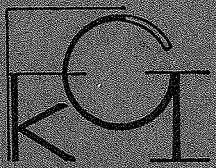
RUNDBRIEF

DER

FACHGRUPPE KÜNSTLICHE INTELLIGENZ

IN DER GESELLSCHAFT FÜR INFORMATIK



Nummer 13 Mai 1978

INHALT

IMPRESSUM

Dieser Rundbrief der Fachgruppe "Künstliche Intelligenz" (KI), vertreten durch den Fachausschuß 6 "Digitale Verarbeitung kontinuierlicher Signale" in der Gesellschaft für Informatik (GI), erscheint in unregelmäßigen Abständen (etwa vierteljährig) und wird den Mitgliedern der Fachgruppe kostenlos zugesandt. Mitglied der Fachgruppe wird jeder, der beim Herausgeber um Eintrag in die Adressenkartei nachsucht. Mitgliedschaft in der GI ist zwar nicht Voraussetzung, sollte aber schon wegen der Kosten, die die GI trägt, die Regel sein (Aufnahmeantrag wird auf Anfrage zugesandt).

Das Ziel dieses Rundbriefes ist es, aktuelle Informationen unter den Mitgliedern der Fachgruppe auszutauschen. Der Herausgeber bittet daher die Leser um möglichst rege Zusendungen von Beiträgen aus dem gesamten Gebiet der KI. Im einzelnen kann es sich dabei u.a. um folgendes handeln:

- Kurzfassungen von Arbeiten oder Berichten
- Beschreibung von laufenden Projekten
- Diskussion von wissenschaftlichen oder wissenschaftspolitischen Themen in Form von Zuschriften an den Herausgeber
- Berichte von Tagungen, Auslandsreisen, etc.
- Hinweise auf interessante Veranstaltungen, Adressenänderungen, offene Stellen, Stellengesuche, etc.

Mit der Zusendung an den Herausgeber ist das Einverständnis des Autors zur Veröffentlichung im Rundbrief verbunden. Die Beiträge werden nicht begutachtet und geben nur die individuelle Meinung des jeweiligen Autors wieder. Sie werden photomechanisch direkt vom Original übertragen und können in Deutsch, Englisch oder Französisch abgefaßt sein.

Herausgeber: Wolfgang Bibel

Institut für Informatik
Technische Universität München
Postfach 202420
D-8000 München 2

Tel. (089) 2105-8186

Anmerkungen des Herausgebers

1) Drei Jahre nach der Gründung unserer Fachgruppe stehen wir vor einem Ereignis, das zweifelsohne für das Gebiet der Künstlichen Intelligenz in Deutschland einen vorläufigen Höhepunkt bedeutet, nämlich der

AISB/GI CONFERENCE ON

ARTIFICIAL INTELLIGENCE

Hamburg: 18–20 July, 1978

Schon das Interesse an der Tagung, das mit gut über 80 eingereichten Arbeiten bekundet wurde, aber mehr noch das Tagungsprogramm selbst (das diesem Rundbrief beigefügt ist), deuten auf eine wissenschaftlich interessante und ergiebige Veranstaltung hin. Zudem wird sie eine einmalige Gelegenheit bieten, die persönlichen Kontakte der über viele Fachgebiete hinweg verstreuten Interessenten an der KI zu vertiefen bzw. zu knüpfen, ohne die jeder Wissenschaftsbetrieb steril bleibt. Es wäre daher zu wünschen, daß insbesondere auch möglichst viele Mitglieder der Fachgruppe an dieser Tagung teilnehmen.

2) Diese Tagung bietet auch eine Gelegenheit, die weitere Arbeit und organisatorische Fragen der Fachgruppe in einem größeren Kreis von Mitgliedern zu diskutieren. Als Termin für eine solche Diskussion ist der

Mittwoch, 19. Juli 1978,

unmittelbar im Anschluß an die Vorträge des Nachmittags vorgesehen. Alle Mitglieder sind dazu herzlich eingeladen. Den genauen Ort und Termin entnehmen Sie bitte Ihren Tagungsunterlagen.

3) Im Forum dieser Nummer finden Sie einen Beitrag von D. de Champeaux zu einem Vortrag von E. Dijkstra, in dem dieser in einer ungewöhnlich scharfen Form gegen die KI ins Feld zieht. Ich verbinde mit dieser Erwiderung die Hoffnung, daß sie eine Diskussion entfacht, die die nutzlosen Gräben überwindet und zum gegenseitigen Verständnis beiträgt.

Die Heftigkeit- mit der die KI mannigfach unter Beschuß genommen wird, erinnert ja bisweilen geradezu an einen Glaubenskrieg, und zwar auch in der Hinsicht, daß sich hier eigentlich "Glaubensbrüder" bekämpfen, überwiegt doch das Einigende in der Zielsetzung bei weitem die sachlichen Unterschiede. So ist man beinahe versucht, die tieferen Gründe für diese Spannungen

in so menschlichen Motiven wie das Streben nach Macht und Ansehen zu suchen. Mit einer solchen Erklärung würde man aber nach meiner Ansicht dieses Phänomen nicht in seiner ganzen Tiefe ausloten können, ebensowenig wie mit dem manchmal gegebenen Hinweis auf uneingelöste frühere Vorhersagen von einigen KI-Leuten (werden doch solche Fehlprognosen in allen Gebieten gemacht, ohne daß sich jemand darüber sonderlich aufregt).

Vielmehr möchte ich die These aufstellen, daß aus einer etwas vergröbernden Distanz betrachtet zwischen der traditionellen Informatik und der KI ein methodologischer Unterschied erkennbar ist, der zwar (notwendigerweise) schlicht quantitativer Natur ist, dessen Ausmaß und Richtung aber in der relativen Sicht des Menschen, seines Selbstverständnisses und demgegenüber seiner Vorstellung von Maschinen, bei vielen eine empfindliche, irrationale Reizschwelle überschritten hat, wodurch solch starke Reaktionen ausgelöst werden. Wenn tatsächlich in dieser Richtung der eigentliche wunde Punkt zu suchen ist, so wäre es für Wissenschaftler angemessen, hierüber rational zu diskutieren, anstatt mit schäbigen Mitteln der Macht oder mit unsachlichen Abwertungen den vermeintlichen Gegner aus dem Felde schlagen zu wollen.

4) Als Redaktionsschluß für RB Nr. 14 ist der 21. 7. 78 vorgesehen.

INDUSTRIAL ROBOTS CONFERENCE

Stuttgart, 30. Mai - 1. Juni

Diese Konferenz vereinigt in sich das 8. International Symposium on Industrial Robots und die 4. Conference on Industrial Robot Technology und findet im Kongresszentrum von Böblingen (zusammen mit einer Ausstellung von u.a. mehr als 30 Robotern) statt.
Anfragen an

Mr. I. Schmidt, Organising Secretary 8ISIR/4CIRT
IPA-Stuttgart, Seestr. 100
D-7000 Stuttgart 1
Tel. 0711-293917 Telex 0723008

4TH INTERNATIONAL CONGRESS OF CYBERNETICS & SYSTEMS

Amsterdam, 21. - 25. August 1978

Dieser alle drei Jahre stattfindende Kongress (zuletzt in Bucharest, Rumänien) umfasst 3 Symposien, 7 Section Meetings (ua. über Artificial Intelligence), ein Open Forum, verschiedene Special Meetings, Austellungen, Poster Sessions, ein Internationales Computerschachturnier und die Verleihung der Norbert-Wiener-Preise.
Anfragen an

Dr. J. Rose (W.O.G.S.C.), College of Technology
Feilden Street
Blackburn BB2 1LH, Lancashire, England, Tel.0254-64321

CALL FOR PAPERS

FOURTH WORKSHOP ON AUTOMATED DEDUCTION
Austin, Texas
January 29-31, 1979

The main purpose of this workshop is the exchange of current results in the area of automatic formal reasoning. We encourage contributions from all methodological viewpoints. The objective is to share generally applicable understanding. Purely theoretical to purely applied results are solicited.

The following is a list of suggested areas of interest but it should not be considered exhaustive:

1. description of automatic deduction techniques
2. analysis (e.g. complexity) of automatic deduction techniques
3. methods of representation of knowledge and rules of deduction
4. applications of deduction
5. reports of implementations of deductive systems

You are invited to submit four copies of a full paper or an extended abstract (recommended length: 1500 words) to Program Chairman:

Prof. Sharon Sickel
Information Sciences
University of California
Santa Cruz, CA 95064

All submissions will be read by several members of the program committee, and evaluated on the basis of relevance, originality, and overall quality. It is important that the paper submitted convey the ideas and contain enough information to enable the program committee to determine the scope of the work. The author's name and affiliation should appear only on a cover sheet attached to one copy.

The deadline for submissions is August 15, 1978. Notification of acceptance will be sent by September 30, 1978. Camera-ready, final papers (maximum: 5000 words) are due November 15, 1978. Proceedings will be distributed at the workshop. In addition, the authors are expected to present the material in the workshop program.

Program Committee

Wolfgang Bibel
Gerard Huet
Aravind Joshi
Donald Loveland
Jorg Siekmann
Richard Waldinger
Richard Weyhrauch

Sharon Sickel, Chairman
William Joyner, Editor

Local arrangements are being handled by:

Prof. W.W. Bledsoe
Dept. of Mathematics
University of Texas
Austin, Texas 78712

1. Symposium der Deutschen Arbeitsgemeinschaft für Mustererkennung

"Bildverarbeitung und Mustererkennung"

in Oberpfaffenhofen bei München, ausgerichtet von der DFVLR

vom 11.10. - 13.10.1978.

1. Deutsche Arbeitsgemeinschaft für Mustererkennung (DAGM)

Die DAGM ist eine Arbeitsgemeinschaft von auf dem Gebiet der Mustererkennung tätigen wissenschaftlichen Gesellschaften. Ihre Aufgabe ist die Koordinierung von Tätigkeiten auf diesem Gebiet. Ihrer Zielsetzung gemäß beabsichtigt die DAGM in regelmäßiger Folge an wechselnden Orten ein Symposium abzuhalten. Die Organisation liegt jeweils bei Institutionen, die hierfür von der DAGM aufgefordert werden. Ziel des Symposiums ist ein Erfahrungsaustausch zwischen Fachleuten auf dem Gebiet der Mustererkennung.

2. Thema und Art der Veranstaltung

Für das 1. DAGM-Symposium wurden Probleme der Bildmanipulation, Bildbeschreibung und Klassifizierung als Themen ausgewählt. Es werden sowohl Übersichts- als auch Kurz-Vorträge (10 Minuten Redezeit plus Diskussion) über laufende Arbeiten erbeten. Zusätzlich ist beabsichtigt, Arbeitssitzungen über spezielle Themen abzuhalten. Außerdem können letzte Ergebnisse und Ausstellungsstücke präsentiert werden. Die Auswahl der Vorträge und die Programmfestlegung erfolgen durch ein Programm-Komitee.

3. Termine

Vortragsanmeldung und Kurzfassung (12-15 Zeilen	bis 17.3.1978
Bestätigung der Annahme des Vortrags erfolgt	bis 1.5.1978
Vorlage des druckfertigen Manuskriptes	bis 1.8.1978

4. Organisatorisches

Das 1. DAGM-Symposium wird in Oberpfaffenhofen von der DFVLR ausgerichtet. Neben den Vorträgen und Diskussionen sind ein Rahmenprogramm und Arbeitssitzungen an einem interaktiven Bildverarbeitungssystem vorgesehen.

Organisations-Komitee:

Dr. E.Triendl (Inst. f. Nachrichtentechnik, DFVLR, Oberpfaffenhofen)
J.P. Foith (Inst. f. Informationsverarbeitung in Technik und Biologie (IITB), Fraunhofer-Gesellschaft, Karlsruhe)
H. Platzer (Inst. f. Nachrichtentechnik, T U München).

Programm-Ausschuß: Niemann (Erlangen), Triendl (Oberpfaffenhofen), Paulus (Braunschweig), Foith (Karlsruhe), Winkler (Karlsruhe)

Die Teilnehmerzahl ist auf 120 Teilnehmer beschränkt. Die zeitliche Reihenfolge der Anmeldung entscheidet über die Teilnahme. Die Teilnahmegebühr beträgt DM 100.- einschließlich Tagungsband, Maschinenbenutzung und Erfrischungen.

5. 2.DAGM-Symposium

Das 2. DAGM-Symposium findet in der ersten Junihälfte 1979 am Institut für Informationsverarbeitung in Technik und Biologie (IITB), Fraunhofer-Gesellschaft, Karlsruhe statt (ehemals IITB-Kolloquium "Mustererkennung"). Als Thema ist vorgesehen "Angewandte Szenenanalyse--Segmentationsverfahren, Formanalysen und Verfolgungsalgorithmen". Nähere Einzelheiten werden später bekannt gegeben.

COLLOQUIUM ON MATHEMATICAL LOGIC IN PROGRAMMING

Debrecen, Ungarn, 10. - 15. September 1978

Dieses Colloquium befaßt sich mit Gebieten wie Semantics, Program Design Methodology, Problem Solving, Programming Tools Based On Logic, etc. Eine 300-Worte-Abstrakt wäre zum 15. April 1978 fällig gewesen.
Weitere Informationen durch
D. Gábor, Logic in Programming
Bolyai János Mathematical Society
P.O. Box 240
H-1368 Budapest, Ungarn

N A C H R I C H T E N

Sprache und Datenverarbeitung

"Anfang Dezember 1977 erschien im Max Niemeyer Verlag, Tübingen, das 1. Heft der schon länger angekündigten Zeitschrift Sprache und Datenverarbeitung. 2 oder 3 weitere Hefte werden bis 1978 folgen. Die Zeitschrift widmet sich allen Bereichen, in denen Sprache mit Hilfe von Datenverarbeitung bearbeitet wird. Sie bringt auch Aufsätze und Berichte, die für das Gebiet "Sprachverstehen" im Rahmen der künstlichen Intelligenz einschlägig sind. Ein weiteres Aufgabengebiet sind die Probleme der Sprachverarbeitung in Informations- und Dokumentationssystemen."
Herausgeber: Winfried Lenders, Bonn,
Harald H. Zimmermann, Regensburg.
Erscheinungsweise: 2 Hefte jährlich, je ca. 100 S., DINA4.
Preis: "Jahrgang DM 48,-; Einzelheft DM 24,-. Bestellungen zum Sonderpreis für Mitglieder der GI von DM 38,- bzw. DM 19,- werden an den Verlag erbeten."

Announcement

Professor Nicholas V. Findler, of the State University of New York at Buffalo, will participate in the teaching and research activity of the Institut für Informatik, Universität Karlsruhe in June and July this year. He would be glad to visit other universities and give colloquia there during the above time period. The following lecture topics are offered for selection:

What are Artificial Intelligence and Heuristic Programming about -- Objectives, Methodology, Solved and Outstanding Problems.

On the Role of Exact and Non-Exact Associative Memories in Human and Machine Information Processing.

Some Novel Approaches to Machine Learning.

An Interactive Environment for the Simulation of Several Robots Which Can Learn, Plan Their Actions and Co-Exist.

Studies in Machine Cognition Using the Game of Poker.

The Problems of Time, Retrieval of Temporal Relations, Causality and Co-Existence.

On the Complexity of Decision Trees, the Quasi-Optimizer, and the Power of Heuristic Rules.

On a Heuristic Search Strategy in Associative Networks.

Toward Analogical Reasoning in Problem Solving by Computers.

MARSHA, the Daughter of ELIZA -- A Simple Program for Information Retrieval in Natural Language.

A Heuristic Information Retrieval System Based on Associative Networks.

A Similarity Measure between Strings.

A "Universal" Puzzle Solver.

A Few Steps toward Computer Lexicometry.

Please write to him now at:

Department of Computer Science
State University of New York at Buffalo
4226 Ridge Lea Road
Amherst, NY 14226; USA

His address in the Bundesrepublik will be:

c/o Prof. Dr. A. Schmitt
Institut für Informatik I
Universität Karlsruhe
75 Karlsruhe 1
Zirkel 2
Postfach 6380
Tel.: (0721) 608-3965

NEW RUTGERS/UCI LISP SYSTEM

Date: 10 Apr 1978 (Monday) 1521-EST
From: LEFAIVRE at RUTGERS-10
Subject: NEW RUTGERS/UCI LISP SYSTEM RELEASED
To: @LISP.NET[3442,46] at RUTGERS-10:

Greetings!

The new version of RUTGERS/UCI LISP is (finally) finished. I really want to apologize for taking so long - especially to those of you who have sent me tapes. It was a case of thinking I was "just around the corner" for several months, while bugs kept cropping up and new features were being added. Anyway, things are now stabilized - the system has been extensively debugged here at Rutgers, and everything looks fine. In the process of testing out some new code in the compiler (having to do with compiling MAPs in a better way), I discovered a deeply-embedded bug which took a good deal of time to excise. It had evidently been around for years, and was particularly unpleasant because it didn't cause a compiler error, just slightly screwy code. Anyway, it has been fixed and the compiler seems to be in good shape.

If I already have a tape of yours, I will mail it out tomorrow. If I don't have a tape and you want to get the new system, send me one. I don't know how feasible it is to send large files over the net, but if you want to try, I understand that Rutgers doesn't require you to log in to retrieve files. Everything is in LSP: [11,10] - the system is now called LISP, and at a minimum you need LISP.EXE, LISPC.EXE, LISP.LOD, and LISP.SYM. The Rutgers portion of the documentation (which has been updated extensively) is in LSP:RUTLSP.MAN. You might start by retrieving this document and checking out all of the new features. In addition to the new stuff, there have been a number of (mostly minor) bug fixes, and several outright changes:

- The comment atoms have been changed from *, **, and *** to ; (print to the right) and ;; (print at current indentation level).
- *SORT, *MERGE, and *INSERT are gone (SORT, MERGE, and INSERT now take optional arguments).
- LEXORDERCAR is gone (LEXORDER takes CARs until atoms are reached).
- CURRCOL is now called CHRPOS. Note that CHRPOS and CHRCT now work properly even when the last READ was terminated with an ESCAPE.
- All of the MAP functions now take their CDRs before applying the function (the compiler used to do it one way, the interpreter the other), so one can map over a list and do RPLACDs if desired.
- The QUOTE macro character is now ' (although @ will also be recognized).
- The initialization files are LISP.INI, LISPC.INI, etc.
- The default number base is decimal, with *NOPOINT=T (but *NOPOINTDSK remains NIL).
- The "ignore rest of line char" remains ^Y, but it is only used for files (i.e., it is not sent to terminals).

A conversion file, LSP:CONVRT.LSP, is available for converting existing files with comments, CURRCOLs, etc., to the new conventions, so it should be quite painless. Because of some changes in the compiler, however, existing LAP files should be recreated (to sweeten this prospect for you, the new code should take up slightly less space than the old code because of some new optimizations in the compiler).

As usual, questions or comments to LEFAIVRE@RUTGERS-10.

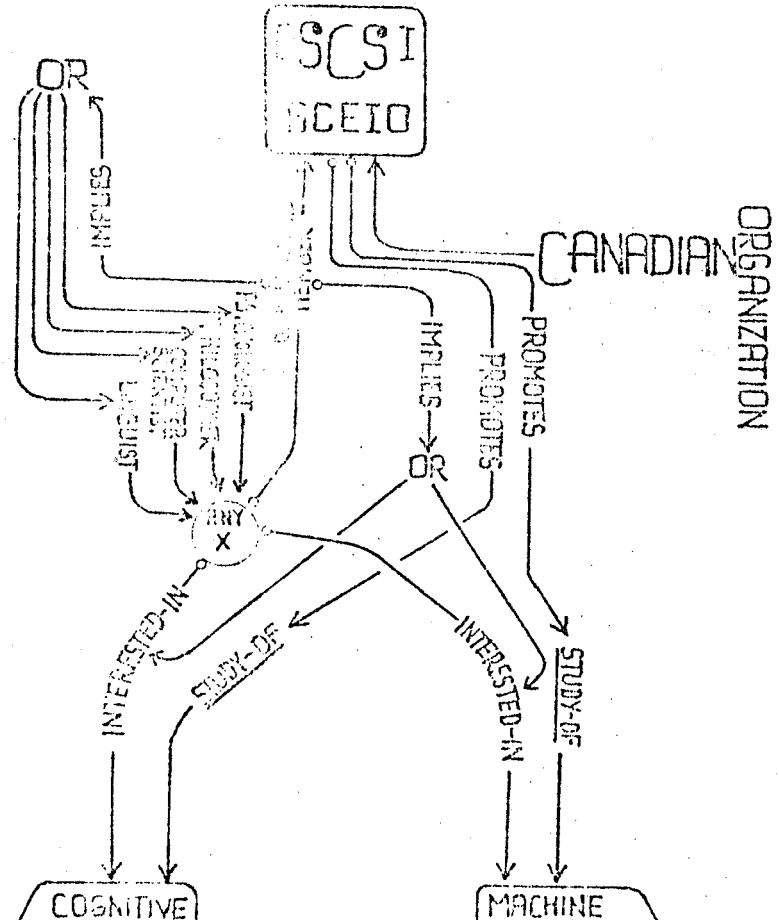
Anfragen an R. Lefaivre, Dept. of Computer Science
Rutgers University
Hill center
New Brunswick, N.J. 08903 USA

.1/no.1

December 1977

SIGCSIS NEWSLETTER

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



Herausgegeben am Dept.of Computer Science, University of Alberta Edmonton, Canada T6G 2H1, von Len Schubert u. Jeff Sampson

Arbeitstagung "Kognitive Psychologie" vom 2.-5.4.1978 in Hamburg

Veranstalter: Prof.Dr.Detlef Rhenius
Dipl.Psych.Hans Ueckert

Die Tagung war die erste ihrer Art und hatte zum Ziel, möglichst alle jene Personen aus dem deutschen Sprachraum miteinander bekannt zu machen, die an der Untersuchung kognitiver Prozesse auf der Grundlage nicht-numerischer Modellbildungen arbeiten. Unter Kognitiver Psychologie ist hier also jene neue Ausrichtung dieser Disziplin zu verstehen, welche die Voraussetzungen für und die Vorgänge bei der komplexen menschlichen Informations-Verarbeitung zum Thema hat. Eine wichtige Forschungs-Methode dazu ist die Computer-Simulation.

Das Programm der Tagung war im letzten KI-Rundbrief abgedruckt. Es gliederte sich in 6 Themenbereiche, zu denen es 40 Referate gab:

I. Kognitive Psychologie als integrativer Bestandteil psychologischer Grundlagenforschung.
(Einführung und Diskussionsleitung: Friedhart Klix, Berlin)

II. Kognitive Organisation der menschlichen Informations-Verarbeitung.
(Einführung und Diskussionsleitung: Dietrich Dörner, Gießen)

III. Sprachliche Kognition und sematisches Gedächtnis.
(Einführung und Diskussionsleitung: Hans Aebli, Bern)

IV. Entscheidungstheoretische Ansätze zur Kognitiven Psychologie.
(Einführung und Diskussionsleiter: Detlef Rhenius, Hamburg)

V. Informationelle Produktionssysteme und Computer-Simulation.
(Einführung und Diskussion: Hans Ueckert, Hamburg)

VI. Anwendungsfragen der Kognitiven Psychologie.
(Einführung und Diskussion: Michael Stadler, Münster)

Das Programm, das gegenüber der Ankündigung im letzten KI-Rundbrief nur unwesentliche Veränderungen aufwies, hatte zwar den Charakter einer Vortragsreihe, aber in den 15-minütigen Diskussionen, die sich jeweils den Vorträgen anschlossen, gab es enga-

gierte Auseinandersetzungen, die zum Teil sehr grundsätzlichen Charakter hatten. Dies begann gleich nach dem ersten Vortrag, wo es um die Auseinandersetzung zwischen Behaviourismus und kognitiver Psychologie ging und wo der Diskussionsleiter durch seinen hervorragenden Sachverstand für einen schwungvollen Einstieg in die Thematik sorgte.

Alle Referate der Tagung (einschließlich der Einführungs-Referate) werden in einem Kongreßbericht veröffentlicht, der von Hans Ueckert und Detlef Rhenius herausgegeben wird. Titel und Verlag stehen bisher noch nicht endgültig fest.

Die Tagung hatte etwas über 200 Teilnehmer, die aus der Bundesrepublik Deutschland, der DDR, Österreich, der Schweiz und aus den Niederlanden kamen. Am Schluß der Veranstaltungen wurde von vielen Teilnehmern der Wunsch geäußert, diese Tagung in festem Rhythmus zu wiederholen, eventuell in einer etwas geänderten Organisationsform. Auf einer kurzen Schlußbesprechung einigten sich die daran interessierten Teilnehmer, bis zum Kongreß der Deutschen Gesellschaft für Psychologie, der im September 1978 in Mannheim stattfindet, Themenbereiche für eine neue Tagung zu formulieren, die eingegrenztere Fragen behandeln sollen, als es jetzt möglich war. Es gab auch eine vage Zusage eines möglichen Veranstalters der nächsten Tagung "Kognitive Psychologie", der sich aber erst auf dem Kongreß in Mannheim festlegen will.

Hamburg, am 18.4.1978

Rhenius

(Prof.Dr.Detlef Rhenius)

Artificial Intelligence for Dijkstra

In a paper for the IC877/ACM conference, Belgium April 77, Dijkstra looks back on the evolution of programming during the last eight years. The title of his paper: "Programming: from craft to scientific discipline" contains his main point. Its purpose is to propagate his ideal that every program should be accompanied by a correctness proof, or more specifically: that every program should emerge from its correctness proof.

Dijkstra has to admit that reality is different and he looks around for the obstructing forces that block the realization of his dream. He locates one force in Floyd's 1967 paper: " Assigning meanings to programms " and sadly writes:

> "..., Floyd's paper had immediately been associated with mechanical verification --or even: discovery-- of formal proofs of the correctness of programs. As a sad result, Programming Methodology has for quite some years been in danger of being killed in its youth by the superstition that underlies so much of the Artificial Intelligence activity, viz. that every thing difficult is so boring that it had better be done mechanically."

Meltzer who lectured the day before Dijkstra on 'Brains and Programs' found this a peculiar view of A.I., and ended his presentation, after apologizing for commenting prematurely, with "Everything difficult is so exciting that it would be marvellous if one were able to understand with the precision of a computer program what was really going on."

Dijkstra, continuing his search for obstructing forces, locates another one in a lack of mathematical knowledge of current programmers and subsequently writes:

> "But whenever a craft is replaced by a scientific discipline, the old members of the guild feel themselves threatened, and quite understandably so. ... today's "programming guild" encompasses --depending on how we count-- between 500,000 and 1,000,000 people, for the majority of whom it is totally unrealistic to expect that they can still acquire a scientific attitude.

> For them the recent developments in programming poses a serious problem, and their existence presents a serious barrier to the more wide-spread adoption of the newer programming techniques."

After having introduced us to this vast mass of superfluous and inferior people Dijkstra attributes to them some bad intentions, adds some Freudian denials and suspects them of wanting to escape under cover of A.I., writing:

> "The conclusion that successful computer programming will eventually require a reasonable amount of scientific education of a rather mathematical nature is not too welcome among the guildmembers: they tend to deny it and to create a climate in which "bringing the computer back to the ordinary man" is accepted as a laudable goal, and in which the feasibility of doing so is postulated, rather than argued. (This is understandable, because its infeasibility is much easier to argue). They create a climate in which funds are available for all sorts of artificial intelligence projects in which it is proposed that the machine will take over all the difficult stuff so that the user can remain uneducated. I must warn you not to interpret the fact that such projects are sponsored as an indication that they make sense: the fact of their being sponsored is more indicative for the political climate in which this happens."

Thus Dijkstra's investigation brings him to the conclusion that there is a lack of climate control which hampers the one and only right point of view. At the same time he launches an odd idea about the aims of A.I.: "take over all the difficult stuff so that the user can remain uneducated". One of the aims of A.I. now is to develop more 'higher' languages and to develop interpreters for subsets of natural language in order to facilitate interaction with computers. So the really difficult stuff may become accessible and the group of people that can have access to computers will again be expanded. Another interesting application will be flexible teaching machines. However, A.I. certainly does not aim at the user remaining 'uneducated'.

Up to this last quotation one may object to Dijkstra's opinions but at least one can follow his pattern of reasoning. Continuing, Dijkstra claims to have good connections with the future:

> "I am convinced that all these projects will fail, and that, the more ambitious they are, the more miserably they will do so. Hence I consider these projects as rather foolish, and for programming as a scientific discipline worth teaching, as rather harmless rearguard actions. As a bit pathetic, even.
>
> In the meantime these projects can still do a lot of harm. They can do so by their false promises, pretending that the sophistication of their future systems, combined with decreasing hardware costs, makes it economically attractive to forsake our educational obligations to the next generation. Needless to say, falling into that seductive trap would be the cultural blunder of the decade."

Having gone thus far off the road of emotionless argument and fighting heroically against self-generated nonsense ("pretending ... generation") it is only a small step for Dijkstra to add a handful of paranoia and to continue and end his paper with:

> "In another respect I sometimes fear that the harm has already been done. There is a wide-spread folklore that in particular correctness proofs for computer programs are intrinsically so long, tedious, boring, uninteresting and prone to error, that the mechanization of their verification is a must. The assumption, however, is wrong: correctness proofs for programs can be --and should be!-- just as beautiful, fascinating and convincing as any other piece of mathematics. But the rumour to the contrary is constantly spread by the advertizing campaigns for the mechanical verification systems. The fact that the most outstanding feature of most artificial intelligence projects seems to be the heavy advertizing campaigns deemed necessary for this support, should instal into our minds a healthy mistrust and suspicion."

The programming technique as envisioned by Dijkstra with correctness proofs being the backbone of programming has, in my view, a chance to be adopted on a large scale, only when mechanical deductive and inference capabilities, as currently being developed by A.I., becomes available. At the same time there are doubts whether explicit correctness proof programming effectively solves the 'software crisis', as Dijkstra is tirelessly drumming, but I retrain from digging out these doubts in this context.

The current A.I. community still suffers from the discrepancy between its achievements and the expectations that were evoked by its predecessors to obtain funds for a new scientific field, competing with a multitude of well-established disciplines. And possibly the unrealistic expectations were induced by imprecise, ambigous aims, formulated in anthropomorphic terms. Dijkstra has not yet observed that these days A.I. projects are phrased overly carefully to prevent raising expectations that are not intended at all.

At the end of the sixties the Software Managers were confronted with the law that a large program system contains a constant number of bugs. Dijkstra then created high expectations and seduced them with 'Structured Programming' as the way out. Eight years later, explaining away the lack of concrete success by declaring the majority of programmers as uneducated, sounds grotesque. Now giving the advice to Software Managers to look for beauty, to be sensitive to fascination and to trace their convincedness, becomes ridiculous. Calling programming a scientific discipline and aspiring up to the high standing of mathematics, we consider misleading, and it may be interpreted as a frenzied, cover up operation.

Researchers that have to deal with problems of noisy, mutilated, complex input, e.g. speech recognition and picture analysis, are experimenting with programming techniques that can briefly be characterized as 'making the best of it'. Programs written using these techniques, have the interesting property of being robust in the sense that removal of code degrades their behavior only gradually. It may be worthwhile to investigate whether these techniques can have a wider application.

Dennis de Champeaux
University of Amsterdam

BÜCHER - BERICHTE

Automated Theorem Proving: a logical basis, von D.W. Loveland
North-Holland P, C., 1978
(eine Besprechung im nächsten Rundbrief ist vorgesehen)

Artificial Intelligence, von P.H. Winston, Addison-Wesley 1977.

Harold Boley, Fachbereich Informatik, Univ. Hamburg, Schlüterstr.70, 2 Hamburg 13

THE FIT LANGUAGE FOR ARTIFICIAL INTELLIGENCE (IFI-HH-M-55/78)

Abstract

The new AI language FIT is presented, which attempts to unify, generalize, and advance a number of important programming language concepts. The single non-atomic expressions of FIT are fitments, which fit fitters to fittees in unification of applications, matches, and assignments, which apply, match, and assign functions, patterns, and variables to arguments, instances, and values respectively. Fitments themselves - not only atoms - are used as variables, so that function and consequent theorem definitions become assignments of bodies to invocation patterns, name-directed and pattern-directed invocations are unified, and computation, like look-up, becomes valuation. Similarly, many-eyed demons are defined to watch an arbitrary number of fitments; in particular, generalized types ask for arbitr-ary relationships between variables.

An Investigation into the Goals of Research in Automated Theorem Proving as Related to Mathematical Reasoning, von F.M. Brown, Dept. Art. Intell. Report 49, Edinburgh, März 1978.

We argue that theorem provers based on domain dependent knowledge are not the kinds of deductive systems that are needed as a component of a mathematical reasoning system. The reason for this being that such systems are not extensible in that they cannot assimilate and use new deductive knowledge produced by a mathematical reasoning system.

Algorithm Analysis through Synthesis, von K. Clark u. J. Darlington, Imperial College, Dept. Comp. & Control, London, Oct. 1977

Automatic construction of Algorithms and Data Structures Using a knowledge base of Programming Rules, von D.R. Barstow, Stanford AI-Memo 308, Nov. 1977.

Despite the wealth of programming knowledge available in the form of textbooks and articles, comparatively little effort has been applied to the codification of this knowledge into machine-usable form. The research reported here has involved the explication of certain kinds of programming knowledge to a sufficient level of detail that it can be used effectively by a machine in the task of constructing concrete implementations of abstract algorithms in the domain of symbolic programming.

J. Darlington:

PROGRAM TRANSFORMATION AND SYNTHESIS: PRESENT CAPABILITIES

Dept. Comp.& Control, Rep. Nr 77/43, Imperial College, London Sept. 77.

Abstract

This paper concentrates on the practical aspects of a program transformation system being developed. It describes the present performance of the system and outlines the techniques and heuristics used.

The design of efficient data representations, J. Darlington, Imperial College London, Dec. 77.

It is a recognised programming discipline to first approach a task at an appropriate level employing high level data types and then to design appropriate structures that will represent these data types efficiently in the computer. For example priority queues can be efficienty implemented as bionomial trees, Vuillemin (1976), or trees as vectors, Floyd's treesort (1964). In this paper I would like to show how consideration of the computations being performed at the higher level can assist in the design of suitable data representations. The programming language and manipulation rules I will use in this paper are based on those outlined in Burstall and Darlington (1977) and some similarity with that paper will be assumed.

Program transformations involving unfree data structures, An extended example, J. Darlington, Imperial College, London, Oct. 77. Rep. Nr. 44.

requested x

Fuzzy sets and artificial intelligence, Prof. Dr. D. Lazak, Dept. of Organization and Management, University of Groningen, Netherlands

Abstract

Digital information processing used in todays MIS and process control systems has so far not been able to generate a fully self contained problem solving system cycle including adaptable and intelligent mind fields as central decision base. This task of intelligent decision making is left to human elements within those problem solving systems thus generating many failures and dangerous situations for the involved system components. This paper shows basicly how those central intelligent decision bases have to be constructed logically and physically in order to overcome the biological limitions of intelligent decision power. The most remarkable result of this investigation is that a quantitative description of the general problem solving cycle including intelligent mind field elements is possible and that this cycle is invariant against the hardware realization of the problem solving system i.e. this cycle including intelligent mind field modules (which will be described quantitatively) is valid for systems of artificial intelligence in the same way as it is valid for systems of natural (biological) intelligence. It can be shown that regarding the intelligent mind field modules a border between those modules and biological intelligence modules does not exist any more because both are reduceable to quantum cybernetical process elements.

Inference rules for program annotation, N. Derschowitz, Z. Manna, Stanford AI-Memo 303

Methods are presented whereby an Algol-like program, given together with its specifications, can be documented automatically. The program is incrementally annotated with invariant relationships that hold between program variables at intermediate points in the program and explain the actual workings of the program regardless of whether the program is correct. Thus this documentation can be used for proving the correctness of the program or may serve as an aid in the debugging of an incorrect program.

Synthesis: Dreams = Programs, Z. Manna, R. Waldinger, Stanford AI-Memo 302

Deductive techniques are presented for deriving programs systematically from given specifications. The specifications express the purpose of the desired program without giving any hint of the algorithm to be employed. The basic approach is to transform the specifications repeatedly according to certain rules, until a satisfactory program is produced. The rules are guided by a number of strategic controls. These techniques have been incorporated in a running program-synthesis system, called DEDALUS.

Presburger arithmetic with bounded quantifier alternation, C.R. Reddy, D.W. Loveland, Dept.Comp.Sc., Duke Univ., Durham, N.Car. 27706, Febr. 1978.

A system for incrementally designing and verifying programs, vol. 1&2,M.S,.Moriconi, ISI/USC, 4676 Admiralty Way, Marina del Rey, CA 90291 USA, Report RR-77-65/66

Vol. 1 describes SID (System for Incremental Development), which is a computer system for incrementally designing and verifying large, complex programs. It executes commands, proposes actions, answers questions, and accepts and reasons about new or changed information. SID has three main, distinctive characteristics: (1) it provides several useful incremental capabilities, including the ability to respond to changes by ensuring that the final problem solution is consistent and by keeping intact still-valid work without complete reprocessing; (2) its user interface has the ability to guide the user through the design and verification, and to engage in an interactive English dialog about the potential effects of changes; (3) it supports a substantial programming language which includes features for generating run time checks, stating concurrent processes and shared data, and developing data abstractions (Approx. 125 pages)

Vol. 2 (appendix) contains a transcript of a session with SID in which a simple message switching network that allows secure, asynchronous message transfer among a fixed number of users is incrementally developed. (Approx 70 pages)

Meta-evaluation as a tool for program understanding, R. Balzer, N. Goldman, D. Wile,Report ISI/RR-78-69, Adr. wie oben

This report describes the technology used in a running system that embodies theories of program well-formedness and informality resolution established by symbolically executing the program to systematically discover the intended meaning of each informal construct within an informal specification.

Mutual Dependencies and some results on undecomposable relations, J.M. Nicolas, ONERA-CERT, B.P. 4025, F-31055 Toulouse Cedex Febr. 1978

In this paper the new concept of mutual dependency in database relations is introduced. Mutual dependency appears as a generalization of functional and multivalued ones. Furthermore, it provides a necessary and sufficient condition for a relation to be undecomposable into those of its projections. Then, attention is paid to more general decompositions and a necessary and sufficient condition for a relation to be undecomposable is proven.

FIRST ORDER LOGIC FORMALIZATION FOR FUNCTIONAL, MULTIVALUED AND MUTUAL DEPENDENCIES.

J.M. Nicolas

ONERA-CERT, Toulouse, France

The purpose of this paper is to show that first order logic is adequate for functional, multivalued and mutual

dependencies in relational data bases. Advantages of using logic instead of tailored formal systems are presented. This paper is decomposed into four sections. The first one presents some notions of logic and theorem proving which are relevant to this study. In the second section, a way to analyze data bases in terms of Logic is briefly indicated. The third section deals with the expression of dependency statements as formulas of Logic. Lastly section 4 is concern with some properties of dependency statements which follow directly from the proposed formalization.

A simplifier based on efficient decision algorithms, G.Nelson, D. Oppen, Stanford, AI-Lab., 1978

We describe a simplifier for use in program manipulation and verification. The simplifier finds a normal form for any expression over the language consisting of individual variables, the usual boolean connectives, the conditional function cond (denoting if-then-else), the integers (numerals), the arithmetic functions and predicates +, - and ≤, the LISP constants, functions and predicates nil, car, cdr, cons and atom, the functions store and select for storing into and selecting from arrays, and uninterpreted function symbols. Individual variables range over the union of the rationals, the set of arrays, the LISP s-expressions and the booleans true and false. The constant, function and predicate symbols take their natural interpretations.

The simplifier is *complete*; that is, it simplifies every valid formula to true. Thus it is also a decision procedure for the quantifier-free theory of rationals, arrays and s-expressions under the above functions and predicates.

Reasoning about recursively defined data structures, D. Oppen, Stanford Univ. ,AI-Lab., 1978

A decision algorithm is given for the quantifier-free theory of recursively defined data structures which, for a conjunction of length n, decides its satisfiability in time linear in n. The first-order theory of recursively defined data structures, in particular the first-order theory of LISP list structure (the theory of CONS, CAR, CDR), is shown to be decidable but not elementary recursive.

Deductive Question ansering on relational data bases,

R. Reiter Dept. Comp.Sc. U.of Brit. Columbia, Vancouver B.C. Canada

The principal concern of this paper is 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. The result is an approach to deductive retrieval which appears to be feasible for data bases with very large extensions (i.e. specific facts) and comparatively small intensions (i.e. general facts). More specifically, a suitably designed theorem prover "sweeps through" the intensional data base, extracting all information relevant to a given query. This theorem prover never looks at the extensional data base. The end result of this sweep is a set of queries, each of which is extensionally evaluated. The union of answers returned from each of these queries is the set of answers to the original query.

One consequence of this decomposition into an intensional and extensional processor is that the latter may be realized by a conventional data base management system. Another is that the intensional data base can be compiled using a theorem prover as a once-only compiler.

On closed world data bases, R. Reiter, Techn. Report 77-16, Dept. Comp. Sc., U. of B. C. , Vancouver, B.C. V6T 1W5, Canada

Deductive question-answering system generally evaluate queries under one of two possible assumptions which we in this paper refer to as the open and closed world assumptions. The open world assumption corresponds to the usual first order approach to query evaluation: Given a data base DB and a query Q, the only answers to Q are those which obtain from proofs of Q given DB as hypotheses. Under the closed world assumption, certain answers are admitted as a result of failure to find a proof. More specifically, if no proof of a positive ground literal exists, then the negation of that literal is assumed true.

In this paper, we show that closed world evaluation of an arbitrary query may be reduced to open world evaluation of so-called atomic queries. We then show that the closed world assumption can lead to inconsistencies, but for Horn data bases no such inconsistencies can arise.

AIMDS - User Manual, ed. by V.S.Sridharan, Dept. of Comp. Sc., Rutgers Univ. New Brunswick, NJ 08903, USA

This is the first and preliminary version of a user manual for the AIMDS system. It is being made available in the hope that attempts to use it in a variety of sample tasks will provide suggestions on ways of augmenting it, thus helping the system achieve some generality. At present many of the design decisions about what to implement first and how to implement have been guided strongly by the application of AIMDS to BELIEVER.

AISB/GI CONFERENCE

HAMBURG: 18-20 JULY 1978

The Conference Program includes:-

- REVIEWS - Programming Methodology *by BOB WIELINGA (Amsterdam)*
  - Representation of Knowledge *by DREW McDERMOTT (Yale)*
  - Vision *by BERT HORN (MIT)*
- SESSIONS include papers on:
  - Applications
  - Automatic Programming
  - Game Playing
  - Heuristic Programming
  - Information Processing
  - Programming Languages for AI
  - Programming Methodology
  - Psychological Models
  - Natural Language Systems
  - Theorem Proving
  - Vision
- DEBATES on:
  - The Implementation of Large AI Programs
  - The Use/Misuse of Heterarchical Control
- HOT-SPOT - For uptodate ideas
- TIME for informal discussions
- CONFERENCE DINNER
- *PRE-CIRCULATED CONFERENCE PROCEEDINGS*

For a provisional program together with a Conference- and a Hotel-booking form, please contact either:-

H.-H. Nagel
Institut für Informatik
Universität Hamburg
2 HAMBURG 13
Schlüterstraße 66-72
WEST GERMANY

or

Derek Sleeman
Dept of Computer Studies
The University
LEEDS LS2 9JT
U.K.

Reduced price group-travel is also available for US/UK participants travelling via London to Hamburg; please contact Derek Sleeman for details.

Those registering for the Conference before 1st June will be sent a copy of *The Proceedings* prior to the Conference. Also, it is advisable to book your accommodation early as there is considerable pressure on the hotels in the area around the University at Hamburg - for more details please see the Conference Program.