numerical values simultaneously via table look up in the Interdata, however the evaluation of boolean functions of many variables requires use of the special processor in order to gain speed.

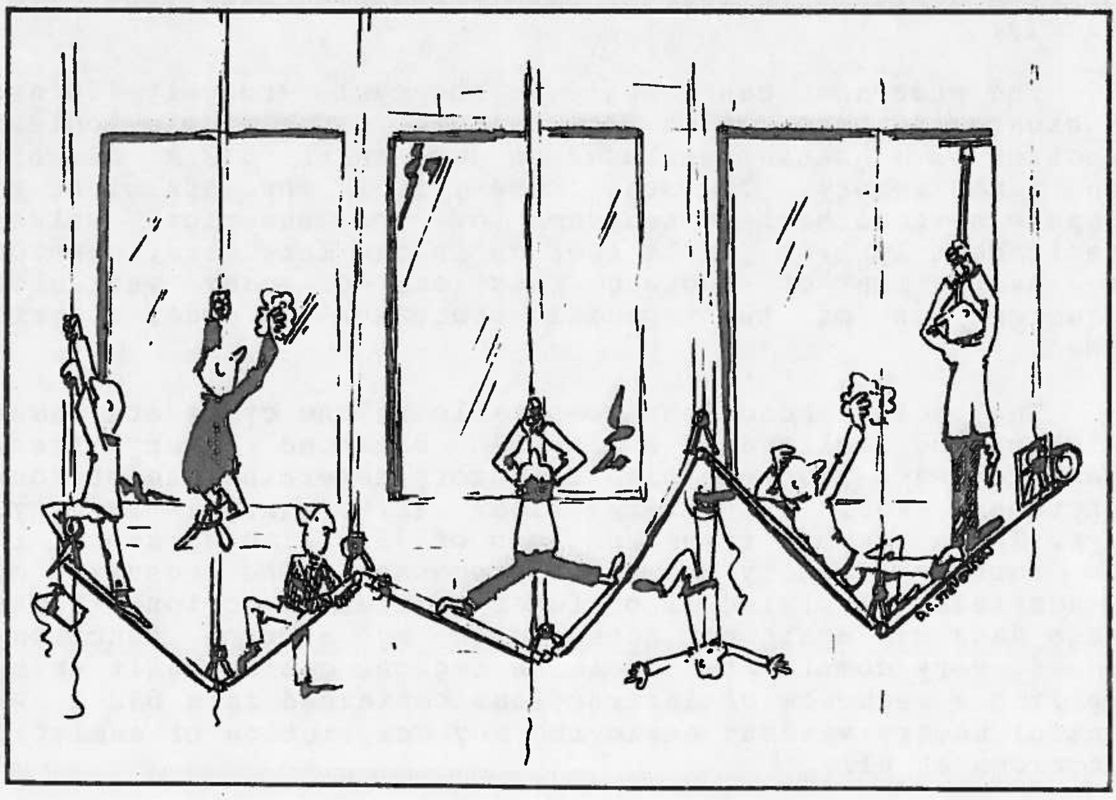
The boolean functions computable in one cycle are those which can be realized by a 31-node balanced binary tree, where for each node two bits of memory determine one of four functions: AND, OR, LEFT, RIGHT (i.e. g(x,y) = xy, x+y, x,y). There are 512 tree functions of 32 variables stored in the function memory of the processor. The results of sequentially applying 32 or fewer of these functions to the image data can again be acted upon by a tree function. Indeed very complicated decision regions can be built up by applying a sequence of instructions contained in a 512 x 42 control memory without employing any computation of analytic functions at all.

In order to suboptimize the number of tree functions required to implement complicated decision regions, which is very important in attaining high speed of classification, an adaptive procedure can be used. This is particularly useful when only a limited number of labelled samples of the classes is available, for then the synthesis problem of the functions is analytically quite hard.

Since the decision regions can be set up to reflect accurately any Bayesian or maximum likelihood decision rule, without the hypothesis of normality etc. of the distributions, a higher accuracy of classification can be obtained in principle than if, say, linear or quadratic discriminants are used. We shall be investigating the time required to obtain a certain accuracy when the processor is used, and comparing it to the time for the standard methods of classification.

There are connections between this work and AI research. Consider for example the use of production systems, where an action is invoked only if a certain condition is satisfied. In order to determine whether conditions are satisfied, a processor for evaluating boolean functions could prove useful for accelerating the process of testing conditions, particularly if the conditions involve many variables.

More documentation on the system will be sent to those requesting it.



CONCURRENT GOAL SATISFACTION

University of British Columbia
Department of Computer Science
Vancouver, B.C.

An Approach To Deductive Question-Answering

Raymond Reiter

This paper is concerned with a variety of issues which arise in deductive question-answering. Its principal concern is with the design of a retrieval system which combines current techniques for query evaluation on relational data bases with a deductive component in such a way that the interface between the two is both clean and natural. More specifically, a suitably designed theorem prover "sweeps through" the intensional data base (i.e. the set of general facts about the domain of interest), extracting all information relevant to a given query. The end result of this sweep is a set of queries, each of which is then evaluated over the extensional data base (i.e the set of specific facts). The union of the answers returned from each of these queries is the set of answers to the original query. Since the theorem prover never accesses the extensional data base this approach appears to be feasible for data bases with very large extensions and comparatively small intensions. For a suitably defined class of data bases, we prove the completeness of this approach i.e. that all answers to a given query will be returned. This result holds in the presence of equality, and for incompletely specified worlds.

A feature of many deductive question-answering systems is that they function in closed world mode which means, roughly speaking, that what cannot be proved is taken to be false. Such systems rely on different proof techniques from those of first order logic. We provide a formal theory for closed world query evaluation which justifies these proof techniques, and which guarantees that all closed world answers will be returned. In addition, we point out that the closed world assumption can lead to inconsistent data bases, and we characterize a natural class of data bases for which such inconsistencies cannot arise.

This paper also addresses some issues on how best to structure a data base. We adopt, as a structuring principle, the elimination of infinite deduction paths. One way of doing so is to treat definitions in a special way. Another involves a kind of intension-extension trade-off. By "filling in" the extensions of certain suitably chosen relations, one can guarantee only finite deductions.

Part of this paper is concerned with issues of integrity. It turns out that, because of the particular representation and query language which we use, certain kinds of integrity constraints can be enforced for queries and data base updates.

Finally, we propose an approach for compiling the intensional data base. Roughly speaking, this involves determining, at data base creation time, proofs for all of the relations in the data base. Subsequently, at query evaluation time, these proofs are appropriately combined to yield all proofs for that query.

Available as a Bolt, Beranek & Newman technical report, Cambridge, Mass. Sept. 1977, 161 pages.

WHISPER: A Computer Implementation Using Analogues in Reasoning

Brian V. Funt
Artificial Intelligence Laboratory
Stanford University

Technical Report 76-9, based on the author's thesis of the same title, submitted on March 31, 1976, Department of Computer Science, U.B.C.

The use of an analogue as an aid to a problem solving program is investigated. A working system, the advantages of the analogue it uses, the mechanisms required, and the interaction with other forms of knowledge are described.

The program WHISPER, uses a diagram together with procedures for modifying it, as an analogue of a situation involving a stack of arbitrarily shaped rigid bodies. It determines a stack's stability and predicts the motions of any unstable object by examining the situation's diagram. The analogue is particularly valuable in detecting discontinuities in an object's motion. For example, collisions with other objects or cliffs an object might slide over can be 'seen' in the diagram rather than having to be inferred from a description of the situation.

WHISPER uses a simulated parallel processing 'retina' to look at the diagram which is encoded in a two-dimensional array. It consists of a fixed number of processors operating in parallel and communicating only with their immediate neighbours. WHISPER'S retina resembles the human retina in some respects. Its resolution decreases away from its center. It can be moved to fixate on different sections of a diagram.

A set of domain independent features are extracted from WHISPER'S diagrams by procedures, called perceptual primitives, which execute on the parallel processing retina. Example features are: symmetry of an object, similarity of two objects, and contacts of an object with other objects. In addition to these primitives, the retina can be used to 'visualize' the rotation of an object without having to move it directly in the diagram.

The advantages of analogues are classified in terms of two categories according to whether a correspondence exists between the behaviour of the analogue and the behaviour of the external situation, or whether a correspondence exists between the static configurations of the analogue and those of the external situation. Some reasons for the effectiveness of analogues are presented.

A Knowledge Identification Phase Of Natural language Analysis

Roger Alexander Browse
MSc Thesis, U.B.C., January 1977

Case organization of verbs has provided a powerful mechanism for natural language analysis systems. However, only simple semantic-marker-like information has been used to determine the acceptibility of lexical elements as case-role fillers. Actually, this ability is influenced by more intricate relations among words. In addition, a case-based view of semantic knowledge often leads to the separate specification of each shade of meaning of a verb.

These two problems are addressed in this thesis. A case-like organization of semantic knowledge which includes a network of relations among lexical elements is presented. Any piece of information contained in the system may be used as a case-frame specification, or it could be used as information which determines case-role fulfillment. Rules for the use of this information have been designed to permit a single case-frame to recognize many shades of meaning of a verb, even to the point of accepting metaphoric language use.

The network of relations is hierarchically organized, and knowledge is retained at many levels of generalization. Along with the existence of case-organization in the network, these multiple levels provide some control over the traversal of the network.

A small implementation is provided to demonstrate the use of a variety of strategies for fitting case-frames to input. The model is intended as a bottom-up component for the identification of those pieces of information which may be relevant to a given input.

Vision Research Strategy: Black Magic, Metaphors, Mechanisms, Miniworlds and Maps

Alan K. Mackworth

Machine vision will advance substantially only if it continues to develop a coherent theory. As with all fledgling sciences, the framework for such a paragidm has emerged as a result of restricting the scope of attention to limited but non-sterile domains that serve the current needs of the theory. An example of such a domain is the class of freehand sketches. These occupy a position in vision analogous to that of speech in that they are designed for person-to-person communication and thereby have a rich, conventional semantics which can be exploited. The goals of a project to understand sketches are given. A very brief description of a program, MAPSEE, that interprets sketch maps illustrates the argument. A conservative partial segmentation yields a variety of cues which invoke models that interact according to a uniform control structure: a network consistency algorithm. The necessary deficiencies of the segmentation, their effect on the interpretation and using the interpretation to refine the segmentation are all mentioned. This example is used to focus discussion of a variety of vision issues such as the chicken-and-egg problem, the power of descriptive models and their corresponding weaknesses, the incremental nature of constraint methods, cue-model hierarchies, the modularity and generality problems and procedural adequacy. Finally a cyclic theory of perception is used to characterize a variety of vision programs.

In Advance Papers of Workshop on Computer Vision Systems, Amherst, Mass., June 1-3, 1977. (to be published by Academic Press). Paper available from author.

<u>On Reading Sketch Maps</u>
Alan K. Mackworth
Technical Report 77-2, May 1977

A computer program, named MAPSEE, for interpreting maps sketched freehand on a graphical data tablet is described. The emphasis in the program is on discovering cues that invoke descriptive models which capture the requisite cartographic and geographic knowledge. A model interprets ambiguously the local environment of a cue. By resolving these interpretations using a network consistency algorithm for n-ary relations, MAPSEE achieves an interpretation of the map. It is demonstrated that this approach can be made viable even though the map cannot initially be properly segmented. A thoroughly conservative, initial, partial segmentation is described. The effects of its necessary deficiencies on the interpretation process are shown. The ways in which the interpretation can refine the segmentation are indicated.

(In <u>Proc</u>. <u>Fifth Int</u>. <u>Joint Conf</u>. <u>on Artificial Intelligence</u>, M.I.T., Cambridge, MA August 22-25, 1977)

<u>An Approach to the Organization of Knowledge for the Modelling of Conversation</u>
Gordon McCalla, Ph.D. Thesis (the author is presently at the Dept. of Computer Science, University of Toronto).

A description of an approach to the modelling of conversation is presented. It is suggested that to have any hope of succeeding at this endeavour, the problem must be tackled principally as a problem in pragmatics rather than as one in language analysis alone. Several pragmatic aspects of conversation are delineated and it is shown that the attempt to account for them raises a number of general issues in the representation of knowledge.

A scheme for resolving some of these issues is constructed and given computational description as a set of (non-implemented) LISP-based control structures called |LISP. Central to this scheme are several different types of objects that encode knowledge and communicate this knowledge by passing messages. One particular kind of object, the pattern expression (|PEXPR), turns out to be the most versatile. |PEXPRs can encode an arbitrary amount of procedural or declarative information; are capable, as a by-product of their message passing behaviour, of providing both a context for future processing decisions and a record of past processing decisons; and make contributions to the resolution of several artificial intelligence problems.

A model of conversation is then proposed and some examples of typical conversations that might occur in the general context of attending a symphony concert are detailed in |LISP. It is suggested that conversation is goal oriented behaviour; and, in fact, the model is presented in terms of level of goal: from higher level non-linguistic goals through scripts directing both sides of a conversation, speech acts guiding one conversant's actions, and, finally, language level goals providing a basic parsing component for the model. In addition, a place is delineated for belief models of the conversants, necessary if utterances are to be properly understood or produced. The embedding of this kind of language model in a |LISP have yeilded a rich pragmatic environment for analyzing conversation.

University of Edinburgh
Edinburgh, Scotland

Mechanizing Structural Induction
Raymond Aubin
PhD. Thesis, 1976

This thesis proposes improved methods for the automatic generation of proofs by structural induction in a formal system. The main application considered is proving properties of programs. The theorem-proving divides into two parts: (1) a formal system, and (2) proof generating methods.

A formal system is presented which allows for a typed language; thus, abstract data types can be naturally defined in it. Its main feature is a general structural induction rule using a lexicographic ordering based on the substructure ordering induced by type definitions.

The proof generating system is carefully introduced in order to convince of its consistency. It is meant to bring solutions to three problems. Firstly, it offers a method for generalizing only certain occurrences of a term in a theorem; this is achieved by associating generalization with the selection of induction variables. Secondly, it treats another generalization problem: that of terms occurring in the positions of arguments which vary within function definitions, besides recursion controlling arguments. The method is called indirect generalization, since it uses specialization as a means of attaining generalization. Thirdly, it presents a sound strategy for using the general induction tule which takes into account all induction subgoals, and for each of them, all induction hypotheses. Only then are the hypotheses retained and instantiated, or rejected altogether, according to their potential usefulness. The system also includes a search mechanism for counter-examples to conjectures, and a fast simplification algorithm.

Massachusetts Institute of Technology
Artificial Intelligence Laboratory
Cambridge, Masschusetts

COMEX: A Support System for a Commodities Expert
James L. Stansfield

The intelligent support system project is developing an assistant for a commodities expert (COMEX). Large amounts of qualitative and quantitative information from many sources need to be managed. I describe COMEX-0, a prototype system written in FRL, a frame-based language (Goldstein & Roberts. 1977). COMEX-0 builds frame structures by conversing with a user. It has a complaint handling system, frame-structure matching and simple reasoning. New event types are represented in terms of defined ones using INSTANTIATION and AGGREGATION. These twc methods combine with frame inheritance and constraints to make up a general event representation mechanism. This leads to the idea of generic patterns of relationships between frames. I call these CLUSTERS. I discuss a problem that arises concerning constraints in clusters which I call the generic constraint problem. I conclude with a hypothetical scenario and problem statement for future work on qualitative reasoning within COMEX.

Descriptive terms: qualitative reasoning, frames, intelligent support systems, clusters generic constraint problem, event representation.

State University College
Potsdam, N.Y.

Problems in Theory Formation
Alois Glanc

We study practical algorithms for special cases of the following general logical problems that are essential in Theory Formation.

Given a set of models, find a corresponding suitable axiomatic system.

Given a formula, find a suitable model of the formula.

The following techniques are employed: enumeration using formula and model schemas, modification of similar models and formulas, creation of "canonical" formulas and models.

Queen's University
Kingston, Ont.

Automatic Rib Detection in Chest Radiographs
C. M. Brace

IRPIL - An Interactive Pattern Recognition Laboratory for Teaching and Research
S. Christodoulakis

Computer Perception and Description of Curved 3-D Surfaces
K. Martin

Automated Texture Analysis of Chest X-ray Lung Fields
C. M. Haig.

If you would like copies of the above reports, please contact Jeffrey H. Kulick, Asst. Professor, Dept. of Computing & Information Science, Queen's University, Kingston, Ont. K7L 3N6.

National Research Council of Canada
Ottawa, Canada

Computer-Aided Learning -- A Cooperative Research Project Of The National Research Council of Canada
J.M. Brahan

In 1969, following a preliminary feasibility study, the National Research Council (NRC) commenced a joint research and development program on Computer-Aided Learning (CAL) in cooperation with Canadian educational organizations and industry. The primary goal of the program is to ensure the availability, by 1980, of a viable and cost-effective computer-aided learning system for Canadian users. Within the framework of this principal objective, a number of sub-objectives have been identified, namely, to ensure that CAL systems developed for use in Canada are: acceptable to Canadian educational authorities and users; manufactured in Canada with maximum feasible Canadian content; economical to manufacture and use; assured of a long, useful life through develcpment and application of essential technological standards; and sufficiently advanced by 1980 that NRC contributions to their further development will be minimized.

The program is organized to bring together the educator, the research worker, the systems designer and the manufacturer, permitting efforts to be appropriately shared in accordance with expertise. Educators develop course materials and evaluate the system in a variety of applications while technologists at NRC and in industry concentrate on system hardware and software development. It is important to note that the role of NRC in the program does not include the development of curriculum content. It consists of developing the technical facilities necessary for their preparation by the educator. The NRC participation in the program consists of several activities: the operation of a CAL research network; development of specialized terminal equipment; development and implementation of CAL programming languages, software and technological standards; development of techniques for automatic classification and retrieval of reference materials; and the support of industry in the development of a CAL competence.

Le Natal-74. Definition et Developpement d'un Langage de Programmation EAO pour Repondre Aux Besoins Nationaux

J.W. Brahan, B.A. Colpitts, J.R. Goguen,
A.M. Hlady, R.A. Orchard
Section de recherches informatiques

La mise au point de cours dont le contenu permet de bien realiser le potentiel de l'ordinateur en tant que moyen d'arriver a l'enseignement individualise exige un tres grand effort. Pour justifier cet effort, le contenu doit etre accessible a un tres grand nombre de personnes. Toutefois, un obstacle majeur apparait du fait des differents langages de programmation utilises pour preparer les cours. En vue de surmonter l'obstacle le Comite associe de technologie pedagogique du Conseil national de recherches a constitue un groupe de travail dont les membres participent a des projets d'enseignement a l'aide d'ordinateurs (EAO) au Canada afin de definir un langage de programmation EAO satisfaisant les besoins des utilisateurs. Le rapport de ce groupe a pris la forme d'une specification fonctionnelle qui a ete publiee en 1972. Cette specification a forme la base d'un contrat avec la compagnie IBM Canada Ltd. pour une definition detaillee d'un langage. Le travail, dans le cadre de ce contrat, a ete termine en 1974 et des versions en anglais et en francais du Manuel de specification et du Guide de l'auteur pour le langage NATAL-74 ont ete publiees. Les points caracteristiques du langage comprennent de vastes possibilites pour l'affichage de l'information grace a des dispositifs varies et a de nombreuses formes d'entree au choix de l'utilisateur. Cet ensemble s'appuie sur des possibilites de calcul et de controle fournissant a l'auteur de cours un outil souple d'emploi pour creer un milieu d'enseignement individualise.

Le NATAL-74 etant une partie du projet EAO du Conseil national de recherches, il est mis en oeuvre sur un ordinateur DECsystem-10, ce qui fait que le langage est disponible pour les membres du reseau et qu'il leur permet de participer aux essais et a la revision. Les points interessants du langage NATAL-74, sa mise en oeuvre initiale et ses applications preliminaires seront discutes dans cette communication en meme temps que des plans pour le transfert de la mise en oeuvre sur un deuxieme systeme d'ordinateur.

Natal-74 -- Definition & Development of a Course Authoring Language to Meet National Requirements

Development of the course materials required to realize the potential of the computer as a means of providing individualized instruction demands a major effort. To justify this effort, the materials must be available to a wide audience. However, a major barrier to widespread use results from the variety of programming languages used in their preparation. In a step towards overcoming this language barrier, the National Research Council's Associate Committee on Instructional Technology convened a working panel, whose members came from computer-aided learning (CAL) projects across Canada, to define the requirements of a CAL programming language to meet user needs. The report of this panel took the form of a functional specification which was published in 1972. This specification formed the basis of a contract with IBM Canada Limited, for a detailed language definition. Work under this contract was completed in 1974 and English and French versions of an Author Guide and Specification Manual for NATAL-74 have been published. Features of the language include the provision of extensive facilities for display of information on a variety of devices and for acceptance of user input in many forms. This is supported by computational and control capabilities to provide the course author with a versatile tool for creating an adaptive teaching environment.

As part of the National Research Council's CAL project, NATAL-74 is being implemented on a DECsystem-10 computer, making the language available to network members and permitting them to participate in the process of testing and revison.

Short summary of Picture processing and Pattern recognition activity.

T. Kasvand

Research Projects:

T.Kasvand: Human locomtion studies. Programs for analysing 35 mm film transparencies of "specially marked" subjects have been transferred to a new computer. The xy coordinates of the marks on the subject are obtained in computer readable form.

S. Hung: Image compression, based on local neighbourhood logic. Relatively simple algorithms requiring few arithmetic operations produced compression ratios of 6 to 10 without noticable degradation of gray level and colour pictures.

T. Kasvand, C. Brace: Preliminary studies ("Phase 1") of hot-spot removal from high altitude wide angle aerial photographs indicate that considerable improvement in picture quality is possible. Subjective as well as quantitative evaluations of the corrected photographs will be undertaken.

T. Kasvand: Study of the performance and alteration of character recognition masks for the Canadian Post Office. The masks need continuing updating to keep abreast with varying printing fonts.

T. Kasvand: Improvement of undersea seismic signal records. The nature of this problem is being defined.

T. Kasvand: General scene analysis. Some of the described in the reference. Reference: T. Kasvand, Some Observations on linguistics for Scene Analysis; in Data Structures, Computer Graphics and Pattern Recognition; edited by A. Klinger et al. Acad. Press 1977, p.p. 179-209.

A. Fabbri (Univ of Ottawa, Ph.D Thesis): Methods of pattern recognition and numerical classification applied to geological map data. Aims:

(A) quantitative capture and statistical analysis of spatial parameters (e.g. shapes, distributions, area/contact relationship) from geological map and rock thin or polished sections;

(B) area classification for environmental models;

(C) correlation between geological variables coded from maps and the occurrence of mineral deposits,

(D) comparisons between some hardware and software methodologies for texture analysis.

J. O'Brien (University of Ottawa, B.A. Sc. Thesis): The major effort in his thesis is directed towards nonlinear adaptive edge detection in graylevel pictures.

Visitor's Position

We have a visitor's position, supported financially from about 25% to 75% of his salary at "home establishment". This year the position was occupied by Mr. H. Tamura from the Electrotechnical Laboratory, Tokyo, Japan. His interests are in texture analysis and image data bases.

Summer Student Program

In the summer student program, a student may work on a problem which provides material for a B.A.Sc Thesis.

Courses

A course on picture processing at the University of Ottawa, Dept. of Computer Science, is planned for the Spring term (78). T. Kasvand usually gives it during the evening hours.

Stanford University
Stanford, California

The HYDROID Project
R.G. Smith

We are interested in the study of problem solving in a distributed processing environment, and in the constraints placed on a distributed processor architecture by the applications for which it is intended. For purposes of the project, "distributed processing" is taken to mean parallel processing in which the individual processes are loosely coupled.

We are presently investigating AI applications, such as heuristic search and rule-based inference, together with distributed data base applications. Alternative processor node interconnection methods are also being studied. In each case, computer simulation is the primary research tool. We use a formalism known as the CONTRACT NET [1], in which individual tasks are dealt with as contracts that exist between pairs of processor nodes.

Reference: 1. R. G. Smith, "The CONTRACT NET: A Formalism for the Control of Distributed Problem Solving," presented at the 5th IJCAI, Cambridge, Mass., August 1977 (a longer version is available from the author).

A Model for Learning Systems
R. G. Smith, T. M. Mitchell, R. A. Chestek, and B. G. Buchanan, STAN-CS-77-605 (HPP-77-14), March, 1977 (also presented at the 5th IJCAI).

A model for learning systems is presented, and representative AI, pattern recognition, and control systems are discussed in terms of its framework. The model details the functional components felt to be essential for any learning system, independent of the techniques used for its construction, and the specific environment in which it operates. These components are performance element, instance selector, critic, learning element, blackboard, and world model. Consideration of learning system design leads naturally to the concept of a layered system, each layer operating at a different level of abstraction.

Simcn Fraser University
Burnaby, B.C.

The Leonardo Project

The Leonardo project represents the combined interests of various members of Computing Science, Centre for the Performing Arts, Kinesiology, Communications Studeis (all departments in the Faculty of Interdisciplinary Studies) and Audio-Visual and their resources. It has been recently stimulated by the acquisition of an Evans and Sutherland Picture System-1. There are currently ongoing projects in acoustic composition and synthesis (Barry Truax), the animation of dance notation (Tom Calvert, Jerry Barenholtz), an interactive animation facility (Jerry Barenholtz, Doug Seeley), and the development of video synthesizers (Wayne Coss, Norman Jaffe).

A pivotal activity of the project will be the offering of three workshops on the Computer and the Artist given by Barenholtz, et al., one per semester. A national clientele will be sought for the summer '78 workshop. For further information contact the chairman of the project, Doug Seeley in Computing Science.

A Special Research Tool for Artificial Intelligence: Research Proposal
Nick Cercone

Specific proposals have been made for representing many natural language constructions for the purpose of automated language comprehension, vide Cercone [1975a, 1975b] and Cercone and Schubert [1975]. Attention has been focused on the representation of states, events, actions, logical and natural language quantifiers, expressible intentionality, modalities, adverbials, comparatives (both implicit and explicit), and the meanings of complex concepts, such as "walking", "laughing", "building", etc. The interpretation and manipulation of factual knowledge (as found in English declaratives) has been systematically formulated and an experimental computer implementation has been developed. Word sense disambiguation is handled with relative ease and declarations of the kind - "the pilot BANKED his plane over the river BANK near the BANK that be BANKS on for good BANKING service" - present little difficulty. The hierarchical recursive nature of the representation for concepts (no primitives) allows complex concepts to be handled efficiently in general English usage. Currently the experimental program can handle sentences such as "John is the perfect fat man" (lambda abstracted predication), and "Short, small Anne is eating the delicious, light, round, yellow, cake."

The following problems are still under investigation and solutions are formulating slowly: (1) Incorporate a general method for handling adverbials as functors (in the sense that Creswell [1973] uses them); (2) Formulate mechanisms for including procedural and other "non-factual" knowledge in the existing representation (earlier attempts, such as those by Winograd [1972] and Charniak [1972], worked well when the understanding endeavour was sufficiently constrained to a suitable micro-world. These two approaches have had little follow-up and are not appropriate in a more general representation. When the ramifications are determined of distinguishing node types in the semantic network, procedural and other "non-factual" knowledge can be included); (3) Formulate superimposing organisations into the semantic network to allow rapid access to contextually pertinent facts for a concept and plausible inferences (Minsky's [1974] recent work on "frame systems" is important but develops no theoretic method for determining contextual relevance among concepts); and (4) Develop accessing algorithms to allow rapid manipulation of large scale natural language databases. Whereas many good algorithms exist for the storage, retrieval, and manipulation of relatively static, special purpose, data bases, there exist few algorithms to manipulate a large scale natural language data base that allows as flexible and powerful a representation of natural language as the one developed above.

In order to investigate these problems in a meaningful yet inexpensive way, I propose tc construct a special purpose research tool using recently developed microprocessor technology. Specifically, I would like to construct a microprocessor-based microprogrammed list processing machine in which the primitive instructions carried out on the machine would correspond to list processing operations (cf. the LISP macine, Greenblatt et al., 1977). Successful computer implementations of the above research depend in part on the computing power available, in part on list processing languages (which presently are interpretive, hence slow), and in part on the degree to which the primitive machine operations facilitate encoding of natural language processing algorithms. This type of computing presently requires enormous amounts of machine resources on conventional general purpose computers which are commonly found in university computing centres. Because of today's rapid technological advances, personally tailored computing is available at low cost. A research tool for the purposes discussed above would require a minimal amount of equipment, should be able to be constructed rapidly (12-19 months), and should provide long range benefits as a research tool.

References

Cercone, N. [1975a]. "Representing Natural Language in Extended Semantic Networkds", Ph.D. Thesis, Report TR75-11, Department of Computing Science, University of Alberta, Edmonton, Alberta, Canada.

Cercone, N. [1975b]. "The Nature and Computational Use of a Meaning Representation for Word Concepts", American Journal of Computational Linguistics, Volume 2, AJCL Microfiche 34, pp 64-81, 1975. (also 13th Annual Meeting ACL Abstracts).

Cercone, N. and Schubert, L. [1975]. "Toward a State-Based Conceptual Representation", Advance Papers of IJCAI-75, Tbilisi, USSR, pp 83-90.

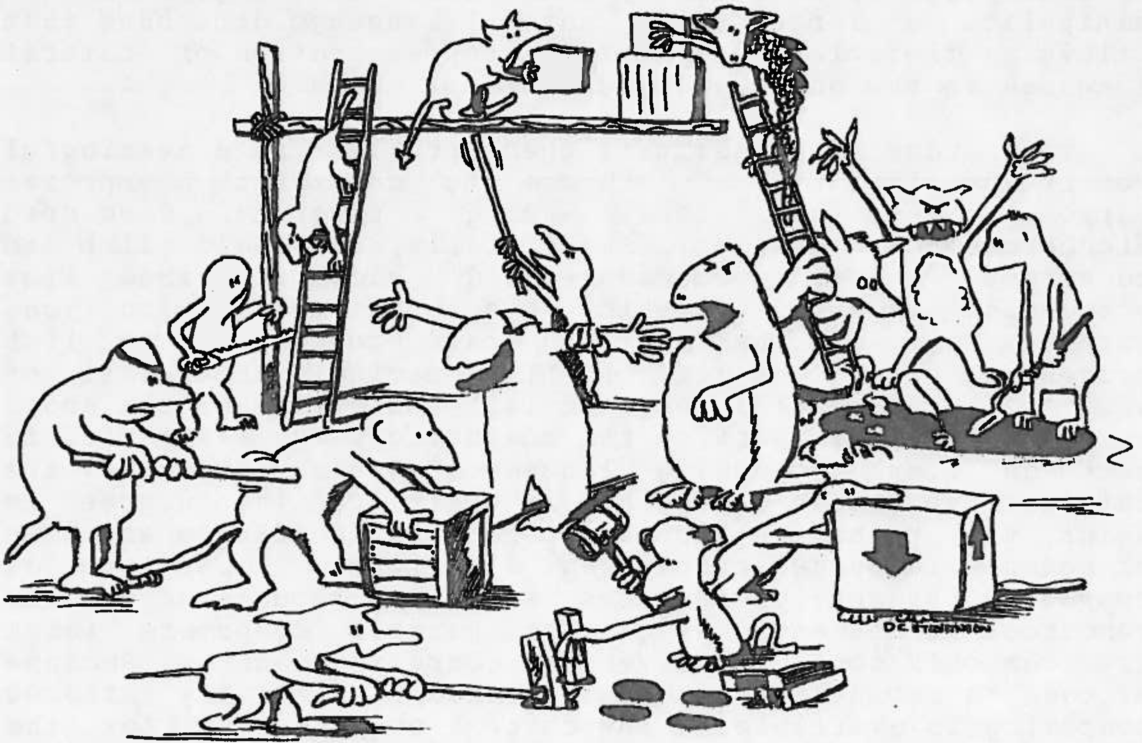
Charniak, E., [1972]. "Toward a Model of Children's Story Comprehension", AI TR-266, Ph.D. Thesis, MIT, Cambridge, Mass.

Cresswell, M., [1973]. LOGICS AND LANGUAGES, Methuen and Company Ltd., London, England.

Greenblatt, R. (1977). "The LISP Machine", AI Working Paper 81, MIT, Cambridge, Massachusetts.

Minsky, M., [1974]. "A Framework for Representing Knowledge", AI Memo 306 AI Lab., MIT, Cambridge Mass.

Winograd, T., [1972]. UNDERSTANDING NATURAL LANGUAGE, Academic Press, New York.



HETERARCHY

University of Toronto
Department of Computer Science
Toronto, Ont.

Using AI to Fill-out Your Individual Income Tax Return

Bob Kidd, Peter Schneider, and Yannis Vassiliou

This project describes the design of a system for filling out an Individual Income Tax Return. The system uses a semantic network knowledge base to retain information so as to carry on a natural dialogue with a user. This knowledge base is used to elicit information from the user, place this information in the knowledge base to determine his income tax status, respond to simple questions from the user, and actually fill out his Individual Income Tax Return.

Two Views of Data Semantics: A Survey of Data Models in Artificial Intelligence and Database Management

Harry K.T. Wong and John Mylopoulos

The goal of this paper is to establish that there exists a strong relationship between current issues of data models in Database Management and representations of knowledge in Artificial Intelligence. The different data modelling techniques that have been used in the two research areas are surveyed and classified into Predicate Calculus-basedm Network, and Procedural ones. The similarities and differences between them are presented.

Planning Speech Acts

C. Raymond Perrault and Philip R. Cohen

We are building a program that conducts a dialogue with a user, helping him to perform a task. The need for a model of the user's beliefs and goals in such a program is stressed. We show how such a user model can be organized and how speech acts can be represented as operators in a STRIPS-like planning system. Examples of the development of plans incorporating speech acts are given.

Progress Report from the "Computer Model of Conversation" Project

C. Raymond Perrault

The past year has seen considerable progress in the development of our model of conversation. The project's aim is to improve man-machine communication by integrating language use with goal directed non-linguistic actions. We achieve this by formulating speech acts as operators in a STRIPS-like planning system, i.e. as programs with examinable effects and preconditions (Fikes and Nilsson [1971]). The speech acts performed by the participants in a conversation are assumed to influence their beliefs and goals; proper performance of speech acts also depends on these beliefs and goals. The system should thus maintain a model of its user.

From the point of view of language generation, our approach provides the system with reasons for its utterances. The illocutionary force of its utterances (e.g. whether they are requests, informs or questions), much of their propositional content (e.g. what is requested, conveyed, or asked) and the position of the speech acts in the overall task follow from the application of the problem solving algorithm to a given (possibly non-linguistic) goal. Similarly, the system ought not to perform a speech act unless its preconditions are satisfied. For example, the system should not request the user to perform some act unless it has good reason to believe that the user can in fact perform the act.

For language recognition, situating a speech act in a plan helps to resolve the speech act ambiguity problem (is "Can you pass the salt?" a question about ability or a request for the salt?) and useful information in dealing with utterances in which no illocutionary force indicator is present (e.g. noun phrase utterances).

Phil Cohen's program OSCAR generates plans containing speech acts. It contains a memory model including a model of the user's beliefs and goals, which in turn may include a model of the user's model of the system's beliefs and goals etc. The number of levels of belief/goal embedding increases automtically when necessary. The planning algorithm can operate on different levels, allowing the system to plan in terms of the plans of the user. OSCAR generates request and inform speech acts, as well as yes/no and Wh-questions, which are compositions of request and inform. The proper handling of questions requires the system to be able to know that the user knows a piece of information without the system itself knowing what that information is. OSCAR can also plan complex speech acts in a world involving several users with possibly different beliefs and goals.

OSCAR is organised as an object-centered (Bobrow and Winograd [1976], Levesque [1977]) active semantic network (Norman and Rumelhart [1975]), that includes partitions as in Hendrix [1975]. It is implemented in Spitbol on an IBM 370/165 under TSO. Graph manipulation routines were written by J. Mylopoulos and C. Reason and the interactive graph definition package is due to J. Allen. A description of OSCAR can be found in Perrault and Cohen [1977], and in Cohen's forthcoming thesis.

James Allen has been studying plan recognition in the OSCAR system. He is currently at work on recognizing speech acts contained in a collection of dialogues between passengers and a clerk at a train station. These were collected and partially analysed by Mary Horrigan (Horrigan

[1977]).

Corot Reason is investigating the use of the goal structure in determining noun phrase reference, following suggestions in Searle [1969].

References

Bobrow, D. G., and T. Winograd, "An Overview of KRL, a Knowledge Representation Language", Tech. Report CSL-76-4, Xerox Palo Alto Research Center, 1976.

Fikes, R.E., and Nilsson, N.J., "STRIPS: A New Approach to the Application of Theorem Proving to Problem Solving," Artificial Intelligence, 2, pp. 189-208, 1971.

Hendrix, G., "Expanding the Utility of Semantic Networks through Partitioning", Proceedings of IJCAI4, Tbilisi, 1975, pp. 115-121.

Horrigan, M., "Modelling Simple Dialogs", Tech. Report 108, Dept. of Computer Science, Univ. of Toronto, 1977.

Levesque, H. J., "A Procedural Approach to Semantic Networks", Tech Report, Dept. of Computer Science, University of Toronto, 1977.

Norman, D., and D. Rumelhart (eds.), *Explorations in Cognition*, W. H. Freeman and Co., 1975.

Perrault, C.R., and P.R. Cohen, "Planning Speech Acts". A. I. Memo 77-1, Dept. of Computer Science, U. of Toronto, June, 1977.

Searle, J. R., *Speech Acts*, Cambridge Univ. Press, 1969.

Computer Analysis of Temporal Reference

Robin Cohen

In order to classify the temporal reference in English sentences, it is necessary to work on two levels. The sentences must first be analyzed linguistically for temporal references, and then these references must be stored into some representative structure which can be accessed easily.

The linguistic theory developed here is an elaboration of the theory of SRE conceived by Reichenbach [REICHENBACH 47] and studied by Hornstein [HORNSTEIN]. Essentially, tense is seen as the key to determining time references. Each tense is then conceptualized as a configuration of three markers - S, representing the time of speech, E, representing the time of event, and R, representing the point of view of the utterance. The ordering relations among the markers is to reflect the actual ordering in time of the represented speech, event, and reference times (e.g. simple past is represented by the configuration: $E,\underline{R}S$ - E is previous to S since the time of event is prior to the time of speech). Rules are set up to govern the manipulation of the SRE configuration in the presence of syntactic constructions relevant to time, and to assign a time interpretation to a sentence, based on the underlying SRE configurations of its clauses.

The set of linguistic analysis algorithms we develop is extensive, affording an interpretation for the underlying events of sentences of varied grammatical construction. We consider such constructions as temporal adverbs, temporal connectives (when, while, after, and the controversial before), nontemporal connectives, relative clauses, analysis between sentences, and special constructions such as continuous actions, hypothetical events, and complementizer phrases with speech acts. The analysis rules allow an entire paragraph to be analyzed as a whole, with the necessary relations between all events exposed.

The sturcture we create to hold the picture of time is an ordered time-line, inspired by the model of Kahn [KAHN 75]. The ordered time-line consists of events with definite temporal specifications, arranged chronologically. Then, attached to various events are temporal chains of other events.

The program devised to test the theories is designed to operate on a paragraph of English sentences, analyze the temporal references according to the developed linguistic algorithms, create a picture of time, and respond to questions about the temporal relations between events in the paragraph. The design of the time-line is a set of labelled nodes and edges, maintained in three graphs: TIMES, containing nodes representing the different times, linked by 'AFTER' edges, EVENTS, containing nodes representing the different events and their temporal relations, and TIME-LINE, using nodes from both TIMES and EVENTS to show events with definite time specifications. The separation into three graphs allows organization of search methods for the question-answerer.

The theory is designed to be used as a front-end analysis for larger systems which need temporal information. It is hoped that the methods will be of particular use to conversation understanding systems, which may make use of the additional information on the time of speech.

References

[COHEN 77] Cohen, Robin; "Computer Analysis of Temporal Reference"; U. of Toronto Technical Report No. 107, May 1977.

[HORSTEIN 75] Hornstein, Norbert; "As Time Goes By: A Small Step Towards a Theory of Tense"; Montreal Working Papers in Linguistics, Volume 5, December 1975.

[KAHN 75] Kahn, Kenneth M.; Mechanization of Temporal Knowledge; MIT Project MAC Technical Report 155, September 1975.

[REICHENBACH 47] Reichenbach, Hans; Elements of Symbolic Logic; The Macmillan Co., New York, 1947.

University of Waterloo
Waterloo, Ont.

AI At The University of Waterloo

In Waterloo there is a small group in the Department of Computer Science engaged in teaching and research in AI. In teaching, our most important activity is CS 486, our 4th year undergraduate introduction to AI. It will be taught for the third time in the Winter Term off 1978. The course content is roughly Winston's text with the part on programming replaced by similar material on PROLOG (see blurb below). The last time CS486 was taught we had students do the following assignments in PROLOG:

1) basic list manipulation

2) AND-OR trees, with an application to a trivial game

3) paths in graphs, with an application to solving a jugpcuring problem

4) analyzing "pictures" with Huffman's theory of scenes with polyhedral objects

Tom Pietzrykowski and Phil Ccx are currently working on a graphical system for mechanical deduction, which overcomms many of the deficiencies of linear deduction systems. In particular, unification is performed in such a way that backtracking can be intelligently done in the case of unification failure.

Maarten van Emden is interested in logic programming, working on applications in defining and querying databases; also on verification, synthesis, and abstract modelling of ordinary (i.e. non-logic) programs. He is studying definition and implementation of an improved version of PROLOG. (By the way, logic programming is an active and fruitful field of research; those interested should ask Maarten for a copy of the forthcoming issue of the Logic Programming Bulletin).

The Roberts Interpreter For Prolog

PROLOG is an interactive program language for a wide variety of symbol manipulaton problems. PROLOG is a language of "very high level" because it allows a user to program by writing specifications of algorithms written as recursive definitions in a simple mathematical formalism. The formalism is first-order predicate logic; PROLOG is based on this logic in roughly the same way as LISP is based on lambda-calculus. The comparison with LISP is appropriate because the domain of application is the same and because recent implementations of PROLOG run comparable programs at roughly the same speed as LISP interpreters do.

Among the advantages of PROLOG are the following features: recursive data structures akin to those proposed by Hoare, matching between patterns for recursive data structures, a relational view of data, procedure call by pattern matcing, and interchangeability of relational data and procedures. Several of these features have been included into the successors of LISP, such as MICEOPLANNER, QLISP, CONNIVER, which have been used in artificial intelligence research. In PROLOG these features are not ad-hoc additions but natural consequences of the way in which PROLOG is based on predicate logic. As a result the language definition is simple and the interpreter is small.

The computational use of logic and the design of the PROLOG language were developed in 1971 by R.A. Kowalski of Imperial College (then at the University of Edinburgh) and A. Colmerauer of the University of Aix-Marseille. A team headed by Colmerauer built several interpreters. One of these was written in FORTRAN and has been used in several other universities. D. Warren of the University of Edinburgh has designed a compiler for PROLOG which has been implemented for the DEC-10 macine by Warren and by L.M. Pereira and F. Pereira of the National Civil Engineering Laboratory in Lisbon.

The Waterloo interpreter for PROLOG has been adapted from the Marseille FORTRAN interpreter by G.M. Roberts of Calnek, Price and Associates (then a student in the University of Waterloo). Roberts also programmed in System/360 assembler code the interpreter is the same as that of the Marseille interpreter, there are several important improvements:

1) Adequate error diagnostics
2) Facilities for recovering from error situations
3) A more powerful and balanced repertoire of system functions
4) Greatly increased speed of translation and execution.

With the Warren compiler or the Roberts interpreter, PROLOG has about the same speed as a LISP interpreter for comparable programs (for example, not using the PROG feature of LISP). Because of the advanced features and simple language definition PROLOG is a very attractive alternative to LISP for course use and research in areas requiring flexible symbol manipulation such as symbolic mathematical computation, artificial intelligence, compiler writing, etc.

The Roberts interpreter occupies approximately 20 K bytes of storage and runs at present only under VM/370. It is documented in Roberts's "An Implmentation of PROLOG". For further information on PROLOG see D. Warren, L.M. Perpira, F. Perpira: PROLOG-the language and its implementation compared with LISP. SIGART/SIGPLAN symposium on AI and Programming Languages, University of Rochester, August 1977.

University of Western Ontario
Department of Computer Science
London, Ont.

The Role of Conventions in Generating Descriptions Of Line Drawings
A.H. Dixon

The paradigm adopted by many scene analysis systems requires the generation of a line drawing of the image from which a "high-level" program attempts to construct a description. This line drawing is often stored as a graph whose vertices correspond to the endpoints of the line segments discovered.

Diagrams are examples of scene domains where the scenes themselves are line drawings. However, a diagram is more than just a line drawing as it usually includes additional features called labels. Further, although there is frequently a large number of ways in wich a particular diagram can be described, it is most often described in nearly, if not exactly, the same way. One reason for this may be that knowledge about certain conventions about diagrams for the particular problem domain are being used to formulate a description. These conventions appear to be rules which govern the ways in which labels are assigned and also how the line segments from which the diagram is constructed, are oriented.

A model which incorporates knowledge about labelling and orientation conventions for a particular class of diagrams with more general knowledge about the problem domain was presented at IJCAI in Boston this summer (1). Using a subset of Euclidean geometry as the problem domain, the model has been implemented in the planner type language POPLER 1.5. With test cases selected from a textbook on Euclidean geometry (2), the implementation generates a description for a diagram which is intitially specified as a set of straight line segments. The generated description can then be compared with that given in the text for which the diagram was an illustration.

In the system the role of a convention is to specify the search for some particular feature of a line drawing. If successful, a description of the feature is inserted into an associative database, and antecedent theorems are invoked. Such theorems attempt to formulate more complex descriptions as a consequence of additions to the database. The final state of the database represents a description of the diagram.

The initial assertions therefore play an important role in determining which antecedent theorems are tried first, as this affects the intermediate contents of the database from which the final description is derived. Since the role of the conventions is to make these initial assertions, their relative importance can be observed by altering the order in which they are applied to the diagram.

Text examples have been taken from a subset of Euclidean geometry involving figures containing only straight lines and using cnly alphabetic vertex labellings. A small number of conventions which seem the most obvious candidates have been represented procedurally in the knowledge base of the system. A typcial example is: "The primary triangle, it it exists, is frequently labelled ABC".

At present it appears as though there does not exist a set of conventions whose optimum order of application is uniquely defined. This is not too surpprising, since the use of conventions seems to have evolved from the effort to provide more lucid diagrams, rather than being defined formally. However, the conventions do provide an efficient way of generating appropriate descriptions. The most appropriate description for the purposes of evaluation is the one which accompanies the diagram as given in the text. Those descriptions which did not satisfy this criterion were nonetheless reasonable alternatives and in no case was a totally inappropriate description generated.

1. A.H. Dixon. "Generation of descriptions for line drawing", Proceedings IJCAI, 1977 (to appear)

2. Isaac Todhunter (ed), EUCLID'S ELEMENTS, J.M. Dent and SONS, 1933.

Image-Matching In The Context Of A Semi-Autonomous "Mars Rover".

A.K. Dewdney

Research since last May has centred on the construction of an experimental test program to determine the feasibility of steepest-descent image-algorithms [1] for use in detection, tracking and orientation applications aboard the proposed 1984 Mars Rover vehicle currently being planned by J.P.L. A vist to J.P.L. was made in December, 1976, to visit the project and discuss the current visualization of the mission with personnel involved.

The test vehicle being used at that time consisted of a 4-wheeled cart with onboard AD/DA conversion equipment, stereo television cameras and a laser range finder. Programmers were constructing a rather high-level package embodying several current A.I. techniques. It was not immediately clear whether all these techniques would be useful - even if they could be made reliable enough to function at the level of autonomy visualized by the Rover research term.

Perhaps only a low level of autonomy is really necessary: the Rover's trip to some not-too-distant feature could easily be broken into segments, each autonomously controlled but based on cues supplied by "earth control". Such cues would consist basically of segments of the current panorama indicated by Earth. One would be the target, say, a large boulder or mound, and others would be various features along the way, considered as easily recognizable landmarks or, possibly, hazards to be avoided.

The research currently conducted by me uses a space-series of blow-up, detailed pictures from a "planetary" site near Sudbury. There can be placed in front of our television camera and fed into the Interdata for analysis. Through them, a wide variety of Rover movements can be simulated. The test program is now nearing completion.

The basic question to be tested is how long a program can keep a given feature "in view" as it moves through the landscape and, therefore, has continually to view it from a different angle and at differert distances. Interestingly enough, if the space sequence of such appearances can be broken down finely enough, say in decimeter steps at worst, there is reasonable hope of being able to match the stored image of this feature with the current image so that (a) the same feature is found again, and (b) it is still centred in the storage matrix allotted to it.

[1] A.K. Dewdney. Analysis off a steepest-descent image-matching algorithm. Pattern Recognition, (to appear).

University of Washington
Seattle, Washington

AI at U of W

People actively involved in Artificial Intelligence at the University of Washington include Steven L. Tanimoto (machine vision), Earl B. Hunt (memory and concept learning), Alistair Holden (speech recognition, image processing, air traffic control), and David Johnston (models of learning). At this time their efforts are spread out over

3 departments: Computer Science, Psychology and Electrical Engineering.

For further information contact Steven L. Tanimoto, Assistant Professor, Department of Computer Science, 112 Sieg Hall, FR-35, University of Washington, Seattle, Washington 98195.

University of Alberta
Department of Computing Science
Edmonton, Alberta

Report of the University of Alberta Working Group on Adaptive Systems, Biological Information Processing, and Simulation (ASBIPS ?)

In addition to the recently completed study of adaptation using reproductive meta-plans (summarized below), the group is looking at representation problems associated with applications of genetic algorithms for adaptation, spike train analysis for neurophysiological experiments, simulation of pattern formation in developing systems, simulation of spatial phenomena in population ecology, and aspects of the measurement of shape. A book tentatively entitled Biological Information Processing: Current Theory and Computer Simulation is in preparation by J.R. Sampson, with K. Morgan of the Genetics Department. In an ancillary project, implementation has progressed to the syntax acquisition stage in the study of computer acquisition of natural language (see the report on pp. 52-54 of Vol. 1, No. 3).

Adaptive Search Using a Reproductive Meta-Plan
R.E. Mercer and J.R. Sampson
To appear in Kybernetes, 1978

A reproductive plan is a type of adaptive procedure devised by Holland which embodies many principles found in the adaptation of natural systems through evolution. This study develops a reproductive meta-plan, an adaptive procedure that controls the modification of parameter values in a reproductive plan. After a brief review of Holland's formalizaticn of reproductive plans, a non-reproductive meta-plan devised by Cavicchio is discussed. In response to some of the limitations of Cavicchio's approach, a reproductive meta-plan is developed. Experiments with computer implementations of both meta-plans show improvement in a reproductive plan's utility gain when the reproductive meta-plan is used.

Organizing Factual Knowledge in a Semantic Net
R.G. Goebel
M.Sc. Thesis, 1977
Present address: Dept. of Computer Science, UBC.

An organization for the factual knowledge in a semantic net has been formulated in which the facts involving any given (particular or generic) concept are accessible via a topic hierarchy associated with that concept. The topics in each hierarchy are governed by multivalued topic predicates whose degree of truth determines the extent to which a given proposition belongs to a given topic. In addition, the network representation itself has been extended to include credibility distributions over truth values for propositions. This permits the simultaneous representation of degrees of belief and degrees of truth,and the calculation of these quantities for logical combinations of independent propositions. A working implementation has been produced, including an input parser for an "English-like" infix form of predicate calculus, and the full topic hierarchy mechanism for retrieval of facts relevant to a question. The mechanism appears to provide a satisfactory solution of the "symbol-mapping" problem in Artificial Intelligence.

<u>Decision Theory and Automatic Planning</u>
D.T. Johnson
M.Sc. Thesis, 1977

A decision-theoretic planning strategy in the tradition of control strategies such as mini-max game playing, A* search (Hart, Nilsson, and Raplael) and MULTIPLE (Slagle) is proposed. However, the strategy uses changes in both probability and expected cost, and takes into account both the cost of developing a plan and of executing the plan. A hand-simulated STRIPS-like robot is described, using a simplified form of the proposed decision method. The workability and advantages of the method are demonstrated.

<u>Computer Matching of Stereo Pictures</u>
R.K. Jain
M.Sc. Thesis, 1977

The thesis describes a fast method for finding corresponding points in a stereo picture pair. First target areas are selected which contain sufficiently many non-colinear edge points and are therefore suitable for matching by exhaustive search. For such target areas, potential matches are found by a newly developed technique called "up and down correlation." Regions of constant displacement are then grown with a refined version of Hannah's algorithm. Efficiency is ensured by the use of running sums and other techniques. Both theoretically and in practice, the reliability and speed of the algorithm compare favourably with those of earlier algorithms.

<u>Predictability and Randomness</u>
L.K. Schubert
Tech. Report TR77-2, 1977

The paper is concerned with the probabilistic prediction of infinite sequences. Relationships are explored between the class of all effective predictive methods, a particular class of methods proposed by Solomonoff, and the theories of randomness of Martin-Lof and Schnorr.

Knowledge and Inference: Work in Progress

L.K. Schubert and students under his direction are working on the problem of procedural attachment in a semantic net, efficient algorithms for spatial reasoning (to supplement net-based inference methods), and grammatical inference. The ultimate objective is a language understanding and reasoning system written in ALAI (see C. Gray, ALAI: A Language for Artificial Intelligence, M.Sc. Thesis, Dept. Computing Science, 1976). An unrelated project which is nearing completion is the development of a new edge-following algorithm.

On the Detection of Changes in Digital Images
S.K. Kenue
Ph.D. Thesis, 1977

This thesis deals with the detection of changes in digital pictures. If pictures have been obtained at different times and under different environmental conditions, two basic steps, namely registration and normalization, are essential for change detection. A normalization method is developed which changes the histogram distribution of a picture to any desired distribution. Also, a picture enhancement method is proposed for enhancing small features, which may not be ordinarily visible. Applications of this method include improved edge and contour detection and noise removal.

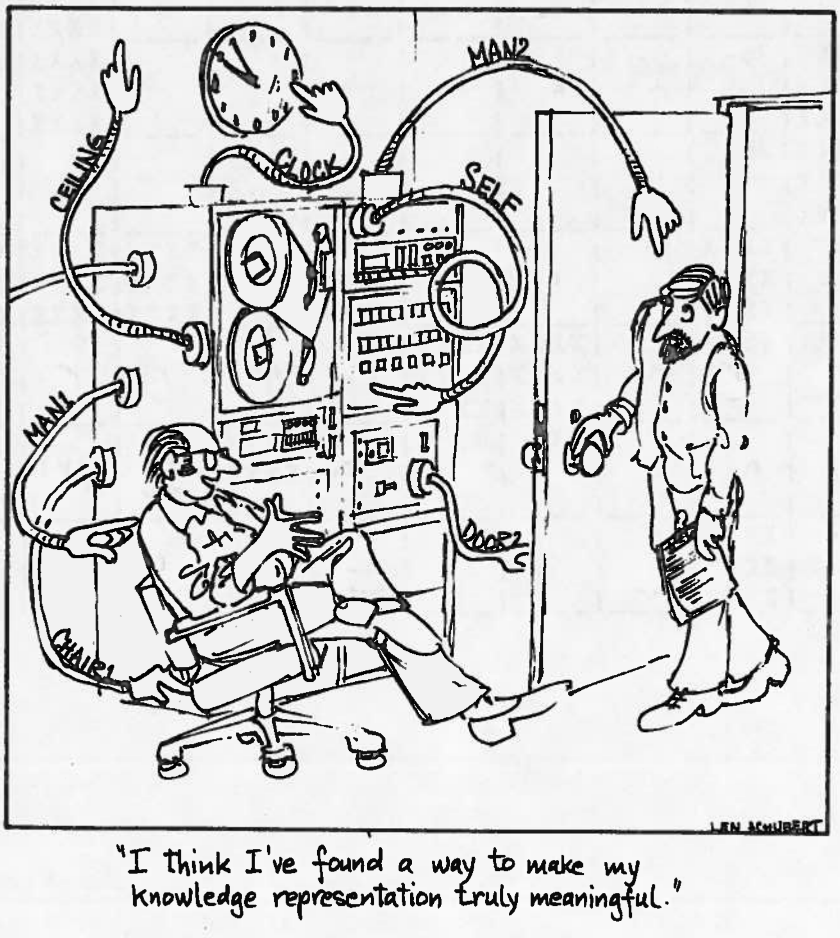
Registration is concerned with the precise alignment of two or more pictures. An automatic selection procedure for control points is given and some problems encountered in selecting them are presented. If the coordinates of the control points are incorrect, an interative feedback approach is presented to correct them. The procedure was applied to a number of pictures of sizes 256*256 to 501*501, with a resulting registration accuracy of about 1 to 2 pixels for the first order transformation and about 1 pixel for the second order transformation. The registered pictures were generated by the nearest neighbor, bilinear interpolation, and cubic convolution techniques. A technique for the registration of pictures with large scale differences is given. For template matching, a fast algorithm which uses heuristic information is developed and compared with other methods.

The detection of changes is accomplished by subtracting gray levels in the corresponding pixels of the registered pictures. The difference pictures are then thresholded to yield binary change pictures, in which isclated pixels are removed.

Book in Preparation: FROM ASSOCIATIONS TO STRUCTURE
Kellogg Wilson
Departments of Computing Science and Psychology

This book argues for a modified form of Hebbian association theory to account for a wide range of findings, especially the development and use of language. It attempts to integrate findings from Computer Science regarding the processing of natural language with psycholinguistic theory and evidence. In addition, there is coverage of conceptual structure beyond the sentence, the development of language and cognition, the relation between consolidation and induction, and the nature of consciousness and adaption. Results from a variety of areas of psychology and computer science are discussed.

Much current cognitive psychology, especially of the structuralist variety, argues for the innate determination of much of cognitive processes with top-down control over behavior. Most artificial intelligence programs implicitly take the same position since they are usually endowed by their creators with considerable a priori structure and no capacity to learn. While this practice can be justified for some limited purposes, artificial intelligence does not adequately model the (perhaps rare) human ability to engage in intelligent abstraction from experience.



"I think I've found a way to make my knowledge representation truly meaningful."

PUZZLE SECTION

(answers on last page)

CROSSWORD

by Jeff Sampson

1 A	2 T	3 N	XXXX	4 B	5 E	6 T	7 A	XXXX	8 S	9 S
10 C	U	E	XXXX	11 E	G	A	L	12 I	T	E
13 T	I	T	14 H	E	XXXX	XXXX	15 G	O	A	L
XXXX	XXXX	XXXX	16 E	XXXX	17	18	O	XXXX	19 R	F
20 S	21 A	22 M	U	23 E	L	A	R	24 T	XXXX	XXXX
25 ?	XXXX	26 O	R	R			I		XXXX	27 I
XXXX	XXXX	28 N	I	G			T		29	E
30	31 F	XXXX	32 S	O	S	XXXX	33 H	XXXX	XXXX	XXXX
34 S	L	35 O	T	XXXX	XXXX	36 A	M	37 U	38 C	39 K
40	I	N	I	41	42	X	XXXX	43 B	M	I
44	P	XXXX	45 C			E	XXXX	46 C	G	D

ACROSS

1. Augmented Transition Network
4. Type of cutoff
8. WW II German military unit
10. Hint
11. Liberte, _____, fraternite
13. Collect a tenth part
15. GPS state
16. With 29 Down, symbol for Element No. 68
17. With 18 Down, proportionately
19. Type of amplifier: Abbr.
20. Early AI gamesman: Last name plus nickname
25. With 33 Across, academic medical facility: Abbr
26. SHAKEY's course, at times
27. See 21 Down
28. Table used by 20 across
30. _____-added demon
32. Code signal
33. See 25 Across
34. Frame terminal
36. Run _____
40. Search process
43. English automotive firm: Abbr.
44. Proofreader's notation
45. Type of grammar
46. Ruminant's repast

DOWN

1. Schank primitive
2. Parson bird
3. Similarity or semantic _____
4. Spelling _____
5. For example: Abbr.
6. Expression of gratitude: Brit. colloq.
7. Effective procedure
8. A-_____ algorithm
9. Ego, identity
12. Class of computer operations: Abbr.
14. Serving to discover
17. SHRDLU's reasoning devices
18. See 17 Across
20. West Coast AI center
21. With 27 Across, field in CS
22. _____ amis
23. Cogito, _____, sum
24. Initials of Southern US school
27. That is: Abbr.
29. See 16 Across
30. Doctrines
31. _____-flop
35. PUT-_____, Blocks-World procedure
36. Cutting tool
37. West Coast AI center
38. Eastern AI center
39. Young goat
41. Milliampere: Abbr.
42. _____ it were

CRYPTOGRAMS

1. YJZW NY S XJV FQM XHGZW

2. DZNLQXJS NZCJPR ZH XFZQYFX WMC PWMYQWYJ

3. QWGJXQ FXMN RYZJXQ YCMJXN ZWTYZB

DOUBLE-CROSTIC

by Anne Brindle

1 W	2 U	3 X	4 P	5 V	6 T	7 D	8 J	XXXXX XXXXX XXXXX	9 N	10 Z2	11 G	12 S	13 F	14 U	15 J
XXXXX XXXXX XXXXX	16 B	17 Z1	18 L	19 N	20 O	21 R	XXXXX XXXXX XXXXX	22 Y	23 A	24 P	25 L	XXXXX XXXXX XXXXX	26 Z	27 Z3	28 H
29 Y	30 X	31 G	32 S	33 D	34 Z2	XXXXX XXXXX XXXXX	35 C	36 F	37 P	38 H	XXXXX XXXXX XXXXX	39 A	40 K	41 E	42 U
XXXXX XXXXX XXXXX	43 N	44 Z	45 J	46 T	47 B	48 R	49 I	50 L	51 K	XXXXX XXXXX XXXXX	52 P	53 Q	54 D	XXXXX XXXXX XXXXX	55 A
56 U	57 M	XXXXX XXXXX XXXXX	58 Z	59 Z3	XXXXX XXXXX XXXXX	60 F	XXXXX XXXXX XXXXX	61 Z1	62 O	63 X	64 W	XXXXX XXXXX XXXXX	65 C	66 D	67 J
XXXXX XXXXX XXXXX	68 V	69 P	70 G	71 B	72 O	73 Z2	74 D	75 I	XXXXX XXXXX XXXXX	76 P	77 Z1	78 N	XXXXX XXXXX XXXXX	79 L	80 H
81 Q	82 Z3	83 R	84 D	85 A	XXXXX XXXXX XXXXX	86 P	87 F	88 Z	89 K	90 T	XXXXX XXXXX XXXXX	91 Z2	92 P	93 L	XXXXX XXXXX XXXXX
94 X	95 A	XXXXX XXXXX XXXXX	96 D	97 V	XXXXX XXXXX XXXXX	98 P	99 N	100 X	101 W	102 C	103 K	XXXXX XXXXX XXXXX	104 P	105Z1	106 S
XXXXX XXXXX XXXXX	107 P	108 F	109 A	XXXXX XXXXX XXXXX	110Z2	111 P	XXXXX XXXXX XXXXX	112 Z	113 D	114 O	115 V	116 F	117 J	118 D	119 C
XXXXX XXXXX XXXXX	120 A	121Z2	122 B	123 P	124 G	125 I	XXXXX XXXXX XXXXX	126 P	127Z2	128 L	129 R	130 T	131 E	132 H	133 U
134 D	135 O	136 X	137 W	XXXXX XXXXX XXXXX	138 C	139 A	140 N	XXXXX XXXXX XXXXX	141 M	142 Q	143 O	144 P	145Z1	146 A	147 T
148 H	149 N	150 R	XXXXX XXXXX XXXXX	151 H	152 B	153 M	XXXXX XXXXX XXXXX	154 P	155 V	XXXXX XXXXX XXXXX	156 E	157 L	158 B	XXXXX XXXXX XXXXX	159Z2
160 J	161 Z	162 U	163 C	164 H	165 K	166 F	XXXXX XXXXX XXXXX	167Z2	168 U	169 D	170 E	171Z1	172Z3	173 W	174 P
XXXXX XXXXX XXXXX	175 Z	176 S	177 N	178 Y	179 H	180 H	181 D	XXXXX XXXXX XXXXX	182 K	183 L	184Z2	185Z3	XXXXX XXXXX XXXXX	186 I	187 T
188 S	189Z2	190 I	191 H	192 R	193 P	XXXXX XXXXX XXXXX	194 J	195 A	196 M	197 R	198Z1	199 W	200 B	201 Q	XXXXX XXXXX XXXXX
202 U	203 R	204 I	205 P	206 C	XXXXX XXXXX XXXXX	207 J	208 K	209 L	210 P	211 V	212 A	213 Z	214 Y	XXXXX XXXXX XXXXX	XXXXX XXXXX XXXXX

INSTRUCTIONS

First guess the words which satisfy several of the definitions, and fill in the numbered blanks. Then copy these letters into the corresponding numbered spaces in the diagram. (The letters in the diagram help you go the other way.)

When filled, the diagram will contain a quotation from a book. Only the horizontal direction is significant, and squares with XXXXX denote blanks between words. (The end of a line is not necessarily the end of the word.)

As the diagram develops you can use context to fill in letters which will help you guess more words below. Finally, the first letters of all the words (in order A - Z3) will spell out the the author and title of the book from which the quotation in the diagram comes.

WORDS

A. "_____ hearts we see thee rise", from "O Canada", (2 words) — 39 212 120 195 146 55 95 109 23 139 85

B. Fiery, produced by volcanic action — 122 71 200 158 16 47 152

C. Celebrated London prison (until 1902) — 65 163 35 102 138 206 119

D. Memorable features of Konisberg (2 words) — 54 113 169 7 84 118 33 134 96 74 66 181

E. Bonelike structure — 41 170 36 156 131

F. _____ Monkey, PLANNER program — 108 116 87 60 13 166

G. Near — 70 11 124 31

H. A. Christie murder mystery (2 words, followed by "of Rye") — 80 164 132 148 38 180 28 191 151 179

I. Rest — 190 49 186 204 125 75

J. One who is unfaithful (comp.) — 194 67 160 207 117 45 15 8

K. Narrow strip of land connecting two bodies of land — 40 103 165 208 182 89 51

L. Without delay, immediately — 18 183 50 128 25 93 209 79 157

M. Conception, pattern, mental image — 196 57 153 141

N. Pantomimes, riddles — 43 99 149 78 177 140 19 9

O. Son of Daedalus — 62 135 143 20 72 114

P. Authors, "The Design and Analysis of Computer Algorithms" (4 words) — 69 52 111 107 154 4 24 37 92 76 98 104 210 174 205 144 193 126 86 123

Q. Canadian Minister of Justice in 1974 — 81 53 142 201

R. Constantinople — 83 21 48 129 197 203 192 150

S. "Let there be _____", 2 Samuel 1:21(2 words) — 176 32 106 12 188

T. First-rate; highest point (comp.) — 90 147 46 6 187 130

U. Trespass, intrude — 56 168 14 133 2 202 162 42

V. Chance, fate (2 words, followed by "the draw") — 68 5 115 211 97 155

W. Composite organism which grows on rocks and trees. — 137 199 1 64 173 101

X. Collison; strong effect — 100 3 30 136 63 94

Y. Clutch, grasp — 214 22 29 178

Z. Home of the Eskimos — 175 112 26 88 213 58 44 161

Z1 _____ Island Pine, evergreen houseplant — 105 145 61 17 77 171 198

Z2 Creature from "Alice in Wonderland" (2 words) — 10 91 127 34 121 167 184 189 159 73 110

Z3 Call forth, elicit — 27 172 59 82 185

ANSWERS TO PUZZLES

CROSSWORD

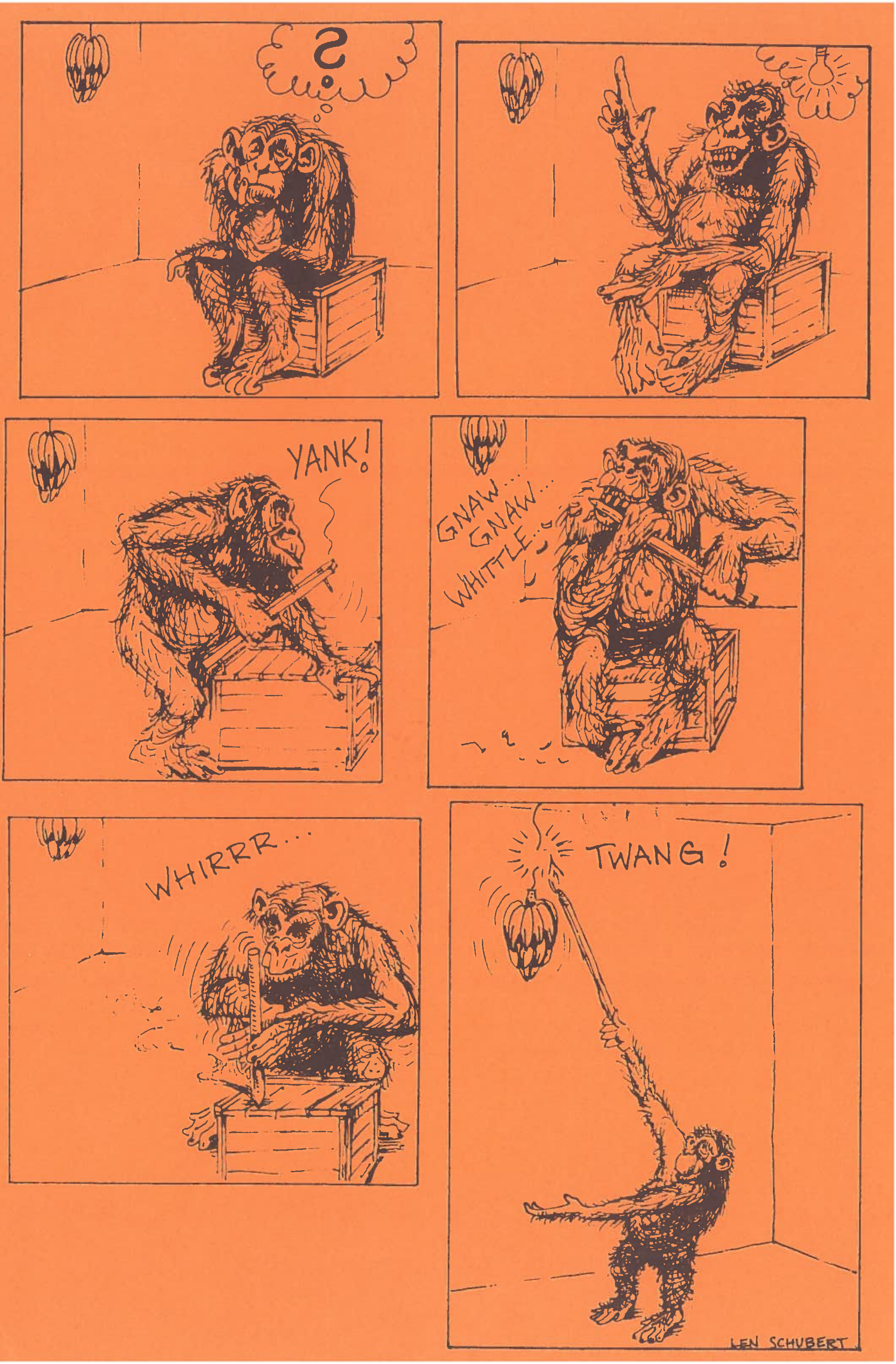
1 A	2 T	3 N	XXXX	4 B	5 E	6 T	7 A	XXXX	8 S	9 S
10 C	U	E	XXXX	11 E	G	A	L	12 I	T	E
13 T	I	T	14 H	E	XXXX	XXXX	15 G	O	A	L
XXXX	XXXX	XXXX	16 E	XXXX	17 P	18 R	O	XXXX	19 R	F
20 S	21 A	22 M	U	23 E	L	A	R	24 T	XXXX	XXXX
25 U	XXXX	26 E	R	R	A	T	I	C	XXXX	27 I
XXXX	XXXX	28 S	I	G	N	A	T	U	29 R	E
30 I	31 F	XXXX	32 S	O	S	XXXX	33 H	XXXX	XXXX	XXXX
34 S	L	35 O	T	XXXX	XXXX	36 A	M	37 U	38 C	39 K
40 M	I	N	I	41 M	42 A	X	XXXX	43 B	M	I
44 S	P	XXXX	45 C	A	S	E	XXXX	46 C	U	D

CRYPTOGRAMS

1. PICK UP A BIG RED BLOCK
2. COMPUTER MODELS OF THOUGHT AND LANGUAGE
3. DONALD PLUS GERALD EQUALS ROBERT

DOUBLE CROSTIC

> Computer science offers rich metaphors. Work with computers has led to a rich new language for talking about how to do things and how to describe things. Metaphorical and analogical use of the concepts involved enables more powerful thinking about thinking.
>
> --Winston, Artificial Intelligence


?
YANK!
GNAW... GNAW... WHITTLE...
WHIRRR...
TWANG!
LEN SCHUBERT