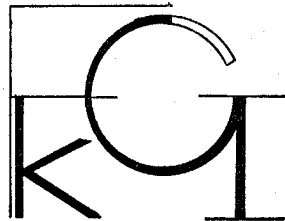


R U N D B R I E F
D E R
FACHGRUPPE KÜNSTLICHE INTELLIGENZ
IN DER GESELLSCHAFT FÜR INFORMATIK



Nummer 17

Juni 1979

INHALT

Bemerkungen des Herausgebers	1
Übersicht über KI-Projekte (zusammengestellt von P. Raulefs)	2
Kommentare zum SPIEGEL-Gespräch über Künstliche Intelligenz:	
J. Siekmann	10
P. Raulefs	19
Tagungsberichte	
E. Sánta-Tóth, M. Szöts (Budapest): Colloquium on Logic in Programming (Sátgótárján, Sept. 78)	20
P. Mosses / J. Loeckx 2nd Int. Workshop on Semantics of Programming Languages (Bad Honnef, 19.-23. März 1979)	27
Tagungsankündigungen	
Workshop on Formal Bases for Data Bases (Toulouse, Dez. 79)	56
Int. Summer Seminar on AI (Dubrovnik, Aug. 1979)	57
Neue Papiere	59
Sonstige Mitteilungen	62

I M P R E S S U M

Dieser Rundbrief der Fachgruppe "Künstliche Intelligenz" im Fachausschuß 6 ("Kognitive Systeme") der Gesellschaft für Informatik (GI) erscheint in unregelmäßigen Abständen (etwa vierteljährlich). Der Rundbrief wird den Mitgliedern der Fachgruppe kostenlos zugesandt. Mitglied der Fachgruppe wird jeder, der beim Herausgeber um Eintrag in die Adressenkartei nachsucht. Da die Versandkosten von der GI getragen werden, sollte eine Mitgliedschaft in der GI die Regel sein, nicht jedoch Voraussetzung für die Mitgliedschaft in der Fachgruppe "Künstliche Intelligenz". Aufnahmeanträge für die GI sendet der Herausgeber auf Anfrage gern zu.

Ziel dieses Rundbriefes ist, aktuelle Informationen unter den Mitgliedern der Fachgruppe auszutauschen. Der Herausgeber bittet daher alle Leser um möglichst rege Zusendungen von Beiträgen aus dem gesamten Gebiet der Künstlichen Intelligenz. Beiträge können z.B. folgendes behandeln:

- Kurzfassungen und Hinweise auf Veröffentlichungen und Berichte
- Beschreibungen laufender Projekte
- Diskussion wissenschaftlicher oder wissenschaftspolitischer Themen
- Berichte von Tagungen und Auslandsreisen
- Hinweise auf interessante Veranstaltungen, insbes. Tagungsankündigungen
- Offene Stellen, Stellengesuche

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: Peter Raulefs
Institut für Informatik III
Universität Bonn
Kurfürstenstr. 74
5300 Bonn 1
Tel. (02221) 73-5614/5611

Redaktionsschluß für die nächste Ausgabe: 1. September 1979

Bemerkungen des Herausgebers

Diese Ausgabe enthält erstmals eine aufgrund der früher versandten Fragebögen zusammengestellte Übersicht über KI-Projekte im Leserkreis des KI-Rundbriefes. Es ist beabsichtigt, diese Rubrik in zweierlei Weise fortzusetzen:

- (1) Laufende Ergänzungen: Gibt es in Ihrer Gruppe Änderungen der personellen Zusammensetzung bzw. haben Sie mit der Bearbeitung neuer Zielsetzungen begonnen, so teilen Sie mir dies bitte auf anliegendem Fragebogen mit.
- (2) Neueinträge: Ist Ihr Projekt bisher nicht in dieser Übersicht vertreten, so bitte ich Sie um Mitteilung auf anliegendem Fragebogen zur Aufnahme in den nächsten KI-Rundbrief.

Eine auf den neuesten Stand gebrachte Gesamtübersicht soll bei Bedarf (d.h. hinreichend vielen akkumulierten Neueinträgen/Änderungen) erscheinen.

Die recht erfreuliche Resonanz auf die Fragebogen-Aktion im Dezember 78 veranlaßt mich zu der Vermutung, daß die Trägheitsschwelle für die Zusendung von Mitteilungen durch Vordrucke gesenkt wird. Der anliegende Vordruck dient als Mehrzweck-Fragebogen auch zur Mitteilung über neue Papiere. Bitte machen Sie regen Gebrauch davon.

Nr. 17 der Zeitschrift DER SPIEGEL (23.4.79) enthält ein Spiegel-Gespräch "über künstliche Intelligenz" (so der Untertitel), das offenbar motiviert war durch eine ZDF-"Querschnitt"-Sendung mit einer Demonstration von Chess 4.6 und HAM-RPM. Obwohl solche Pressebeiträge keinen fachlichen Beitrag zu unserem Gebiet leisten, sollte man nicht unterschätzen, daß sich die Öffentlichkeit ihre Vorstellungen und Urteile über unser Gebiet notwendig ausschließlich aufgrund solcher Beiträge bildet. Da eben diese Öffentlichkeit unsere Arbeit finanziert und von evtl. Ergebnissen betroffen ist, sollten wir uns verpflichtet fühlen, an einer sachlichen und korrekten Informierung mitzuarbeiten.

Mir scheint, daß es für KI-Wissenschaftler zwei besondere Gründe hierfür gibt:

- (1) Das Vordringen der Datenverarbeitung in viele Bereiche unserer Gesellschaft löst aufgrund unklarer Vorstellungen über ihre Techniken Ängste aus, die zu emotionalen Abwehrreaktionen führt. Die Zielsetzungen der KI stellen dabei das Selbstverständnis vieler Menschen besonders stark in Frage (dies gilt schon durch die recht irreführende Bezeichnung "Künstliche Intelligenz").
- (2) Die Vorstellungen von Nicht-Experten über die Datenverarbeitung ordnen der KI häufig ein überproportional großes Gewicht zu. Ich halte es für wichtig, daß KI-Wissenschaftler die Beiträge ihres Gebietes in richtige Proportionen setzen zu den anderen Gebieten der Informatik. Es sollte dann klar werden, daß die KI nur ein Teilbereich der Informatik unter vielen ist. Es sollte zudem herausgearbeitet werden, inwiefern die KI auf anderen Gebieten aufbaut, und wann sie Ergebnisse zu andersartigen Fragestellungen beitragen konnte.

Ich schlage Ihnen vor, auch diese beiden Punkte bei der Beurteilung des Spiegel-Gesprächs zu berücksichtigen. Sie sind nachdrücklich dazu aufgefordert, Stellungnahmen einzusenden.

Der Tagungsband zur 4. Arbeitstagung Künstliche Intelligenz (Bad Honnef, Feb. 79) wird allen Teilnehmern kostenlos direkt zugesandt. Alle anderen Interessenten werden gebeten, den Band (vorwiegend aus erweiterten Kurzfassungen bestehend) gegen Erstattung der Selbstkosten auf dem anliegenden Formular zu bestellen.

P. Raulefs (Bonn)

ÜBERSICHT ÜBER KI - PROJEKTE

Zusammengestellt von P. Raulefs, Universität Bonn

Im Dezember 1978 wurde dem KI-Rundbrief ein Fragebogen beigelegt, in dem in der Künstlichen Intelligenz arbeitende Forschungsgruppen um folgende Angaben gebeten wurden:

- A. Bearbeitete Gebiete / Arbeitstitel des Vorhabens
- B. Mitarbeiter am Projekt (mit Leiter/Kontaktperson)
- C. Z. Zt. in Vorbereitung befindliche Papiere (mit Studien-, Diplom-, Dr.-Arbeiten)

Die folgende Zusammenstellung der aus den zurückgesandten Fragebögen entnommenen Angaben ist grob in die folgenden Gebiete eingeteilt:

1. Natürliche Sprachverarbeitung
2. Mechanisches Beweisen
3. KI-Programmiersprachen
4. Rechnergestützte Programmentwicklung und -analyse
5. Rechnergestütztes Problemlösen
6. Bildverarbeitung und Mustererkennung

Da die Zusammenstellung nur auf den zurückgesandten Fragebögen beruht, erhebt sie keinen Anspruch auf Vollständigkeit. Weitere Angaben dieser Art, insbesondere auch aktuelle Ergänzungen sind sehr willkommen und sollen in eine laufende Rubrik des KI-Rundbriefes aufgenommen werden.

1. NATÜRLICHE SPRACHVERARBEITUNG

1.1. Fachhochschule Berlin

Arbeitstitel des Projektes: Natürlichsprachlicher Zugriff auf eine Datenbank

Leiter: Prof. Dr. W. Brecht FB 2, Techn. Fachhochschule Berlin
Luxemburger Str. 10, 1000 Berlin 65

Papiere in Vorbereitung: (1) Windhaus. Prinzipien eines Chart-Parsers.
(2) Oehme. Eine Datenbankabfrage in einem relationalen Modell.

1.2. Universität Hamburg

Arbeitstitel: Simulation von intelligentem, natürlichsprachlichen Dialogverhalten und der damit verbundenen kommunikativen und kognitiven Prozesse.

- Leiter: Prof. Dr. W. v. Hahn Germanisches Seminar der Universität Hamburg
Von-Melle-Park 6, 2000 Hamburg 13.
Tel. (040)4123-4784/2566

Mitarbeiter: Dr. W. Hoepfner, Dipl.-Inf. W. Wahlster, A. Jameson
mehrere studentische Hilfskräfte

Dokumentation: Die Projektarbeit wird in einer laufenden Reihe von Berichten (bisher 7) und Memos (bisher 5) dokumentiert.

1.3. Universität Hamburg

Arbeitstitel: Natürlichsprachliches Frage-Antwort-System zur Szenenanalyse (SWYS - Say What You See) SIEHE AUCH UNTER 6.

Leiter/Kontaktpersonen: Dr. B. Neumann, Dr. P. Scheffé
Institut für Informatik der Universität Hamburg,
Schlüterstraße 66-72, 2000 Hamburg 13

Mitarbeiter: K.-J. Hanßmann, B. Pretschner, J. Rasche

Diplomarbeiten in Vorbereitung:

- J. Rasche. Implementation eines Szenenanalyseverfahrens in der Programmiersprache FUZZY (Betreuer: Neumann)
- B. Pretschner. Behandlung natürlichsprachlicher Quantifizierungen in Frage-Antwort-Systemen (Betreuer: Scheffé)
- K.-J. Hanßmann. Linguistische Repräsentation räumlicher Relationen in einem Szenenanalyse-System (Betreuer: Scheffé).

1.4. Batelle-Institut

Arbeitsgebiet: Natural language generation

Leiter/Kontaktperson: Dr. H. Marchand Batelle-Institut e.V., Postfach 900160,
6000 Frankfurt 90

Arbeit in Vorbereitung: Thèse d'Etat über "Natural language generation and its impact on tutorial systems"

1.5. Institut für Deutsche Sprache, Abt. LDV/FuE

SIEHE AUCH UNTER 2.3

Arbeitsgebiet: Informationssysteme mit natürlichsprachlicher Abfrage (A)
Logik von natürlichen Sprachen (B)

Leiter/Kontaktperson: Prof. Dr. D. Krallmann, Virchowstr. 54, 4300 Essen 1

Mitarbeiter: (A) Dr. G.L. Berry-Rogge, M. Kolvenbach, Dr. H. D. Lutz
(B) H. Wulz, Dr. G. Zifonum

Anschrift: Friedrich-Karl-Straße 12, 6800 Mannheim 1

Dokumentation: vgl. Projektbeschreibung im KI-Rundbrief Nr. 16 (März 1979)

(Projekt PLIDIS: Problemlösendes Informationssystem mit Deutsch als Interaktionssprache).

1.6. Universität Heidelberg

Arbeitsgebiet: Automatische Übersetzung

Leiter/Kontaktperson: Prof. Dr. K. Brockhaus Lehrstuhl f. Angew. Sprachwissen-
schaft, Landfriedstr. 12, 6900 Heidelberg

Mitarbeiter: Ch. Hauenschild, E. Huckert, R. Maier

Arbeiten in Vorbereitung: (Dissertationen)

Ch. Hauenschild. Automatische semantische Analyse des Russischen: Artikelre-
konstruktion.

E. Huckert. Automatische Synthese des Französischen aus einer log. Basis.

1.7. Universität Stuttgart

SIEHE AUCH 4.2,5.2,6.2

Arbeitsgebiet: Natürlich-sprachliche Interfaces

Kontaktperson: Dr. J. Laubsch Institut für Informatik der Univ. Stuttgart,
Azenbergstraße 12, 7000 Stuttgart 1

Mitarbeiter: A. Fauser, D. Rösner

Arbeiten in Vorbereitung: (Dissertationen)

A. Fauser. Dialog: Struktur und Computersimulation.

D. Rösner. Zur wissensbasierten Analyse von Texten in einem Dialogsystem.

1.8. Universität Erlangen-Nürnberg

Arbeitsgebiet: Analyse gesprochener Sprache

Leiter/Kontaktperson: Prof. Dr. H. Niemann IMMD/RRZE, Univ. Erlangen-Nürnberg,
Martensstraße 1-3, 8520 Erlangen

Mitarbeiter: G. Görz, W. Hein, Prof. Dr. G. Nees, P. Regel

Arbeiten in Vorbereitung:

G. Görz. Syntaktische und semantische Analyse natürlicher Sprache für ein Dialog-
system über einem definiten Gegenstandsbereich.

W. Hein. Softwaresysteme zur automat. Spracherkennung mit Hilfe paralleler Pro-
grammierung.

P. Regel. Optimierung der automat. Extraktion phonetischer Segmente aus kontinuier-
lich gesprochenen deutscher Sprache.

2. MECHANISCHES BEWEISEN

2.1. University of Texas at Austin

Arbeitsgebiet: Automatic theorem proving

Leiter/Kontaktperson: Prof. Woody Bledsoe Dept. of Computer Sciences, U of Texas,
Austin TX 78712, USA

Mitarbeiter: M. Ballantyne, P. Bruell, M. Lacey, M. Tyson

Arbeiten in Vorbereitung:

M. Ballantyne. Generation and use of counterexamples in analysis (Ph.D.)

W. Bledsoe. General Inequalities.

M. Ballantyne, D. Lankford. Complete sets of reductions

P. Bruell. An agenda driven theorem prover. (Ph.D.-thesis)

M. Tyson. An agenda driven theorem prover (Ph.D.-thesis)

2.2. Technische Universität Berlin

Arbeitsgebiet: Lernende Beweismaschinen

Betreuer: Proff. H.-J. Schneider, D. Siefkes, Dr. E. Konrad

Mitarbeiter: A. Müller Goerzallee 135, 1000 Berlin 45

2.3. Institut für Deutsche Sprache, Abt. LDV/FuE

SIEHE AUCH UNTER 1.5

Arbeitsgebiet: Automatische Beweisverfahren

Leiter/Kontaktperson: Prof. Dr. D. Krallmann, Virchowstraße 54, 4300 Essen 1

Mitarbeiter: W. Dilger

Dokumentation: vgl. Projektbeschreibung im KI-Rundbrief Nr. 16 (März 79)(PLIDIS)

2.4. Universität Karlsruhe

Institut f. Informatik I
Postf. 6380, 7500 Klrh 1

Arbeitsgebiet: Automatisches Beweisen

Untersuchung zur Einbeziehung mathematischen Wissens beim automatischen Beweisen am Beispiel der Automatentheorie

Leiter/Kontaktpersonen: Prof. Dr. P. Deussen, Dr. J. Siekmann

Mitarbeiter: N. Eisinger, M. Lutze, E. Unvericht, E. Vogel, Ch. Walter

Arbeiten in Vorbereitung:

P. Szabo. (Über Unifikationsprobleme) (Dissertation)

E. Unvericht. (Über Unifikationsprobleme) (Dissertation)

Ch. Walter. Induktionsverfahren im Mechanischen Beweisen (Dissertation)

E. Vogel. Unifikation von Morphismen (Diplomarbeit)

2.5. Universität Stuttgart

Arbeitsgebiete: (1) Beweise in der Algebra der binären Relationen

(2) Komplexität in der Algebra der binären Relationen

(3) Automatisches Erzeugen von binären Relationen

(4) Automatische Beweisverfahren

Leiter/Kontaktperson: Dr. W. Schönfeld Institut f. Informatik, Univ. Stuttgart,
Azenbergstraße 12, 7000 Stuttgart 1

Mitarbeiter: H. Dobeschinsky (4), G. Getto (2), U. Hedtstück (3), G. Illner (4),

W. Scholze (2), G. Seidel (3)

Arbeiten in Vorbereitung:

H. Dobeschinsky, G. Illner. Dialoginterpretierer zur Konstruktion und Verarbeitung boolescher Matrizen (Studienarbeit)

G. Getto. Relationentheoretische Modelle des Lambda-Kalküls (Diss.)

U. Hedtstück. Relative Lösbarkeit von Relationengleichungen (Diss.)

W. Scholze. Komplexitätsuntersuchungen zur Speicherung und Verarbeitung boolescher Matrizen (Dipl.)

W. Seidel. Lösungsmengen spezieller Relationengleichungen (Dipl.)

2.6. Universität Bonn

SIEHE AUCH UNTER 3.2, 4.1, 5.1

Arbeitsgebiet: Unifikationsverfahren

Leiter/Kontaktperson: Prof. Dr. P. Raulefs

Institut f. Informatik, Abt. III,
Univ. Bonn, Postf. 2220, 53 BN 1

Mitarbeiter: J. Darscheid, Ch. Peltason (Diplomanden)

Arbeiten in Vorbereitung:

J. Darscheid. Unifikation für absorptive Funktionen (Dipl.)

Ch. Peltason. Unifikation für schwach assoziative Funktionen (Dipl.)

P. Raulefs. Unification for collapsing equational theories.

2.7. Technische Universität München

SIEHE AUCH UNTER 4.4

Arbeitsgebiet: Automatisches Beweisen

Leiter/Kontaktperson: Dr. W. Bibel

Institut f. Informatik, TU München,
Postfach 202420, 8000 München 2

Arbeit in Vorbereitung:

W. Bibel. On matrices with connections

3. KI-PROGRAMMIERSPRACHEN

3.1. Universität Hamburg

Arbeitsgebiete: (1) Entwicklung und Implementierung einer KI-Sprache

(2) Interaktive Systeme

(3) Math. Logik, formale Systeme

Leiter/Kontaktperson: o. Prof. Dr. W. Brauer

Mitarbeiter: H. Boley, Prof. Dr. M. Kudlek, Prof. Dr. I. Kupka, M. Maaß

(keine Zuordnung zu Arbeitsgebieten angegeben)

Arbeit in Vorbereitung:

H. Boley. FIT - Eine neue KI-Sprache (Diss.)

3.2. Universität Bonn

SIEHE AUCH UNTER 4.1, ...

Arbeitsgebiet: Entwicklung und Implementierung der Programmiersprache CSSA

Leiter/Kontaktpersonen: Prof. Dr. P. Raulefs

Mitarbeiter: B. Augenstein, H.-L. Fischer, R. Glücker, A. Kerp, W. Oberdörster,

W. Pein, G. Kleinen, P. Ziegler.

Arbeiten in Vorbereitung:

B. Augenstein. Ein INTERLISP-Interpreter für CSSA.

H.-L. Fischer, P. Raulefs. Introduction to CSSA.

CSSA Programmer's Manual.

R. Glücker. Ein Übersetzer für CSSA (Dipl.)

A. Kerp. (Über Semantik paralleler Prozesse/Diss. vorhaben)

W. Oberdörster. Verifikationsverfahren für einfache parallele Actorsysteme (Dipl.)

G. Kleinen. Ein symbolischer Interpreter für CSSA (Dipl.)

- W. Pein. Interaktive Fehlerbehandlung für CSSA (Dipl.)
P. Ziegler. Eine Benutzerschnittstelle für CSSA (Dipl.)

3.3. Technische Universität Darmstadt

Arbeitsgebiet: Anwendung von Ergebnissen aus der KI beim Entwurf und der Benutzung von Programmiersystemen

Leiter/Kontaktperson: Prof. Dr. H.-J. Hoffmann Inst. f. Prakt. Informatik,
Mitarbeiter: R. Lutze TU Darmstadt, Steubenplatz 12,
6100 Darmstadt

Arbeit in Vorbereitung:

R. Lutze. Dissertation.

3.4. Universität München

Arbeitsgebiete: (1) Very-high-level languages
(2) Interactive programming
(3) Representation of knowledge

Leiter/Kontaktperson: Prof. Dr. Dr. D. Gernert Schluderstr. 2, 8000 München 19

Arbeiten in Vorbereitung: Z. Zt. 4 Dr.-arbeiten und einige Dipl.-/Studienarbeiten

4. RECHNERGESTÜTZTE PROGRAMMENTWICKLUNG UND -ANALYSE

4.1. Universität Bonn

SIEHE AUCH UNTER 2.6, 3.2, 5.4

Arbeitsgebiete: (1) Programmtransformationen
(2) Programmverifikation

Leiter/Kontaktperson: Prof. Dr. P. Raulefs Inst. f. Informatik, Abt. III
Postfach 2220, 5300 Bonn 1

Mitarbeiter: H. Brock, H. Eigenmeier, Ch. Knabe, K. Tramer (Diplomanden)

Arbeiten in Vorbereitung:

H. Brock. Ein Programmverifikationssystem für eine PASCAL-ähnliche Sprache (Dipl.)

H. Eigenmeier/Ch. Knabe/K. Tramer. Ein Übersetzer für abstrakte Datentypspezifikationen. (Dipl.)

4.2. Universität Stuttgart

SIEHE AUCH UNTER 1.7, 5.2, 6.2

Arbeitsgebiet: LISP-Entwicklungssysteme

Leiter/Kontaktperson: Dr. J. Laubsch Inst. f. Informatik, Univ. Stuttgart,
Mitarbeiter: K. Hess Azenbergstraße 12, 7000 Stuttgart 1

Arbeit in Vorbereitung:

Bauer. LISP-Compilierung.

4.3. Universität Stuttgart

SIEHE AUCH UNTER 2.5

Arbeitsgebiet: Rekursive Programmschemata

Leiter/Kontaktperson: Dr. W. Schönfeld

Mitarbeiter: A. Pasztor

Arbeit in Vorbereitung:

A. Pasztor. Programme als partielle Algebren (Diss.)

4.4. Technische Universität München

SIEHE AUCH UNTER 2.7

Arbeitsgebiet: Programmsynthese

Leiter/Kontaktperson: Dr. W. Bibel

Inst. f. Informatik, TU München
Postfach 202420, 8000 München 2

Arbeiten in Vorbereitung:

W. Bibel. Syntax-directed, semantics-supported program synthesis.

W. Bibel. Synthese und Analyse von Algorithmen (Vorl.skriptum)

5. RECHNERGESTÜTZTES PROBLEMLÖSEN

5.1. Universität Bonn

SIEHE AUCH UNTER 2.6,3.2,4.1

Arbeitsgebiet: Rechnergestütztes Problemlösen

Leiter/Kontaktperson: Prof. Dr. P. Raulefs

Adresse wie oben

Mitarbeiter: Ch. Eick, A. Knapp, U. Theis (Diplomanden)

Arbeiten in Vorbereitung:

Ch. Eick. Problemlösen mit Hyperplänen. Entwurf eines Problemlöserkonstruktions- und -testsystems (Dipl.)

A. Knapp, U. Theis. Entwurf und Implementierung eines Problemlöserkonstruktions- und Testsystems in der Programmiersprache CSSA (Dipl.)

5.2. Universität Stuttgart

SIEHE AUCH UNTER 1.7,4.2,5.

Arbeitsgebiet: Kognitive Aspekte des Problemlösens

Kontaktperson: Dr. J. Laubsch

Adresse wie oben

Mitarbeiter: Dr. G. Fischer

6. BILDERARBEITUNG UND MUSTERERKENNUNG

SIEHE AUCH UNTER 1.3

Arbeitsgebiet: Auswertung von Fernsehbildfolgen, insbesondere

- Segmentierung von digitisierten Bildern
- Bewegungsanalyse und 3D-Beschreibungen bewegter Körper
- Modellbildung und deren Einsatz bei der Analyse

Leiter/Kontaktperson: o. Prof. Dr. H.-H. Nagel

Inst. f. Informatik, Univ. Hambg.
Schlüterstr. 66-72, 2000 Hambg. 13

Mitarbeiter: L. Dreschler, B. Neumann, R. Radig, E. Tóth (als Gast)

6. BILDVERARBEITUNG UND MUSTERERKENNUNG

6.1. Universität Hamburg

SIEHE AUCH UNTER 1.3

Arbeitsgebiet: Auswertung von Fernsehbildern, insbesondere

- Segmentation von bewegten Bildern
- Bewegungsanalyse und 3D-Beschreibungen bewegter Körper
- Modellbildung und deren Einsatz in der Analyse

Leiter/Kontaktperson: Prof. Dr. H.-H. Nagel Adresse s. 1.3

Mitarbeiter: L. Dreschler, B. Neumann, R. Radig, E. Tóth (als Gast)

6.2. Universität Stuttgart

SIEHE AUCH UNTER 1.7,4.2,5.2

Arbeitsgebiet: Mustererkennung

Leiter/Kontaktperson: Dr. J. Laubsch Adresse s. 4.2

Mitarbeiter: K. Hanakata

7. SONSTIGE

7.1. Inst. f. linguistische Datenverarbeitung, Münsingen

Arbeitsgebiet: Wörterbuch der künstlichen Intelligenz (Englisch-Deutsch)

Leiter/Kontaktperson: Dr. H. E. Bruderer Inst. f. linguistische Datenverarbeitung,
Finkenweg 3
CH-3110 Münsingen, Schweiz

Bemerkungen:

1. Auf mehreren Gebieten arbeitende Gruppen sind unter den jeweiligen Kategorien mehrfach benannt und mit Querverweisen versehen ("SIEHE AUCH UNTER ...").
2. Die Reihenfolge der Nennungen soll keine (versteckte) Wertung enthalten. Vielmehr wurde eine ungefähre Nord-Süd-Reihenfolge eingehalten; Abweichungen beruhen auf Versehen.

Beitrag zum SPIEGEL-Gespräch
"Computer als Richter und Arzt?"
Hoimar von Ditfurth, Elektronik-
Experten und Schachmeister über
künstliche Intelligenz" (Nr.17, 23.4.79)

Die Theologen haben ihr Glockenläuten und
die Physiker haben ihr Lachen.

B. Brecht, Das Leben des
Galilei, Bild 9.

In der Nr. 17 vom 23. April 1979 trug DER SPIEGEL gewissen intellektuellen Strömungen der Zeit Rechnung und veröffentlichte ein Gespräch über KÜNSTLICHE INTELLIGENZ. Teilnehmer waren Klaus Brunnstein, Hoimar von Ditfurth, Frederic Friedel, Robert Hübner, Gisbert Jacoby, Fritz Küderli, Frieder Schwenkel, Horst Soboll und Spiegelredakteur Werner Harenberg. Stellen wir uns einmal vor, die SPIEGEL-Redaktion wäre ihrer journalistischen Sorgfaltspflicht nachgekommen und hätte erkannt, daß es sich bei der "KÜNSTLICHEN INTELLIGENZ" um ein Fachgebiet handelt, das zwar viele der Fragestellungen behandelt, die früher philosophischen Biertischgesprächen vorbehalten waren, diese Fragestellungen jedoch wissenschaftlich und insbesondere konstruktiv angeht: Kurzum, angenommen, die Herren wären nicht nur unter sich gewesen, sondern es hätte ein KI-Fachmann an dem Gespräch teilgenommen, der in diesem Fach- wie auch in jeder anderen akademischen Disziplin üblich - ein Fachstudium absolviert hat und, als weitere Annahme, auch auf diesem Gebiet noch akademisch arbeitet: Im nachfolgenden geben wir exklusiv für die Leser der KI-Nachrichten das Gespräch wieder, wie es dann hätte laufen können (normal gedruckt), wenn es nicht gelaufen wäre (*kursiv gedruckt*), wie es gelaufen ist.

SPIEGEL: Für 149 Mark verkauft das Versandhaus "Quelle" neuerdings einen sprechenden Kleincomputer, mit dem englische Vokabeln eingepaukt werden können. Die Schachcomputer, die in Kaufhäusern und Fachgeschäften zwischen 248 und 398 Mark kosten, sollen schon in wenigen Jahren weit spielstärker sein als heute. Für rund 2000 Mark gibt es Tischcomputer, die nicht nur rechnen und Daten speichern, sondern auch Karteien führen, spielen, zeichnen und musizieren. Dringt der Computer in die Wohnstuben vor, Herr Professor Brunnstein?

BRUNNSTEIN: Davon bin ich überzeugt. Sie erwähnen nur die ersten Ansätze. Der "Quelle"-Computer paukt nur 230 Wörter, hat eine schnarrende Stimme und ist wohl noch zu teuer. Die Schachcomputer kosten auch zuviel und spielen schwach, sie könnten schon heute wesentlich besser und billiger sein. Von den gängigsten Tischcomputern gibt es in der Bundesrepublik 10 000 Exemplare, in den USA sind allein im vorigen Jahr etwa 170 000 verkauft worden. Ich nehme an, daß wir hier wie bei jeder anderen Entwicklung zehn Jahre hinter Amerika zurück sind. Dort gibt es Tischcomputer in großer Zahl etwa seit 1975, also wird diese Welle bei uns Anfang der achtziger Jahre kommen.

SPIEGEL: Und wieviel werden die Heimcomputer dann kosten?

BRUNNSTEIN: Der gängigste, "Commodore 2001", kostet in Amerika 300 Dollar und bei uns

2 300 Mark. Der Preis könnte in Deutschland um ein Drittel niedriger sein, und ich nehme an, daß er auch dahin tendieren wird, wenn erst die Konkurrenz größer geworden ist und das Massengeschäft begonnen hat. Es wird florieren, wenn die Leute erfahren, welchen Freizeitwert und sonstigen Nutzen diese Rechner haben.

SPIEGEL: Nennen Sie uns einen besonders überraschenden Nutzen.

BRUNNSTEIN: Der Tischcomputer könnte, ergänzt durch wenig Zusatzelektronik, leicht so programmiert werden, daß er auf das Öffnen von Türen oder Fenstern reagiert, wenn das Haus leer ist, dann die nächste Polizeiwache anruft und meldet: "Im Haus Breite Straße 45 wird gerade eingebrochen, bitte schicken Sie einen Peterwagen." Ebensogut kann er bei einem Brand Feueralarm geben.

KI-MANN: Die Gefahr ist, daß wir die technologische Entwicklung, die auf uns zukommt, mit solchen Vorstellungen, die an sich richtig sind, verniedlichen: wenn es nur darum ginge, daß in Zukunft jeder seinen eigenen kleinen Heimcomputer zum Basteln und Schachspielen hätte, brauchten wir uns keine Gedanken zu machen und uns hier nicht zusammenzufinden. Das Problem ist jedoch, daß die Öffentlichkeit auf die tatsächlichen technologischen Veränderungen, die nach meiner Überzeugung von größerer Tragweite sein werden als vieles, was die "industrielle Revolution" bisher gebracht hat - obwohl sie

natürlich nur eine konsequente Fortsetzung des einmal beschrittenen Weges sind - daß die Öffentlichkeit also auf diese Veränderungen kaum bzw. gar nicht vorbereitet ist, obwohl wir einige der wenigen spektakulären Veränderungen bereits heute sehen.

SPIEGEL: Können Sie das etwas konkreter sagen?

KI-MANN: Lassen Sie mich die Antwort in 3 Gebiete aufteilen: die sozialen Veränderungen, die wir direkt einsehen können und die in bereits produktfertigen technologischen Entwicklungen ihre Ursache haben: ich denke da an den Druckerstreik wegen der Einführung der neuen Setztechniken. Ebenso bereits absehbar ist die Mechanisierung der Verwaltungsarbeit im staatlichen und privaten Bereich: ich könnte mir beispielsweise vorstellen, daß 80 bis 90 % der heute in der Verwaltung arbeitenden Beamten in wenigen Jahren nützlichere und interessantere Arbeit tun könnten. Ebenso in diesen Bereich fallen die Veränderungen, die durch den Einzug von Robotern in den Produktionsbereich bewirkt werden: ich denke da an die sogenannten Hand-Eye-Roboter, die selbständig Industrieteile sortieren, Lager verschrauben, Schweißnähte ziehen und einfache 'assembly-tasks' (z. B. eine Wasserpumpe zusammenzubauen) durchführen können. Diese Entwicklung wird jedoch bei weitem in den Schatten gestellt durch Produkte, die heute bereits machbar wären, der Abschluß der Grundlagenforschung jedenfalls zum Teil in Sicht ist, die jedoch noch nicht produktreif sind. Hierbei denke ich an medizinische Diagnose-Programme, automatische Richter, künstliche 'Mathematiker', den Computerpsychiater (oder Computerbeichtvater) ebenso wie an Computerprogramme, die gewisse militärische oder politische Entscheidungen selbständig treffen können. Es ist sicher nicht nötig, hier auf die sozialen und ethischen Konsequenzen solcher Technologie hinzuweisen (und wohl auch nicht, daß ich selbst diese Entwicklung bei weitem nicht nur positiv sehe). Wir werden uns daran gewöhnen, daß wir Produkte mit einer gewissen - je nach Geldbeutel - eigenen 'Intelligenz' kaufen werden: Autos, Waschmaschinen, Industrieanlagen usw. Um ein fiktives Szenario zu schildern: wir werden in Häusern leben, die ein 'Computergehirn' im Keller haben, das praktisch alle Funktionen des Hauses kontrolliert, mit angenehmer Stimme den Wetterbericht vorträgt, bevor wir das Haus verlassen, den Wagen bereits hat warmlaufen lassen, Türen und Fenster kontrolliert - und auch, wie Prof. Brunnstein bereits erwähnte, notfalls die Feuerwehr bzw. Polizei benachrichtigt.

Dieser Großcomputer im Keller wird auch einen mobilen Haushaltsroboter (d.h. also, einen Roboter, der über Fernsehaugen 'sehen'

und sich fortbewegen kann) kontrollieren, der einfache Verrichtungen im Haushalt ausführt: Staubwischen, Staubsaugen, gelegentlich an der Steckdose seine eigenen Batterien auffüllen, Geschirr wegräumen und spülen usw. Man wird den Bedarf nach gewissen Informationen über Zugverbindungen, Kochanleitungen, günstigen Zeitpunkten der Gartenarbeit usw. einfach im Wohnzimmer stehend in den Raum sprechen und je nach Bedarf schriftlich oder gesprochen erfüllt bekommen

SPIEGEL: Wenn ich Sie hier unterbrechen darf: Sie sind nicht ein wenig in's Schwärmen geraten, Herr Professor?

KI-MANN: Nun, wann und inwieweit wir ein solches Szenario haben werden, hängt natürlich von der weiteren politischen und wirtschaftlichen Entwicklung ab, nicht zuletzt natürlich auch davon, ob wir dies eigentlich wollen - ich kann nur sagen, daß es in nicht zu ferner Zukunft möglich wäre und jedenfalls heute in wesentlichen Zügen bereits machbar ist.

SPIEGEL: Sie erwähnten einen dritten Bereich.

KI-MANN: Dieser Fragenkomplex betrifft unser Selbstverständnis oder, um es ein wenig pathetisch mit Max Scheler zu sagen: 'Die Stellung des Menschen im Kosmos'. Die Tatsache, daß es Maschinen von dem Menschen vergleichbarer Intelligenz gibt bzw. geben wird, hat natürlich philosophische Konsequenzen, die in Deutschland nur ungenügend bzw. gar nicht zur Kenntnis genommen werden: ich denke dabei an die Bücher von Margaret Boden oder Aaron Slomon, deren philosophisches bzw. philosophisches Denken durch die KÜNSTLICHE INTELLIGENZ (Artificial Intelligence) profund beeinflusst wurde.

SPIEGEL: Kommen wir zu konkreteren Problemstellungen zurück:

Der Sprach-Lerncomputer "spricht" so wenig wie ein Tonband, denn er wiederholt immer nur dieselben Wörter. Aber es wird bald Computer geben, die wirklich wie Menschen sprechen und Gesprochenes verstehen können. Das übersteigt die Vorstellungskraft vieler Menschen, Herr Professor Schwenkel, wann wird es soweit sein.

SCHWENKEL: Schon in wenigen Jahren, denn in den Labors amerikanischer Universitäten gibt es den sprechenden Computer heute schon. Hier steht eine umwälzende Entwicklung bevor, die das Image der Computer vollständig ändern wird. Sie werden für viele nicht mehr die unheimlichen Maschinen sein, sondern zu einer Art Person werden.

SPIEGEL: Bereits vor etwa zwölf Jahren erregte ein Computerprogramm "Eliza" viele Amerikaner, weil es sich mit Menschen über deren Probleme unterhalten konnte. Sein Erfinder, Joseph Weizenbaum erlebte es, daß ihn Benutzer aus dem Zimmer schickten, weil

sie über ihre Intimsphäre mit dem Computer sozusagen "unter vier Augen" sprechen wollten.

BRUNNSTEIN: Daran läßt sich erlauben, was uns jetzt bevorsteht. Denn "Eliza" war ja ein geschriebener Dialog. Auf dem Bildschirm erschienen die Fragen des Computers, die Antworten mußten wie auf einer Schreibmaschine eingetippt werden. Und das stumme Gespräch war primitiv, "Eliza" nahm nur Wörter aus den Antworten auf und verwandte sie zu neuen Fragen.

SPIEGEL: Weizenbaum selbst nannte "Eliza" deshalb eine Schauspielerin, die sich aussern kann, aber selbst nichts zu sagen hat.

SCHWENKEL: Nun aber wird der Computer richtig sprechen, mit einer angenehmen Stimme, und er wird einen echten Dialog führen können, weil er das vom Gesprächspartner Gesagte im tieferen Sinne richtig verstehen kann.

SPIEGEL: Können Sie uns an einem Beispiel sagen, was möglich sein wird?

SCHWENKEL: Nun, Sie werden mit dem Computer so sprechen können wie mit einem Schalterbeamten bei der Bundesbahn. Ein solcher Dialog beginnt damit, daß Sie Ihr Reiseziel nennen und er endet damit, daß Sie mit der Fahrkarte weggehen. Alles, was dazwischen gesprochen werden muß, wird der Computer sagen können.

SPIEGEL: Geben Sie uns eine Prognose, wie es in acht oder zehn Jahren auf dem Hamburger Hauptbahnhof zugehen wird. Welche Funktionen werden noch von Menschen, welche werden von Computern ausgeübt werden?

SCHWENKEL: Wir werden dann nur noch ein paar Bahnpolizisten brauchen, alles andere werden die Maschinen erledigen.

SPIEGEL: Sie werden auch die Züge abfahren lassen?

SCHWENKEL: Selbstverständlich. Die rote Mütze kommt ins Museum.

SPIEGEL: Aber heute kommt mancher sogar mit den Fahrkartensautomaten nicht zurecht. Was geschieht, wenn eine Oma aus dem Bayrischen Wald auf dem Hamburger Hauptbahnhof dem Computer sagt, daß sie "an die Elbe" will und nicht weiß, wie sie fahren soll?

SCHWENKEL: Dann wird der Computer sich ein wenig mit ihr unterhalten, ihr ein paar Fragen stellen und ihr schließlich eine S-Bahnkarte nach Blankenese verkaufen und ihr natürlich auch noch den Bahnsteig sagen.

SPIEGEL: Welche Aufgaben, die nach Volksmeinung nur Menschen lösen können, werden heute schon Computern übertragen?

KÜDERLI: Ein anschauliches Beispiel ist der Computer im Unterricht. Sogar ein Dialog zwischen dem Computer als Lehrenden und dem Menschen als Lernendem ist keine Utopie mehr,

sondern in Anfängen schon heute Wirklichkeit.

SPIEGEL: Wo, zum Beispiel?

KÜDERLI: Beim "Pauken" von Lernstoff jeder Art, von simplen Vokabeln bis zu schwierigen mathematischen Aufgaben, kann der Computer den Lehrer sogar ersetzen. Man kann in einem unserer Programme auch schon die chinesische Schrift lernen, ohne daß es noch eines Lehrers bedarf.

SPIEGEL: Der Computer kann auch eindrucksvolle Experimente im Physik- oder Chemieunterricht simulieren. Aber kann er auch etwas tun, was ein Lehrer nicht tun kann?

KÜDERLI: Ich glaube nicht, daß ein Lehrer die Atomstruktur oder die Bahn von Satelliten so anschaulich beschreiben oder darstellen kann, wie es einem Computer möglich ist, der auch blitzschnelle Vorgänge - etwa wenn eine Kugel ein Gewehr verläßt - gleichsam in Zeitlupe vorführen kann.

KI-MANN: Gestatten Sie mir, daß ich hier unterbreche, andernfalls würden wir uns weitere 45 Zeilen über das vom Standpunkt der Computerintelligenz eher uninteressante Gebiet des Computerunterstützten Unterrichts unterhalten ...

KÜDERLI: (aufgebracht) ... wieso uninteressant?

KI-MANN: Entschuldigen Sie, ich meine natürlich nicht, daß dieses Gebiet an sich uninteressant ist - ganz im Gegenteil übrigens - die Ausgangsfrage des SPIEGEL war jedoch, welche Aufgaben heute schon dem Computer übertragen werden können, die nach Volksmeinung nur Menschen zu lösen in der Lage sind - und dazu gibt es aus der KI-Forschung nun sehr viel eindrucksvollere Beispiele (Herr Küderli verläßt den Raum in Richtung SPIEGEL-Bar): ich denke dabei etwa an DENDRAL, ein Programm, das in der Lage ist, Massespektrallinien, die zur chemischen Analyse erzeugt werden, zu analysieren und Hypothesen über die chemische Zusammensetzung des analysierten Stoffes zu generieren. Dieses Programm hat eine Performanz erreicht, die es den meisten menschlichen Chemikern weit überlegen macht und längst in vielen chemischen Labors im praktischen Einsatz ist.

Ein weiteres Beispiel ist MYCIN, ein Programm, das eine bestimmte Augenkrankheit diagnostizieren kann und eine erstaunliche Performanz erreichte, die es den meisten Medizinern überlegen sein läßt.

Ich denke ebenfalls an Programme beispielsweise, die selbständig mathematische Theoreme zu beweisen in der Lage sind und - trotz aller Skepsis die hier angebracht ist - selbst offene mathematische Probleme haben beweisen können.

SHAKEY, der mobile Roboter der Stanford-Gruppe, ist in der Lage, einfache Befehle in einem aus Blöcken und Zimmern bestehenden Szenario auszuführen und dabei auftretende Probleme zu lösen. FREDDY, der Roboter

der Edinburgh-Gruppe, ist in der Lage, ein Spielzeugauto selbsttätig wieder zusammenzubauen, wenn es vorher vor seinem (Fernseh-)Auge in die einzelnen Bestandteile zerlegt und zur Schulung einmal wieder zusammengesetzt wurde.

SPIEGEL: Wieso erfährt die Öffentlichkeit von diesen Dingen nichts - oder anders herum gefragt: wieso werden solche Roboter nicht eingesetzt?

KI-MANN: Ein Grund liegt darin, daß diese Roboter noch nicht bis zur Produktreife entwickelt sind und es noch immer beträchtliche technische - nicht prinzipielle - Probleme gibt. Der zweite Grund liegt in dem enormen Kapitalbedarf; um es mit einem Zitat von Prof. Raphael, SRI, zu sagen: "Gastarbeiter sind heute noch billiger". Ein Zitat übrigens, das mir wegen seines impliziten Zynismus zutiefst zuwider ist, die unserem System zugrunde liegenden Mechanismen jedoch hinreichend genau wieder gibt.

SPIEGEL: Worin würden Sie die Höhepunkte dieser bisherigen Forschungen sehen?

KI-MANN: Das ist schon deshalb schwierig zu sagen, weil natürlich jeder KI-Wissenschaftler die auf seinem Spezialgebiet erbrachten Leistungen für die bedeutsamsten hält. Ich könnte mir jedoch vorstellen, daß für den Außenstehenden die bereits erwähnten sprachverstehenden Programme zu den faszinierendsten Entwicklungen gehören: es fällt mir immer wieder schwer, zu glauben, daß ein mit der Wissenschaftsgeschichte vertrauter, geistig aufgeschlossener Mensch nicht spontane Begeisterung - und angesichts dieses technischen Potentials- auch tiefe Betroffenheit zeigt. Denken Sie beispielsweise an den einfachen Tisch, der heute im Deutschen Museum steht, auf dem Otto Hahn die erste Kernspaltung gelungen war. Trotzdem glaube ich ...

SPIEGEL: Wieso glauben Sie, daß solche wissenschaftlichen Leistungen nicht die entsprechende Resonanz finden?

KI-MANN: Nun die KI-Forschung tut sich leider in Deutschland immer noch sehr schwer, innerhalb der Informatik ihren Platz zu finden.

SPIEGEL: Glauben Sie, daß die Situation sich ändern wird?

KI-MANN: Wenn wir wieder die USA hier als Maßstab nehmen und den von Prof. Brunnstein bereits erwähnten Zeitfaktor von 10 Jahren berücksichtigen, bin ich sicher, daß sich auch bei uns einiges ändern wird. Nur könnte es dann, um Prof. Goos - einen der Mitbegründer der deutschen Informatik - zu zitieren, bereits zu spät sein, und wir wären in die hoffnungslose technologische Abhängigkeit von den USA geraten, die sich ein auf Export bedachter Industriestaat wie

Deutschland nicht leisten kann.

SPIEGEL: Unser KI-Mann erwähnte vorher den 'Computerpsychiater' und Programme zur medizinischen Diagnose: halten Sie, Herr Prof. Brunnstein, solche Dinge in Ansätzen tatsächlich für machbar?

BRUNNSTEIN: Mehr als das. Es gibt Systeme, in denen nicht nur Informationen von und über Patienten gesammelt werden, sondern bei denen der Rechner auch die Diagnose stellt. Es gibt sie meines Wissens nur in den USA, insbesondere in Boston. Ob der Rechner eines Tages auch zur Therapie fähig ist, wage ich nicht zu sagen. Da sind ja unter anderem noch Aspekte wie Gefühl und Zuspruch von Bedeutung.

Aber es ist in der Tat eine grundsätzliche Entscheidung notwendig, ob wir sie lapidare Rechtsfälle entscheiden, ob wir sie automatische Kriege führen lassen wollen. Das alles können sie heute schon, oder es ist prinzipiell binnen kurzem erreichbar.

SPIEGEL: Was heißt es, automatisch Krieg zu führen?

BRUNNSTEIN: Stellen Sie sich vor, ein Computerprogramm verfügt über alle notwendigen Daten hinsichtlich der militärischen Stärke verschiedener Länder sowie über bestimmte politische Wertungen und Zielsetzungen. Es sei ausserdem an die militärischen Instrumente angeschlossen, beispielsweise könne es den Abschub von Raketen steuern.

SPIEGEL: Bis hierhin bietet sich kein technisches Problem?

BRUNNSTEIN: Nicht das geringste. Es gibt sogar schon Ansätze dazu. So wurde die Entlaubungsaktion im Vietnamkrieg, als die Vietcong in Urwäldern entdeckt werden sollten, mit Computerhilfe strategisch vorbereitet.

Ich kann mir nicht nur vorstellen, daß militärische und politische Entscheidungen bis hin zu einer Kriegserklärung allzu stark von der Bereitstellung und Analyse der Fakten durch den Computer beeinflusst würden. Ich habe sogar die Sorge, daß dies in einigen militärisch und technologisch potenten Ländern schon heute geschieht.

SPIEGEL: Es besteht dann also die Gefahr, daß sich die Intelligenzprothesen und Denkezeuge, wie Sie die Computer vorhin genannt haben, insofern verselbständigen?

BRUNNSTEIN: Ich fürchte, ja.

SPIEGEL: Wird der Computer, je besser er sich nutzen läßt, um so stärker zum Schrecken werden für viele, die fürchten müssen, durch ihn von ihren Arbeitsplätzen vertrieben zu werden?

BRUNNSTEIN: Zweifellos. In den nächsten 30 Jahren wird es zu einer drastischen Reduktion des Anteils menschlicher Arbeit zugunsten der maschinellen Arbeit kommen. Das hat die Schattenseite, daß viele Arbeitsplätze ver-

nichtet werden, ohne daß in annähernd gleicher Zahl ähnliche Arbeitsplätze geschaffen werden. Es hat die positive Seite, daß frustrierende Arbeit von Maschinen erledigt wird. Und spätestens um die Jahrtausendwende wird das Hauptproblem der Gesellschaft die Bewältigung der Freizeit sein. Dann wird der kleine Computer in der Wohnung so selbstverständlich sein wie heute das Fernsehgerät.

SPIEGEL: Könnten Sie sich vorstellen, daß ein sprechender Computer zum therapeutischen Werkzeug wird, das man Nervenkliniken und psychiatrischen Zentren an die Hand gibt? In jeder Stunde könnten mehrere hundert Patienten von einem Computersystem behandelt werden, schielte der amerikanische Psychiater Kenneth Mark Colby von diesen Möglichkeiten.

KI-MANN: Machbar wäre dies, und es gibt in den USA ernsthafte Entwicklungsarbeiten in dieser Richtung. Natürlich wecken solche Arbeiten starke Emotionen in den meisten Menschen, beispielsweise nennt Weizenbaum in seinem Buch diese Forschung schlicht "unmoralisch".

SPIEGEL: Würden Sie denn den Einsatz von Computern für diese Zwecke befürworten?

KI-MANN: Ich halte diese Entwicklungen für eine konsequente Fortsetzung der Haltung, die auch heute vielfach - weniger perfekt - Behandlungsmethoden charakterisiert, und ebenso unsere Einstellung gegenüber psychisch kranken Menschen. Nun, wo der zusätzliche Einsatz des Computers möglich geworden ist, nur wegen des Computers von Unmoral zu sprechen, erscheint mir unangebracht.

Ich selbst würde solche Forschungen allerdings ablehnen, jedoch mit einem anderen Argument: nicht die Verwendung des Computers an sich erscheint mir verwerflich, sondern die Behandlungsmethode, die damit notwendig verbunden ist! Denken Sie an Behandlungsmethoden, wie sie heute in England und Italien vor allem praktiziert werden, bei denen der Psychiater mit den Patienten zusammen lebt, wo nach Möglichkeit die Patienten in ihrer vertrauten Arbeits- und Familiensituation belassen werden, und der Psychiater die 'krankmachenden' Umstände zu ändern versucht - solche aufgeklärten Behandlungsmethoden sind mit dem Computer nicht durchführbar.

SPIEGEL: Läßt sich das Zukunftspanorama der denkbaren Computer-Leistungen noch ergänzen, bevor wir über die Konsequenzen sprechen?

SOBOLL: Daß er zeichnen kann, ist schon auf etlichen Ausstellungen von Computergraphik gezeigt worden. Überraschender wird sein, daß er auch schon komponieren kann, zum Beispiel eine viersätzige Suite für Streichquartett.

SPIEGEL: Wie hört sich das an? Schaurig?

SOBOLL: Nein, sicher nicht. Wenn man die Regeln, nach denen Mozart seine Sinfonien komponiert hat, in die Sprache des Computers übersetzt und ihn Mozarts Gesamtwerk analysieren läßt, dann wird er auch Mozart kopieren und ähnlich wie Mozart komponieren können.

SPIEGEL: Das Ergebnis wäre dann eine weitere Sinfonie, analog dem Rezept etlicher Bestseller-Autoren: Lies fünf Bücher und mach das sechste daraus? Dann wird vielleicht, obwohl Verdi schon lange tot ist und "Falstaff" seine 26. und letzte Oper war, irgendwann eine 27. Verdi-Oper uraufgeführt?

SOBOLL: Sinfonien oder Opern werden Computer wohl in absehbarer Zeit noch nicht komponieren, aber kleine Stücke, auch Lieder oder Arien im Stil Mozarts oder Verdis traue ich Ihnen schon zu.

SPIEGEL: Sie auch, Herr Schwenkel?

SCHWENKEL: Durchaus. Ich würde diese Beispiele nicht auf die Musik beschränken. Auch in der Literatur müßte es möglich sein, Romane - etwa von Günter Grass - zu analysieren, ihre Stilelemente in Algorithmen umzusetzen und so den Computer zu befähigen, einen neuen Roman in Stile von Grass zu schreiben.

SPIEGEL: Wann könnte es soweit sein?

SCHWENKEL: Da bin ich überfragt. Sicher nicht in meiner Generation, vermutlich irgendwann am Anfang des nächsten Jahrhunderts.

SPIEGEL: Dann würde der alte Streit wieder aufleben, ob ein Kopist schöpferisch arbeitet oder nicht. Könnten Sie uns andere künftige Computerleistungen nennen, die einhelliger als schöpferisch gelten würden?

KI-MANN: Ich würde dazu die verschiedenen existierenden Programme auf dem Gebiet des 'problem solving' (Problemlösen) nennen. Ebenso glaube ich, daß das Finden eines mathematischen Beweises Kreativität erfordert, und die Tatsache, daß die existierenden Programme, die dies bewerkstelligen, relativ 'mechanisch' funktionieren, erscheint mir kein Gegenargument zu sein.

Außerordentlich beeindruckend in diesem Zusammenhang ist jedoch ein Programm von D. Lenat (USA), das im Hinblick auf mathematische Kreativität konzipiert wurde. Dieses Programm beginnt mit einem gewissen rudimentären Wissen der Mengenlehre und wird danach auf einen freien 'Kreativitäts-Trip' geschickt und war in der Lage, Teile der Zahlentheorie beispielsweise wiederzuentdecken. Die von dem Programm entdeckten Zusammenhänge waren absolut nicht trivial (z. B. den Satz der eindeutigen Primzahlzerlegung) und nebenbei dem Autor des Programmes selber unbekannt. Es dürfte ein für

allemaal den Existenzbeweis erbracht haben, daß Kreativität durchaus nicht jene nebulöse, nur dem Menschen vorbehaltene Eigenschaft ist, sondern ein rationalisierbarer Vorgang, der auf dem Computer realisiert werden kann.

SPIEGEL: Das klingt ja wirklich fast ungläublich - wieso erfährt man von diesen Dingen nichts?

KI-MANN: Lassen Sie mich ganz offen sagen, daß dies auch an Ihnen, also den Medien, liegt: es ist ja durchaus möglich, daß Sie unter normalen Bedingungen mit den Holmars vorlieb nehmen müssen, aber angesichts der delikaten Situation der KI-Forschung in Deutschland könnte ich mir durchaus vorstellen, daß eine Reihe von aktiven Wissenschaftlern aus wissenschaftspolitischen Gründen zu einem Gespräch mit Ihnen bereit gewesen wären (Prof. von Ditfurth verläßt den Raum in Richtung SPIEGEL-Bar).

SPIEGEL: Wie steht es im Schachspiel, der zur Zeit populärsten Beschäftigung der Computer? Herr Jacoby, nehmen Sie mal an, wir legen Ihnen, dem Bundesliga-Trainer des Hamburger Schachklubs, eine Partie vor, die ein Computer und ein gleich gut spielender Mensch ausgetragen haben, und sagten Ihnen nicht, wer Weiß und wer Schwarz hatte. Könnten Sie es am Stil erkennen?

JACOBY: Ich denke ja. Vom Großrechner "Cyber 176", der hier im SPIEGEL-Haus gegen den Vizeweltmeister Kortschnoi ehrenvoll unterlag, bis zu den hundert- oder sogar tausendmal schwächeren Schachcomputern in den Kaufhäusern haben alle Maschinen gemeinsam, daß sie je nach Art der Stellung unterschiedlich stark spielen.

SPIEGEL: Wann spielen sie stärker, wann schwächer?

JACOBY: "Cyber 176" vollbringt in taktisch komplizierten Stellungen außerordentliche Leistungen und ist dann einem Meister, gelegentlich sogar einem Großmeister, gleichwertig oder sogar punktuell überlegen. Aber in sogenannten toten oder ruhigen Stellungen, wenn also auf dem Brett wenig "geschieht", spielt er wesentlich schwächer. Ein guter Schachspieler hingegen kann dann abschätzen, welche längerfristigen latenten Möglichkeiten in der Stellung drin sind. Und auch im Endspiel ist der Großrechner oft, allerdings nicht immer, schwächer als in der Eröffnung oder im Mittelspiel. Auch spielt er bedenkenloser auf Angriff und auf Materialgewinn als gleichwertige Schachspieler es tun.

SPIEGEL: Und ähnliches gilt auch für die kleinen Schachcomputer in den Kaufhäusern?

JACOBY: Ja, nur daß die Schwächen dort weit ausgeprägter, die Stärken weit weniger eindrucksvoll sind. Der Großrechner spielt im Schnitt so gut, daß ein Schachmeister sich anstrengen muß, gegen ihn zu gewinnen. Die Warenhaus-Computer spielen im Schnitt so schlecht, daß ein guter Hobbyspieler eigentlich ohne große Mühe jede Partie gewinnen müßte. Und auch wer ganz schlecht steht,

sollte nicht aufgeben, sondern bis zum Matt durchhalten. Oft kommt es nämlich gar nicht zum Partieverlust, denn im Endspiel sind die meisten Warenhaus-Schachcomputer geradezu Stümper.

SPIEGEL: Gibt es neben dem Computerschach noch andere Möglichkeiten, die Entwicklung und die Potenz der Großrechner für jedermann anschaulich zu machen?

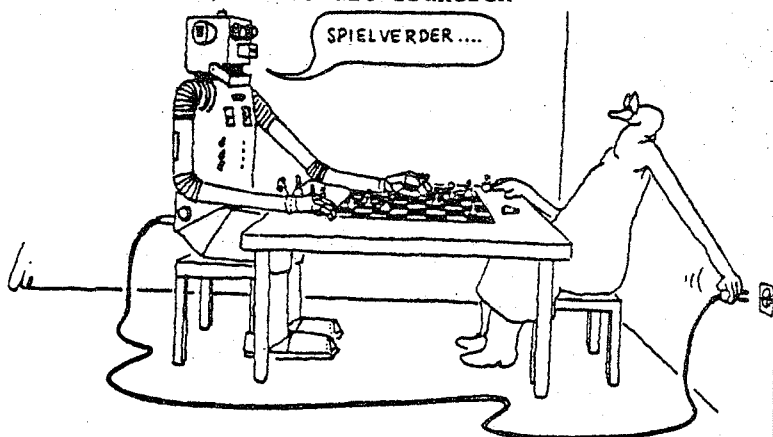
SCHWENKEL: Das wird sich zeigen müssen. Aber ich bin sicher, daß Computerschach weiterhin zu den besten Möglichkeiten gehören wird. Ein Schachprogramm kann die Leitungsfähigkeit eines Rechners etwa so beweisen, wie ein Automobilfabrik beweist.

SOBOLL: Der Vergleich gefällt mir.

KI-MANN: Mir aber nicht! (Dr. Soboll verläßt ent-rüstet den Raum in Richtung SPIEGEL-Bar).

SPIEGEL: Wie bitte?

KI-MANN: Es tut mir sehr leid, Ihnen nun schon wieder in die Parade fahren zu müssen, aber das Schachspielen mit dem Computer hat bei weitem nicht die zentrale wissenschaftliche Bedeutung für die KI, die die nächsten 630 Zeilen dieses Gespräches (wenn ich es nicht unterbrochen hätte) sowie die früheren



Veröffentlichungen des SPIEGEL den Leser glauben machen wollen (Herr Jacoby verläßt den Raum in Richtung SPIEGEL-Bar).

SPIEGEL: Als Gastgeber kann ich leider nicht ebenfalls den Raum verlassen; dürfte ich Sie bitten, dies doch ein bißchen zu modifizieren bzw. zu belegen?

KI-MANN: Dazu sollte man sich zunächst einmal überlegen, wieso eigentlich erwachsene Männer, denen im allgemeinen eine glänzende wissenschaftliche Karriere in einem etablierten Fachgebiet möglich gewesen wäre, sich auf eine so ver-gleichsweise kindische Sache wie Schachspielen einlassen (Dr. Friedel verläßt den Raum in Richtung SPIEGEL-Bar).

Die Antwort ist, daß das Schachspielen an sich diesen Wissenschaftlern natürlich auch gleichgültig ist und bestenfalls als Hobby von Interesse - dagegen ist es als relativ über-schaubares Paradigma für intelligente Prozesse hochwillkommen und insofern stellvertretend auch für andere intelligente Aktivitäten relevant.

BRUNNSTEIN, HÜBNER, SCHWENKEL: Das ist doch selbstverständlich und an anderer Stelle dieses Gespräches auch gesagt worden.

KI-MANN: Aber nicht mit der nötigen Nachdrücklichkeit! Insbesondere ist es durchaus möglich - vorausgesetzt Sie akzeptieren die These vom Paradigma - daß das Schachspiel doch keine so gute Wahl war, wie es scheinen mag: zur Beurteilung eines guten KI-Programmes sind im wesentlichen die Antworten auf zwei Fragen relevant: wie wird das Wissen des Gebietes, für das das Programm als Experte programmiert wurde, repräsentiert? Und zweitens: wie werden die nötigen Deduktionsschritte vorgenommen? Nun, im Computerschach wird (von der Eröffnung und dem Endspiel einmal abgesehen) das Wissen in der Evaluationsfunktion gespeichert und als Deduktionsschritte kann man vielleicht die Minimax (bzw. $\alpha - \beta$)-Strategie ansehen - beides Methoden die bekanntlich nur eine sehr begrenzte Anwendung finden können.

SPIEGEL: Können Sie das etwas verständlicher sagen.

KI-MANN: Dazu müßte man erst erklären, wie solche Spielprogramme funktionieren und dazu ist leider nicht genug Zeit: es kommt auf die technischen Einzelheiten hier auch nicht so an, als vielmehr darauf, daß die bisherigen Schachprogramme nach Methoden arbeiten, die nur sehr begrenzt anwendbar und im wesentlichen auch bereits erforscht sind. Angenommen die Hardwareentwicklung geht so weiter wie bisher und es gelingt in einigen Jahren ein Schachprogramm mit den bisherigen Methoden auf Weltmeister-Niveau zu bringen: das wäre - überspitzt gesagt - ein schwarzer Tag für die Schachforscher, denn dann hätte sich gezeigt, daß Schachspielen eben kein gutes Paradigma war, an dem auch komplizierte KI-Techniken erprobt und entwickelt werden können, ebensowenig wie beispielsweise NIM, KALAH oder DAME geeignet sind.

HÜBNER: (Bereits aufgestanden und im Gehen begriffen) Wenn Sie uns nicht unterbrochen hätten, wären wir auf die alternativen Möglichkeiten zur Schachprogrammierung eingegangen.

KI-MANN: Nur leider ohne allzuviel Verständnis für die Denk- und Arbeitsweise eines KI-Wissenschaftlers. Beispielsweise wären Ihre persönlichen Beiträge zwar interessant, jedoch (übrigens für einen Altphilologen bewundernswert genug!) so sehr von der Anschauungsweise und dem Weltbild eines Physikers geprägt, daß sie vielfach am Kern der Sache vorbei argumentiert hätten. (Dr. Hübner verläßt Kopfschüttelnd den Raum in Richtung SPIEGEL-Bar)

Lassen Sie es mich bitte einmal deutlicher sagen: es geht der KI-Forschung letztlich nicht um die schönen technologischen Produkte, die wir bereits erwähnt haben - und schon gar nicht um irgendwelche Schachprogramme. Das eigentliche Forschungsziel ist, zu verstehen wie Intelligenz "funktioniert", welche Verfahren notwendig sind, um Intelligenz zu verwirklichen (egal ob im Menschen oder in der Maschine) und mit welchen Methoden man dies erreichen kann! Im Laufe der letzten 20 Jahre

KI-Forschung ist nun eine ganz ansehnliche Batterie an solchen Verfahren und Methoden entwickelt und entdeckt worden - zwar meist an konkreten Aufgaben (wie Schachspielen, automatischem Beweisen, Computersehen usw.) jedoch haben diese letztlich dabei eine sekundäre Rolle gespielt.

SPIEGEL: Kommen wir zurück zur Frage der Intelligenz von Maschinen: Gibt es eine Möglichkeit, mehr oder minder objektiv festzustellen, ob eine Maschine intelligent ist?

SCHWENKEL: Ich berufe mich da auf einen berühmten Fachkollegen, den Engländer Alan Turing. Er hat empfohlen, die Antwort auf diese Frage durch einen Test zu finden: Hinter einer Leinwand ist ein Mensch oder ein Rechner - man weiß nicht, was - verborgen. Mit diesem Etwas wird ein Dialog über beliebige Themen geführt. Wenn es binnen einer halben Stunde nicht gelingt herauszufinden, ob eine Maschine oder ein Mensch antwortet, dann muß dem Etwas Intelligenz zugebilligt werden. Solange Computer noch nicht sprechen können, muß die Befragung schriftlich erfolgen.

SPIEGEL: Gab es schon Ergebnisse?

SCHWENKEL: Kein Fachmann würde in absehbarer Zeit dieses Experiment durchführen wollen, jedenfalls nicht bei beliebiger Themenwahl. Bei Beschränkung auf ganz bestimmte Themen, wie zum Beispiel das Schachspiel, könnte die Sache anders aussehen.

SPIEGEL: Gibt es Gebiete, auf denen schon bald eine Maschine den Turing-Test bestehen könnte?

FRIEDEL: In den Vereinigten Staaten habe ich in diesem Jahr Dialogprogramme kennengelernt, die in der Lage sind, sich über politische Themen - Waffenlieferungen nach Afrika zum Beispiel - erstaunlich menschlich zu unterhalten, natürlich noch im geschriebenen Dialog. Dabei kann man die Grundeinstellung der Maschine - linksliberal oder rechtskonservativ - durch eine Programmanweisung im voraus bestimmen.

SPIEGEL: Herr BrunNSTEIN, stimmen Sie mit den Herren von Dittfurth und Schwenkel darin überein, daß die künstliche Intelligenz in allerdings noch ferner Zukunft die menschliche Intelligenz übertreffen wird?

BRUNNSTEIN: Nein. Für mich bleibt der Rechner auch in Zukunft das, was er heute ist: eine Prothese, die unsere menschliche Intelligenz verbessert und verlängert, also ein "Denkzeug", analog zum Werkzeug. Die Computer werden uns immer unterlegen bleiben, weil ihnen manches immer fehlen wird: Selbstbestimmung, freier Wille, Moral, Gefühl, eine eigene politische Meinung, die soziale Komponente.

KI-MANN: Mit dieser Bemerkung werden Sie zwar die Mehrheit der Leser auf Ihrer Seite haben - als Fachmann sollten sie jedoch wissen, daß die Argumente die Sie anführen barer Unsinn und Wunschdenken sind. Man kann natürlich nicht mit Sicherheit sagen, ob eine künstliche Intelligenz eines Tages die natürliche (mensch-

liche) Intelligenz übertreffen wird - es sind jedoch trotz intensiver Suche (und diese Frage wird als aktuelles Problem nun seit weit über 30 Jahren intensiv diskutiert) bisher keine Schranken des Rechners bekannt geworden, die der Mensch nicht auch hätte. In dieser Diskussion sind die von Ihnen angeführten Punkte immer wieder gern angeführte Argumente, die durch Wiederholung jedoch nicht an Überzeugungskraft gewinnen:

▷ "eine eigene politische Meinung":

Herr Friedel hat ja bereits als Gegenbeispiel ein amerikanisches Programm erwähnt, das nicht nur erklärtermaßen eine politische Meinung hat, sondern sogar so funktioniert, daß es sich auf einen beliebigen Standpunkt zwischen "konservativ" und "links-liberal" einstellen kann.

▷ "Moral, Gefühle":

es gibt von Colby ein Programm, das einen in New York lebenden paranoiden Patienten simuliert, dessen Verfolgungswahn darin besteht, sich von den Italienern bedroht zu fühlen. Sie sollten einen Dialog mit dem Programm vielleicht einmal durchlesen, um zu sehen, mit welcher Subtilität die Gefühle (d.h. hier die Ängste) simuliert werden! - Simuliert mit solcher Perfektion, daß Psychiater, denen man das Protokoll vorlegte (und die nicht wußten, daß es ein Maschinendialog war) den "Patienten" eindeutig diagnostizierten. Es ist geradezu erschütternd zu sehen, wie der "Patient" durch seine Ängste immer wieder gezwungen wird, von dem ansonsten rationalen Fluß des Dialogs abzuweichen. Ebenso ist die Frage der "Verselbständigung" solcher Programme, d.h. die Möglichkeit, daß solche Programme sich mehr und mehr der Kontrolle durch den Menschen entziehen, durchaus nicht so einfach vom Tisch zu wischen, wie Sie selbst ja bereits vorher (im Hinblick auf militärische Anwendungen) gesagt haben.

BRUNNSTEIN: Glauben Sie, daß ein Computer irgendwann Vertrauen haben kann? Das ist doch eine Kategorie, die ihm fremd bleiben muß.

SCHWENKEL: Da bin ich keineswegs so sicher wie Sie. Wenn man in der Richtung, die jetzt mit den Dialog-Programmen eingeschlagen wird, zügig vorankommt, dann wird eine Maschine aus solcher Unterhaltung auch den Schluß ziehen können, daß sie Vertrauen in die Worte hat, die sie hört, und dann doch irgendwann auch Vertrauen zu dem, der zu ihr spricht.

BRUNNSTEIN: Aber während auch ein skrupelloser Mensch grundsätzlich zu Skrupeln fähig bleibt, ist ein Rechner zu Skrupeln nicht fähig. Oder anders gesagt: Ein Mensch kann gegen seine Erziehung, ein Computer kann nicht gegen sein Programm handeln.

KI-MANN: (lachend) ...damit haben Sie ja herrlich tief in die Pandorabüchse des "gesunden Menschenverstandes" gegriffen! Lassen Sie mich noch einmal sagen, daß die "Künstliche Intelligenz" ein Fachgebiet ist, das man als solches (bisher leider nur im Ausland) studieren kann: ich selbst habe in einer meiner Prüfungen (wie fast alle KI-Studenten) diese Standardfrage nach der an-

geblichen Determiniertheit von Programmen beantworten müssen, und lassen Sie es mich ganz deutlich sagen, Herr Professor: mit dieser Ansicht hätten Sie nicht einmal das Vordiplom bestanden! (Prof. Brunnstein verläßt den Raum in Richtung SPIEGEL-Bar).

Um die Frage adäquat zu diskutieren müßte man sich darüber unterhalten, auf welchem Abstraktionsniveau man ein Programm betrachten will: auf der niedrigsten Implementations-ebene ist es natürlich absolut deterministisch - das heißt aber noch lange nicht, daß damit auch das Verhalten des Programmes als Ganzes auf dem nächst höheren Niveau ebenfalls deterministisch sein muß, und dies Wissen gehört zum "Einmaleins" eines jeden KI-Studenten.

SPIEGEL: Dieser Dialog zwischen Ihnen, den Informatikern unter uns, mutet ein wenig wie Science-Fiction an; wir werden erst in etlichen Jahren wissen, ob es mehr Science oder mehr Fiction war.

KI-MANN: Das ist absolut falsch: Der Science-Fiction-Aspekt liegt an Ihrem Vorverständnis: wären Sie bereit, sagen wir, ein bis fünf Jahre Ihres Lebens der Sache zu opfern und sich auf ein entsprechendes Bildungs- und Studierenerlebnis einzulassen, so würden Sie die Dinge anders sehen.

Lassen Sie mich noch einmal betonen, daß ich an keiner Stelle in diesem Gespräch über mögliche Programme spreche, sondern über real existierende, laufende Computerprogramme, die von jedermann eingesehen werden können. Ich möchte das schon allein deshalb betonen, weil es in Deutschland zu viele Propheten gegeben hat - ~~beispielsweise~~ - die sich um alle wissenschaftliche Respektabilität geschrieben haben, weil sie beständig Dinge angekündigt haben, die sie nicht hatten. Ebenso muß man zugeben, daß es Prognosen von KI-Wissenschaftlern in den USA gegeben hat, die verfrüht waren und vielleicht aus der augenblicklichen Begeisterung über das bereits Machbare - und Sorge über dessen Folgen - verständlich sind.

Ich glaube allerdings nicht, daß die Fakten das Entscheidende für die meisten Menschen sind, sondern eine gewisse emotionale Voreingenommenheit. Um es mit H. Simon, dem vorjährigen Nobelpreisträger (und übrigens einem der führenden KI-Forscher in den USA) zu sagen: die Menschen haben von der Wissenschaft zumindest drei harte Schläge in Bezug auf ihr Selbstverständnis hinnehmen müssen: zunächst kam der Schock, daß unsere Erde durchaus nicht der von Gott auserwählte Mittelpunkt des Weltalls ist - und es ist ja bekannt, was mit den Wissenschaftlern angestellt wurde, die auf diese Tatsache hingewiesen haben. 300 Jahre später hat man sich zwar etwas humaner verhalten, die Ansichten Darwins, wonach wir nicht als Krone der Schöpfung aus einem einmaligen Schöpfungsakt hervorgegangen sind, jedoch nichtsdestoweniger heftig bekämpft. Und heute erzählen die Wissenschaftler, daß auch die letzte, die Einmaligkeit garantierende, Bastion durch das Auftreten intelligenter Maschinen in's Wanken geraten ist. Lassen Sie mich hinzufügen, daß meiner Überzeugung nach dieser letzte "Vorstoß", eben wegen seiner technologischen Konsequenzen, auch der Folgen-

schwerste ist.

(Prof. Schwenkel beginnt sich zögernd zu erheben.)

SPIEGEL: (Die Stimme klingt hohl in dem nun fast leeren Raum) Meine Herren, vielleicht sollten wir ebenfalls ... damit ich noch allen Gesprächsteilnehmern im Namen des SPIEGEL danken kann.

(Prof. Schwenkel und KI-Mann verlassen den Raum und treffen an der Tür auf Prof. Brunnstein, der als Fachmann sich der Faszination der Wahrheit, die diese auf einen intelligenten Kopf auszuüben in der Lage ist, natürlich nicht hat entziehen können und lauschend an der Tür verblieben war. Untergehakt und aufgeräumt gehen die drei nun ebenfalls in Richtung SPIEGEL-Bar, wo bekanntlich freier Alkohol unbegrenzt zur Verfügung steht und bereits eine SPIEGEL-Sekretärin mit langsamen Bewegungen auf einem der Tische tanzt, bis dieser unter dem Einfluß der sich verlagernden Schwerkraft umzuschlagen droht.....)

Jörg Siekmann

Universität Karlsruhe
Institut für Informatik I
Postfach 6380
7500 Karlsruhe

Irreführend

Kritische Anmerkungen zum SPIEGEL-Gespräch (Nr. 17, 23.4.79) mit dem Titel "Computer als Richter und Arzt? Hoimar von Ditfurth, Elektronik-Experten und Schachmeister über künstliche Intelligenz".

Dieses SPIEGEL-Gespräch "über künstliche Intelligenz" (so der Untertitel) beschäftigt sich mit Fragen, die für die KI-Forschung nur ein recht beschränktes Gewicht besitzen. Mechanisches Beweisen, rechnergestütztes Problemlösen, rechnergestützte Programmanalyse und -konstruktion, Mustererkennung, Szenenanalyse, Repräsentation bereichsspezifischen Wissens und Robotik sind Beispiele für KI-Kerngebiete, die nicht einmal erwähnt wurden. Obwohl über Computersysteme diskutiert wurde, die Dialoge in gesprochener natürlicher Sprache führen können, wurden die eigentlichen Schwierigkeiten der Verarbeitung natürlicher Sprache und Ansätze zu ihrer Lösung nicht einmal angedeutet. In diesem Punkt erhielt ich aus dem Gespräch den Eindruck, daß es vor allem um die Phonetik von Computer-äußerungen ("angenehme Stimme") geht.

Spricht man mit "Künstlicher Intelligenz" die bisherigen Forschungsaktivitäten und -ziele auf diesem Gebiet an, so wurde in diesem SPIEGEL-Gespräch nur wenig über KI gesprochen. Gegenstand des Gespräches war vielmehr

1. Eine ausführliche und recht gute Diskussion über Schachprogramme. Hierbei wurde nicht klar, daß Schachprogramme bisher nur einen recht beschränkten Beitrag zur KI liefern konnten.
2. Entwicklungen, die nicht zur KI gehören (Taschen-/Heimrechner, Computer-Graphik, rechnergestützter Unterricht).
3. Private Zukunftsvisionen einiger Teilnehmer ("Überholen" der menschlichen Intelligenz durch den Computer, vollständige Simulation der menschl. Persönlichkeit im Rechner), die bisher nicht Gegenstand wissenschaftlicher Untersuchungen in unserem Gebiet waren.
4. Einige vage Andeutungen über Systeme, die Situationen im medizinischen, juristischen, militärischen und wirtschaftlichen Bereich analysieren und Entscheidungen in diesen Gebieten fällen können. Gerade die hier angedeuteten Systeme liefern keinen unmittelbaren Beitrag zur KI (und sollten es auch nicht), sondern sind in Bereichen wie Operations Research, Informationssysteme und digitaler Simulation angesiedelt. Besonders erfolgreiche KI-Systeme (wie DENDRAL, MYCIN, ...) werden dagegen nicht erwähnt.

Die eigentlichen Kernprobleme der KI, deren Lösung vielleicht zu recht bemerkenswerten Anwendungen führen könnten, wurden damit nicht angesprochen. Statt eines klaren und verständlichen Überblickes über die KI erhielten die SPIEGEL-Redakteure auf ihre Fragen nur Auskünfte, die insgesamt ein völlig irreführendes Bild über dieses Gebiet liefern.

Meines Wissens nach arbeitet keiner der Gesprächsteilnehmer auf dem Gebiet der KI und es ist mein Eindruck, daß sie in ihrer fachlichen Kompetenz (Ausnahme: Schachprogramme) überfordert waren. Warum hat DER SPIEGEL in einem Gespräch über KI keinen KI-Experten befragt?

Für besonders gefährlich halte ich an diesem Gespräch die dem Leser suggerierten Zukunftsvisionen. Während sie als Denkmodelle sicher nicht nur für Science-fiction-Geschichten interessant und anregend sein mögen, wurde im SPIEGEL-Gespräch nicht klar, daß sie nicht auf belegbaren Resultaten der KI-Forschung beruhen. Unter dem Mantel fachlicher Kompetenz mitgeteilt, können sie im Leser starke emotionale Betroffenheit auslösen. In Anbetracht der sich in den letzten Jahren häufenden unsachlichen Angriffen auf technologische Entwicklungen sehe ich in der klaren Trennung von Vorstellung und Wirklichkeit eine besonders wichtige Aufgabe der Wissenschaftler, wenn sie ihrer Informationspflicht gegenüber der Öffentlichkeit nachkommen.

P. Raulefs
Inst. f. Informatik, Univ. Bonn

Colloquium on Logic in Programming

10-15 September 1978, Salgótarján (Hungary)

Nowadays computer science is one of the most important fields where mathematical logic is applied. That is the reason why a Hungarian mathematical society, named after János Bolyai, organized a colloquium on mathematical logic in programming. The colloquium took place in a picturesquely small Hungarian town, Salgótarján in the second week of September of 1978, just after the Mathematical Foundation of Computer Science (MFCS'78, Zakopane).

In this review we do not intend to give a detailed report on the colloquium, but to use this occasion to discuss the "state of art" of the mathematical logical approach toward programming theory.

The basic problem of the mathematical foundation of computer science is the one of the correct, formal semantics of programming languages. This theoretical question is coupled with a practical one: giving methods for proving properties of programs. These demands make natural the use of mathematical logic: semantics and proof procedures have been studied mainly in this field.

Before starting to discuss what mathematical logic can give to programming, we should say something about algebraic semantics, since this is closely related to logic. Every kind of semantics that meets Frege's principles ("meaning of some compound expression is obtained unconditionally as a combination of the meanings of its constituent parts") can be defined naturally in an algebraic way. This is also applicable to programming languages, as Zs. Markusz's and M. Szóts' talk showed

using Montague's Universal Grammar. Algebraic semantics is widely studied, it is enough to refer to the activity of the ADJ group (E.G. Wagner, J.B. Wright, J.A. Goguen and J.W. Thatcher). During the colloquium there were some talks based on algebraic tools. The most important of them is the one held by B. Courcelle, analysing recursive program schemes.

Semantics based on mathematical logic is not independent of algebraic semantics since the languages of logic themselves can be treated in an algebraic fashion (algebraic logic). The connection between these two kinds of semantics was discussed on the colloquium by I. Németi. He showed how algebraic semantics can be derived from model theoretic semantics. He gave a formalised version of Frege's principle: for a given language (that is for a given triple

\langle set of expressions, class of models, meaning function \rangle

) a grammar is adequate, if the kernel of the meaning function is a congruence relation with respect to the rules of the grammar. The connection between algebraic and logical semantics is not only a theoretical one. As the language of algebra (the language of equations) can be considered as a sublanguage of classical first order logic, they can be used together. A. Bertoni, G. C. Mauri and P. A. Miglioli used this phenomenon describing abstract data types.

One of the important questions of logical semantics is how to choose the appropriate language for describing programs. Though the most widespread and most commonly known language of logic is the predicate calculus, there are several attempts to use higher order or non classical language. The reason for this is that the static character of classical logic makes it hard to use for describing programs having dynamic meanings. (At the MFCS'78 conference P. van Emde Boas gave an interesting review of the recent attempts for constructing non-classical logics of programs.) Several talks in Salgótarján dealt with algorithmic logic, a non-classical logic worked out by Polish mathematicians

(L. Banachowski, H. Rasiowa, A. Salwicki and others) in the late sixties.

In algorithmic logic programs are terms, and a new type of formula is introduced; if K is a program, and p is a formula, then Kp is also a formula. Kp is true, if p is true after the termination of K . A similar logic was constructed at MIT, the so-called dynamic logic. As A. Kreczmar pointed out, the main difference between these two logics is that dynamic logic considers programs as relations, not as functions, to be able to cope with nondeterministic programs. Here $[K] p$ means that p is true after each execution of K , and $\langle K \rangle p$ that there is an execution of K which makes p true. One can see that non-determinism gives rise to a kind of modality. The same trick can be done also in the frame of algorithmic logic, as G. Mirkowska's talk showed.

M. Grabowski showed that algorithmic logic can be expressed in infinitary or second order logic.

Beside the question of expressive power of the semantics describing language, the existence of a proper calculus is a pivotal question. Any kind of logic speaking about programs has to make us able to prove properties of programs. In the field of program verification one of the most important recent results is that there is no complete calculus to prove correctness of programs. This forces several computer scientists to weaken the notion of proof. The dynamic logic takes non recursive axiom system (all the formulas valid in the standard model of arithmetic), algorithmic logic has an infinite rule of inference to handle program loops, a version of the ω -rule. So the completeness of these logics can be stated, but the definability of the notion of proof is lost, that is it can not be decided whether a piece of text is a proof or not. Moreover the notion of completeness is weakened; the set of theorems does not remain recursively enumerable.

H. Andréka, I. Németi, T. Gergely started along an other way. (They and their coworkers will be referred as AGN group.) Their starting point is first order classical logic. They studied why complete calculus for programs could not be found and tried to characterise the class of provable programs. Their model theoretical research showed that the cause of incompleteness is the role of some kind of time scale in the semantics of programs - since their meaning is established by their executions. This role was made explicit: the language was extended by formulas referring to time. T. Gergely's talk summarised the effects of this approach. He pointed out that usual way of analysis assumed that the execution of programs is according to a time scale isomorphic with natural numbers. So every assertions about programs involves the standard model of arithmetic. As he expressed it: " the metalanguage is the language of ω -logic ". To avoid the incompleteness caused by it, non standard models of axiom systems of arithmetic have to be allowed to play the role of time scale, that is programs may terminate in "non-standard" time. This is somehow against our intuition about programs, and forces us to alter some of our ideas. As an example: only a generalised theory of recursive functions can be applied. (See L. Ury's talk.) However this theory makes properties of programs treatable in the frame of mathematical logic. H. Andréka showed, also using non-standand time scale that the method of inductive assertions is complete. Moreover, she proved that a weaker condition does not ensure completeness. After the axiomatization of the method of inductive assertions (done by Z. Manna) this is the most important result in the theory of the method.

However in spite of the results, the novelty of the AGN group's approach makes it hard to accept. A. Salwicki argued against

the necessity of speaking about any kind of time scale. Their discussions made clear that two antagonistic views met at the colloquium. Algorithmic logic built the notions of computing techniques into mathematical logic, not only to study programs, but also to break into mathematics. According to Salwicki, they would like to see an algorithmic centred, constructive mathematics;

"mathematics and computer science are different façades of the same building". The AGN group views this building from the other side. They do not want to make mathematics more algorithmic, but smuggle the tools of abstract mathematics into the analysis of programs. Maybe, it seems that algorithmic logic is more practical for those in programming practice.

However if we look at the practical consequences of both theories, we find that the use of the ω -rule algorithmic logic's proof procedures demands high quality of mathematical knowledge and a lot of invention. On the other hand, the AGN group uses sophisticated mathematical notions to give a correct mathematical theory for finitary, simply usable tools of programming.

The aim of research activities in computer science is not only to make clear what programs and programming are, but to give a theoretical basis and methods for a future programming technology too. Only J. Aszalós talked directly about programming technology. He did not give a new theory or formalism, but showed how a system programmer, who is familiar with mathematical logic, could use it in his practical work.

In the last few years the greatest success of logic in programming has been the PROLOG language. PROLOG is a programming language based on logic. It is the first practical application of the principle "logic as a programming language". The underlying idea is that a certain set of axioms can be considered as a program, and a theorem to prove as the actual task for the program. If we have an automatic theorem prover effective enough, it can play the role of the interpreter. And in the case of Horn formulas the resolution principle proved itself effective enough. PROLOG is widely applied for artificial intelligence problems. In Hungary it has stepped out of the AI laboratories, and it is used in everyday computations. The talks given by M. Bruynoogh, L. Pereira, and P. Szeredi showed that they do their best to improve PROLOG. F. Darvas' and his coauthors' talk informed us about applications in the drug industry.

Realising, that the colloquium showed a certain gap between theory and practice, the organising committee set up "logic and practice" as the subject for the panel discussion. However it soon turned into a discussion on the computing of the future. First we thought that there was a tacit consent about the necessity of making computer systems more intelligent, but several of the participants did not think this possible because of complexity bounds. They did not realise that not the problems themselves, but the methods for their solution were characterised by these bounds. Moreover, the members of the algorithmic logic school did not want to work on developing more intelligent systems. For them assignment and iteration, as the underlying ideas of present programming,

meant the top of the culture of programming. However computer scientists working on PROLOG pointed out how pattern matching and backtracking opened a wide horizon for new possibilities.

We reviewed the colloquium from a special point of view, therefore we omitted several interesting talks. Such were O. Stepankova's on the subject of automatic problem solving, L. Farinas del Cerro's about mechanization in non classical logic and several others. The papers will be published by North Holland in the series "Colloquia Mathematica Societatis János Bolyai".

Authors:

Edit Sánta-Tóth

SzKI, Budapest, Akadémia u. 17. 1054.
HUNGARY

Miklós Szóts

SzÁMKI, Budapest, Pf. 227. 1536.
HUNGARY

REPORT ON THE SECOND INTERNATIONAL
WORKSHOP ON THE SEMANTICS OF
PROGRAMMING LANGUAGES

(BAD HONNEF, MARCH 19-23, 1979)

There were about 35 participants, and 24 of them gave presentations of their current work. The programme proceeded at a leisurely pace, with good time for discussion both during and between the sessions - an essential feature of a real "workshop".

Roughly 1/3 of the participants had attended a similar event in Oberwolfach, 2 months earlier. They should be admired for their energy - Willem de Roever justified going through his slides rather quickly by the fact that so many in the audience had seen them at the Oberwolfach meeting! - but would it not be more sensible to plan such workshops with a decent interval between them ?

Extended abstracts of the talks are given below. For me one of the high points was David Park showing us how to give a sensible semantics for non-deterministic "Kahn-networks" - very pretty!

Honnefer Haus provided a comfortable setting for the workshop - as did the local Weinhaus, in the evening. The fine weather on the Wednesday afternoon encouraged most of us to celebrate the equinox with a walk in the hills.

The Workshop was efficiently organised by J. Loeckx and K. Indermark and supported financially by the DFG and the Minister für Wissenschaft und Forschung des Landes Nord-Rhein-Westfalen.

Peter Mosses
Aarhus, Denmark

List of participants

- ALBER, K. / Lehrstuhl für Informatik / Gaußstraße 12 /
D-3300 Braunschweig
- APT, D. / Kamer H 5-12 / Burgemeester Oudlaan 50 / Erasmus Uni-
versiteit / NL-Rotterdam
- ARNOLD, A. / Université de Poitiers / Laboratoire d'Informatique /
Bâtiment de Mathématiques / 40, av. du Recteur Pineau /
F-86022 Poitiers-Cédex
- de BAKKER, J. / Mathematisch Centrum / 2e Boerhaavestraat 49 /
NL-1091 Amsterdam
- BIBEL, W. / Institut für Informatik / TUM / Arcisstraße 21 /
D-8000 München
- BOEHM, C. / Istituto Matematico 'G. Castelnuovo' / Piazzale delle
Scienze / I-00815 Roma
- CLAUS, V. / Lehrstuhl für Informatik / Postfach 500 500 /
D-4600 Dortmund 50
- COURCELLE, B. / Université de Bordeaux 1 / Mathématiques-Informa-
tique / 351 Cours de la Libération / F-33405 Talence-Cédex
- COUSINEAU, G. / Institut de Programmation / Université Paris VI /
4, Place Jussieu - Cédex 05 / F-75230 Paris
- CREMERS, A. / Lehrstuhl für Informatik / Postfach 500500 /
D-4600 Dortmund 50
- DAMM, W. / Lehrstuhl für Informatik II / Technische Hochschule /
Büchel 29-31 / D-5100 Aachen
- EHRICH, H. / Lehrstuhl für Informatik / Postfach 500500 /
D-4600 Dortmund 50
- FEHR, E. / Lehrstuhl für Informatik II / Technische Hochschule /
Büchel 29-31 / D-5100 Aachen
- GAUDEL, M. / IRIA Domaine de Voluceau / Rocquencourt / B.P. 105 /
F-78150 Le Chesnay
- GUESSARIAN, I. / Laboratoire d'Inform. Théorique et Programmation /
Université Paris 7 / Tour 45-55 / 2, place Jussieu / F-75005
Paris
- HOTZ, G. / Fachbereich 10 - Informatik / Universität des Saarlandes /
D-6600 Saarbrücken
- INDERMARK, K. / Lehrstuhl für Informatik II / Technische Hochschule /
Büchel 29-31 / D-5100 Aachen
- KRÖGER, F. / Institut für Informatik / TUM / Arcisstraße 21 /
D-8000 München 2

- LANGMAACK, H. / Institut für Informatik / Christian-Albrecht-Universität / Olshausenstraße 40-60 / 2300 Kiel 1
- LEHMANN, S. / Fachbereich 10 - Informatik / Universität des Saarlandes / D-6600 Saarbrücken
- LOECKX, J. / Fachbereich 10 - Informatik / Universität des Saarlandes / D-6600 Saarbrücken
- MAZURKIEWICZ, A. / Computer Centre / Polish Academy of Sciences
PKIN, P.O. Box 22 / PL-00.901 Warszawa
- MILNER, R. / Computer Science Department / Edinburgh University /
Mayfield Road / GB-Edinburgh EH 93 JZ
- MOSESSE, P. / Dept. of Computer Science / University of Aarhus /
DK-Aarhus
- NEWBY, M. / Computer Center / Australian National University /
Canberra Act/Australia 2600
- NIVAT, M. / Université Paris 7 / Tour 45 - 55 / 2, place Jussieu /
F-75005 Paris
- OLNHOF, T. / Institut für Informatik / Azenbergstraße 12 /
D-7000 Stuttgart 1
- PARK, D. / University of Warwick Coventry / GB-Warwickshire CV 47 AL
- RAOULT, J. / Université de Paris Sud-Orsay / L.R.I. / Bâtiment 490
F-91405 Orsay Cedex
- RAULEFS, P. / Institut für Informatik / Wegelerstraße 6 /
D-5300 Bonn
- REISSIG, W. / Lehrstuhl für Informatik II / Technische Hochschule /
Büchel 29-31
- de ROEVER, W. / University of Utrecht / Dept. of Computer Science /
Budapestlaan 8, P.B. 80012 / NL-3508 TA Utrecht
- SIEBER, K. / Fachbereich 10 - Informatik / Universität des Saarlandes / D-6600 Saarbrücken
- TIURYN, J./Lehrstuhl für Informatik II / Technische Hochschule /
Büchel 29-31 / D-5100 Aachen
- WADSWORTH, C. / Dept. of Computer Science / King's Building /
University of Edinburgh / GB-Edinburgh EH 9 3 JZ

ABSTRACTS OF THE TALKS

(in the order of their presentation)

INFINITE PROOF RULES FOR WHILE PROGRAMS

Fred Kröger (München)

Many methods and formal proof rules for proving partial correctness of (while) loops by inductive assertions have been proposed in the literature. Well known keywords are for example computation induction, predicate transformers, structural induction and others which all focus - in one way or the other - on the fundamental notion of loop invariants, most concisely expressed in Hoare's original rule $P \wedge [I] P \vdash P[\text{while } B \text{ do } I \text{ od}] P \wedge \neg B$. As also known, this rule bears a certain kind of incompleteness: we cannot really verify all correct while loops by it.

We introduce infinite proof rules (rules with infinitely many premises) for while loops in order to overcome this proof-theoretical incompleteness and to get some insight into it. In fact, partial correctness of while programs can be completely axiomatized using such a rule. However, these rules are also of practical interest: they put the invariant method into a more general framework and open the view to other kinds of inductive assertions carrying a proof through a loop. They naturally suggest to use "parameter representations" of invariants which might be more manageable in some applications. This being principally not different from the invariant method, the infinite rules give rise also to a proper extension of the latter method: the method of generalized invariants. A generalized invariant is an assertion which need not hold after each single run through the loop but recurs in possibly longer (not necessarily fixed) periods. There are programs for which an induction over such arbitrary "pillars" is adequate and much simpler than an induction with a classical invariant.

AN AXIOMATIC PROOF SYSTEM FOR WHILE-PROGRAMS

Jacques Loeckx (Saarbrücken)

The axiomatic system presented is an extension of the first-order predicate calculus. It is characterized by additional formulas and a few additional inference rules.

The additional formulas are of the form

$$ij:q$$

where i and j are cutpoints in a while-program and where q is a formula of the predicate calculus. Intuitively $ij:q$ means that q holds for each path leading from cutpoint i to cutpoint j .

The additional inference rules correspond to classical proof methods: two inference rules may be viewed as an implementation of the inductive assertion method, two further inference rules implement the subgoal induction method and the fifth inference rule implements the well-founded sets method.

For proving the partial correctness or termination of a while-program it suffices to "translate" this program in a set of axioms and then to apply the inference rules.

While being completely formal, proofs are substantially shorter and more readable than proofs in the Hoare system; this essentially results from the simplicity of the formalism used. Some experience with the abstract data programming language Alphard suggests that this may be a definite advantage in practical work.

A formal description of the axiomatic system may be found in [1,2]; a consistency proof is in [2].

- [1] J. Loeckx, I. Glasner, "A calculus for proving properties of programs"; presented at the Intern. Conf. on Math. Studies of Inf. Proc. in Kyoto, 1978 (the proceedings of this Conference are to be published as Springer Lecture Notes in Comp. Science)
- [2] I. Glasner, "Formale Beweise über while-Programme : Ein Kalkül und sein Modell", Diplomarbeit, Fachbereich 10, Universität des Saarlandes, Saarbrücken, 1978

RECURSIVE TYPE OPERATORS WHICH ARE MORE THAN TYPE SCHEMES

C.P. Wadsworth (Edinburgh)

To date, natural examples of recursive type operators have been relatively few and limited in the nature of their recursion. Lists, parameterized on the type of their elements, and binary trees, parameterized on the type of their tips, are two well-known examples; these are "schematic" in the sense that they are a schema for a single recursive type definition at each particular type instance of their type parameter (since, e.g., lists of elements of some type α have tails which are lists of elements of the same type α).

This talk presents programming examples of non-schematic recursive type operators, arising from data structures, called tries, for

fast searching of a collection of records indexed by keys (as in, e.g., Knuth, Vol. 3). Tries, parameterized on the type of the records indexed, become nested recursive type operators when the keys are terms of more general languages than character strings. This is illustrated for a simple language of terms which includes combinations $(t_1 t_2)$, where t_1 and t_2 are terms called the rator and rand, respectively. The genesis of the nested recursion of types is that the (sub-)collection of records indexed by combinations is first partitioned into terms indexed by the terms occurring as rators, the item indexed by each particular rator t_1 being the collection of records indexed by the rands of all combinations whose rator is t_1 ; then the partitioning is continued recursively for rators and rands which are themselves combinations.

Some implications for typechecking are mentioned briefly, observing that the examples lead to recursively defined, polymorphic functions whose inner and outer calls are at different instances of their "generic", polymorphic type.

The generalized tries are currently used for programming the term matching part of the simplification package in the Edinburgh LCF system, to determine "simultaneously" whether a subject term matches any one of a set of pattern terms occurring as the left hand side of simplification rules.

ABOUT ALGEBRAIC SEMANTICS

Irène Guessarian (Paris)

We study two usual properties of program schemes: termination and a simplification property. To this end, we introduce the notion of useless occurrence which enables us to consider these two problems in a unifying setting /1/; algebraic semantics provides us with a quite flexible framework for this study in that we can introduce the exactly desired amount of semantic information and derive the corresponding results.

We begin with a purely syntactic stage, and, working in the free interpretation, obtain easily at this stage quite worthwhile simplifications. Then we introduce some amount of semantic information, and, working in a class of interpretations defined by algebraic conditions, show how we can delete some of the useless occurrences and non-terminating procedures with respect to that class of interpretations; this is achieved by using only the algebraic specifications of the program scheme and of the class of interpretations. Finally, we show how one can, making more and more precise the specifications of the interpretations, derive

in the same way more and more refined simplifications of the program scheme /2/.

/1/ I. GUESSARIAN, Program transformations and algebraic semantics, to appear in TCS, 1979

/2/ F. ERMINE & I. GUESSARIAN, About termination and simplification of program schemes, in preparation

INFINITE TREES IN NORMAL FORM AND RECURSIVE EQUATIONS HAVING A UNIQUE SOLUTION

Bruno Courcelle (Bordeaux)

A system of recursive equations is C -univocal if it has a unique solution modulo the equivalence associated with a class C of interpretations. This concept yields simplified proofs of equivalence of recursive program schemes and correctness criteria for the validity of certain program transformations, provided one has syntactical easily testable conditions for C -univocality. Such conditions are given for equational classes of interpretations.

They rest upon another concept: the normal form of an infinite tree with respect to a tree rewriting system. This concept yields a simplified construction of the Herbrand interpretation of certain equational classes of interpretations.

The equivalence of two program schemes is a very strong requirement since they must compute the same function in every interpretation. We define the C -equivalence (denoted by \equiv_C) similarly, with respect to a given class C of interpretations instead of the full class of all interpretations. This flexibility is necessary for practical applications. In particular, the library of Darlington and Burstall [11] is in fact a set of triples $\langle \phi, \psi, A \rangle$ of two schemes ϕ and ψ and a set A of conditions such that ϕ is C_A -equivalent to ψ (where C_A is the class of all interpretations which meet A).

It has been remarked by Courcelle [4], and by Elgot et al. [12] in a restricted case, that a system Σ has a unique solution in $M_\Omega^\infty(F, V)$, under some not very stringent syntactical conditions. Let us say that such a system is univocal. More generally, a system is C -univocal if it has a unique solution in $M_\Omega^\infty(F, V)/\equiv_C$. We raise the following problem:

Problem 1: Which systems are C -univocal ?

Clearly the answer will depend on the way the class C is specified. We shall only answer to the question by giving sufficient conditions, in the case where C is defined by equations. Before describing these conditions we give two applications to program proofs and program transformations of the concept of C -univocality.

Application 1: Simplified program proofs.

Let Σ be a system of the form $\langle \phi_1 (v_1, \dots, v_{k_1}) = \tau_1, \dots, \phi_n (v_1, \dots, v_{k_n}) = \tau_n \rangle$;

let $\alpha_1, \dots, \alpha_n$ be functions defined by another system of equations Σ' . Assume that $\langle \alpha_1, \dots, \alpha_n \rangle$ satisfies the system Σ modulo \equiv_C i.e. that :

$$(1.3.1) \alpha_i \equiv_C \tau_i \{ \alpha_1 / \phi_1, \dots, \alpha_n / \phi_n \} \text{ for all } 1 \leq i \leq n.$$

(The right hand side of each equation is the substitution of α_j for each occurrence of ϕ_j in τ_i). This means that for all interpretation I in C , $\langle \alpha_{1_I}, \dots, \alpha_{n_I} \rangle$ is a particular solution of Σ hence that $\phi_{i_I} \leq \alpha_{i_I}$ since $\langle \phi_{1_I}, \dots, \phi_{n_I} \rangle$ is the least solution of Σ in I . Since this holds for all $I \in C$ we write this:

$$(1.3.2) \phi_i \leq_C \alpha_i \text{ for all } 1 \leq i \leq n.$$

This is essentially the Recursion Induction of McCarthy [23] or the Fixpoint Induction of Park.

If we also know that Σ is C -univocal, we can infer directly (1.3.3.) from (1.3.1.);

$$(1.3.3) \phi_i \equiv_C \alpha_i \text{ for all } 1 \leq i \leq n.$$

Hence we obtain a powerful method, provided the conditions insuring the C -univocality of Σ are not too difficult to test.

Application 2: Validation of certain transformations of recursive programs.

Let us recall the transformation method of Burstall and Darlington [3] using the "unfold-fold-redefine" technique. Given a system Σ reduced to a single equation: $\phi(v_1, \dots, v_k) = \tau$ (this is only to simplify this informal presentation but the extension to general systems is straightforward) and a fixed class of interpretations C , one wants to find $\Sigma' : \phi'(v_1, \dots, v_k) = \tau'$ such that $\phi' \equiv_C \phi$ and Σ' is simpler than Σ in some sense.

A possible way to find τ' is to build a sequence $\tau_0, \tau_1, \dots, \tau_m$ such

that $\tau_0 = \tau$ and for all $0 \leq i < m$:

$\tau_i \xrightarrow{\Sigma} \tau_{i+1}$ i.e. some ϕ in τ_i has been rewritten as τ
 ("unfolding" of [3]) or
 $\tau_{i+1} \xrightarrow{\Sigma} \tau_i$ ("folding" of [3]) or
 $\tau_i \equiv_C \tau_{i+1}$ by some property of the base functions (such that
 associativity of a binary function) always satisfied
 in C ("using laws about primitives" of [3]).

Assuming that τ_m is "simpler" than τ we would like to take $\tau' = \tau_m \{ \phi' / \phi \}$. A sufficient condition is that the equation $\Sigma' : \phi'(v_1, \dots, v_k) = \tau'$ is C -univocal. It is clear that $\tau_i \equiv_C \tau_{i+1}$ for all i hence that $\phi \equiv_C \tau_m$. This means that ϕ (defined by Σ) is a particular solution of Σ' . Hence $\phi \equiv_C \phi'$ if Σ' is C -univocal. (Note that $\phi' \leq_C \phi$ always holds; an example such that $\phi' \neq_C \phi$ is given in (5.22)).

RATIONAL OPERATIONS ON TREES WITH APPLICATION TO CONTROL STRUCTURES

Guy Cousineau (Paris)

Regular trees can be defined as solutions of regular systems of equations or equivalently as those trees that have only a finite number of distinct subtrees. Here we characterize them as the tree that can be obtained from finite ones by means of a finite number of applications of so-called rational operations including some concatenation and star operation on trees. We should like to emphasize the fact that giving such a characterization was not merely for us a tree-theoretic task (if it had been the case, other notions of rationality could have been chosen) but we wanted our rational operations to be meaningful as program constructs.

The trees we manipulate are elements of $M^\infty(F, \mathbb{N})$ in the notations of Nivat when F is a set of function symbols of arity ≥ 1 and \mathbb{N} the set of integers. Concatenation is defined by $a_1 \cdot a_2 = a_1 [0 \setminus a_2]$ and star by $a^* = \downarrow(a^\infty)$ where a^∞ is the limit of the sequence a^n and \downarrow an operation that decreases by one all the integers that can be found in a tree.

Let us point out some results that can be obtained using this formalism:

1. Semantics of programs containing loops can be very naturally defined and properties of rational expressions induce properties of programs. For example the fact that you can solve a regular system into a rational expression explains why flowchart programs can be structured using repeat - exit and hierarchy results on rational expressions induce the Kosaraju hierarchy on repeat - exit schemes.

2. The rule used to solve regular equations can be used to design a Salomaa-like system for proving identities of rational expressions. From that system, one can derive a sound and complete transformation system to deal with syntactical transformations of programs.

3. The formalism extends very easily to non-regular cases and it gives a way to measure the expressive power of different classes of programs by means of the associated class of trees.

SYNTAX-DIRECTED, SEMANTICS-SUPPORTED PROGRAM SYNTHESIS

Wolfgang Bibel (München)

The semantics of programs simplifies substantially if algorithms are represented by their logic and their control separately. Therefore the question arises whether it is feasible to synthesize the logic and the control of algorithms in separate steps. In this talk a number of strategies for the synthesis of the logic part of algorithms from a given input-output specification of a problem are presented which are centered around a few basic principles. It has been verified for more than ten different algorithms that their uniform application in all cases results in a successful deductive synthesis, among which are a spanning-tree algorithm, a graph-circuits algorithm, a finding-the-ith-smallest-element algorithm, and a linear (string) pattern matching algorithm. In all cases the subsequent determination of the control is a simple, mechanizable task. Also the correctness of the resulting algorithm is guaranteed by the generated deduction.

This provides some support to the thesis that such a "logic programming" approach is not only desirable but also feasible. It also reveals arguments in favor of a classification of algorithms according to relevant syntactic properties of the logic of algorithms as opposed to one according to program (control) structures such as program schemes.

FUNCTIONAL SEMANTICS OF A RELATIONAL DATABASE-
QUERY-LANGUAGE WITH SOME RESPECT TO A HOSTLANGUAGE

Thomas Olnhoff (Stuttgart)

Database-sublanguages become embedded into major programming-languages (e.g. PASCAL by J.W. Schmidt). The semantic-description should not fall short of these sublanguages. A program-state can be split into a non-db-component and a db-component. This talk is mainly concerned with describing a query-sublanguage's semantics as a state-mapping of the db-component.

A database description mostly distinguishes three kinds of models: the external, conceptual and internal model. Several external models are mapped into a conceptual one which is mapped into the internal one. A query is formulated by means of external objects. Its final interpretation is done by internal objects.

The query-language studied has the power of the relational algebra confined to the operators select, join, project. The mappings we allow from one model to the other are noted as well - formed sentences of the query-language (QL). Another language (OQL) is used to express the query on the internal model. Thereby we hope to be able to give the semantics of a query as a state-function being the composition of "elementary" functions that can be thought of as abstractions of modules interpreting the query in an actual implementation.

How equivalence of sentences q and oq (q of QL, oq of OQL) can be decided is sketched.

The Vienna-Definition-Method is used as the metalanguage for the presentation of this talk.

AN EXERCISE IN BEHAVIOUR ALGEBRA;
PROVING THAT A QUEUE WORKS PROPERLY.

Robin Milner (Edinburgh)

A queue system built from active cells was considered. Each cell represents a place in the queue, and it may be in one of three states. It may contain a member, in which case it can (if allowed)

pass this member forward to the next higher place, and become empty itself. An empty place has two capabilities - it may receive a member passed from the next lower place, or it may receive a signal from the next lower place that it should become the end place. An end place has two capabilities; it may signal to the next higher place (if empty) to become the end place, or it may receive a new member from outside and then divide into two places - an occupied place and an end place. Any communication requires both participants to perform it simultaneously.

In Behaviour Algebra, the behaviours of the three kinds of place may be defined by mutual recursion

An occupied place:	Place (a)	=	$\bar{\alpha}_1$ a. Empty
An empty place :	Empty	=	β_1 x. Place (x)+ β_2 .End
An end place :	End	=	$\bar{\alpha}_2$.NIL+ γ x.(Place(x) \circ End)

The five ports $\bar{\alpha}_1, \bar{\alpha}_2, \beta_1, \beta_2, \gamma$ are for the following purposes

$\bar{\alpha}_1$:	Pass a member (up)
$\bar{\alpha}_2$:	Signal (up) to become end place
β_1	:	Receive member (from below)
β_2	:	Receive signal (from below) to become end place
γ	:	Receive member (from outside)

The chaining combinator \circ is used to join ports of places ($\bar{\alpha}_1$ to β_1 , $\bar{\alpha}_2$ to β_2); e.g. the queue with two members (a_1 and a_2) and no empty places is represented by

Place (a_1) \circ Place (a_2) \circ End.

The exercise consisted in defining, by recursive equations, the behaviour that a queue should have (that is giving its specification) and then proving that this is identical with the behaviour of the composite queue system.

For this purpose, the Laws of Behaviour (presented in "Synthesis of Communicating Behaviours" by the author, in Proc. 7th MFCS Colloquium, Zakopane, Poland, 1978 (Springer - Verlag)) were used, together with three further laws which are part of an equational system which we hope to prove complete (that is, behaviours will be provably identical iff they are observationally equivalent). The proof used a simple case of induction upon the iterates of recursively defined behaviours.

SEMANTICS FOR CONCURRENTLY COMMUNICATING
FINITE SEQUENTIAL PROCESSES, BASED ON PREDICATE TRANSFORMERS.

W.P. de Roever (Utrecht)

Denotational semantics are obtained for the flow-of-control of finite, communicating, sequential processes in Hoare's language CSP (CACM aug. '78). The project derives its interest from occurrence of two different forms of nondeterminism, (a) local nondeterminism restricted to one process only, and (b) global nondeterminism, which can only be resolved after consultation between processes. Since only finite executions of processes are focussed upon, and resulting local final states of each process should satisfy postconditions - e.g., a parallel sorting routine should produce a sorted outcome - we utilize a variant of Dijkstra's weakest precondition semantics $wp[S] q$, S denoting a CSP command, q a postcondition on tuples (s, T) of local states s , and communication futures T . Integration of communication in this framework takes place by introducing (a) a-priori semantics for every process P_i , describing P_i 's behaviour for every message communicable which contributes to termination of S inside q , and (b) a binding operator which "binds" the a-priori semantics of all communicating processes by establishing matching pairs of input/output communications, preserving the relative order in which messages are communicated. A system of simple equations for constructs of CSP is obtained improving upon previous result by Francez, Hoare, Lehmann, de Roever. The rule for boolean guarded commands for non-communicating computation must be changed because of inter-process communication. Equations for do-loops are of the same order of complexity as for the sequential case.

THE USE OF A FORMAL SEMANTICS TO PRODUCE AND
PROVE COMPILERS

M.C. Gaudel (Iria, Rocquencourt)

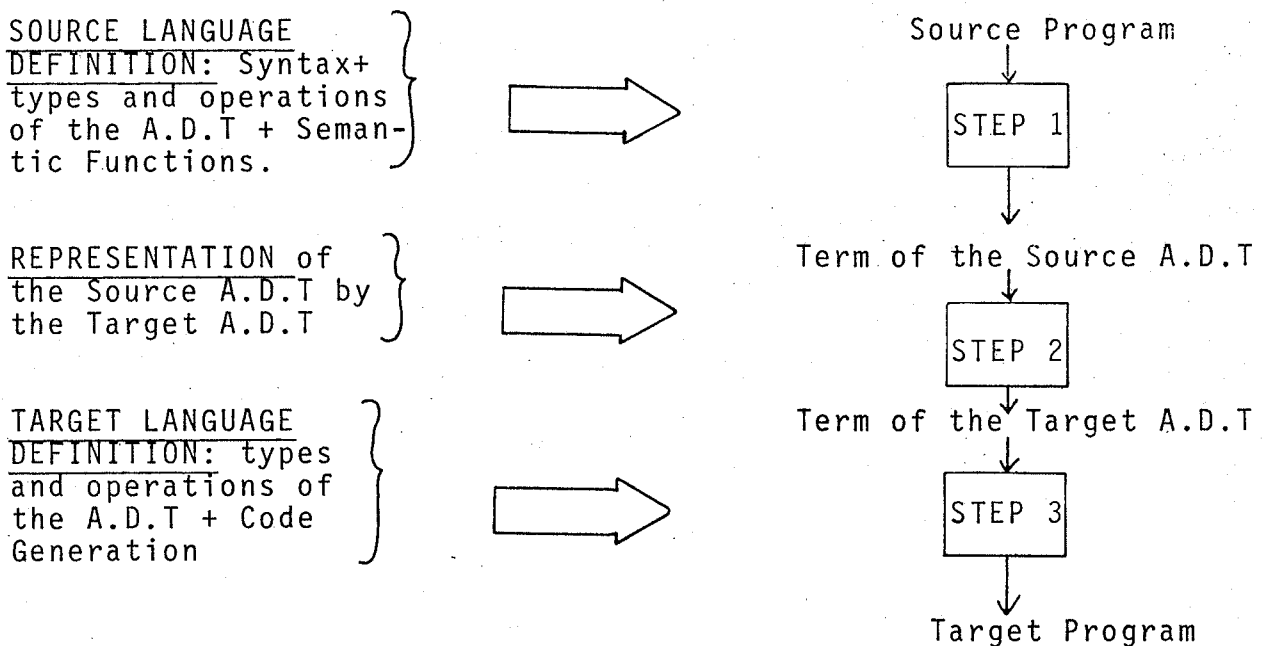
C. Pair (CRIN, Nancy)

We present a formal semantics which is used to specify, produce and prove compilers. The basic idea is to use some results on the representation of algebraic abstract data types and the validation of such representations.

In the system currently developed, a definition of the Source

Language and of the Target Language is given. The same formalism is used to describe the semantics of these two languages. Then the translation process can be specified in a formal way, and proved. To each language is associated an algebraic abstract data type, i.e. a list of names of type; a list of names of operations with their domains and co-domain; a list of axioms to define the relationship between the operations. The semantic value of a program is a term of the abstract data type. The translation of the semantic values of Source programs to semantic values of Target programs can then be specified and proved as the representation of an abstract data type by another one.

The general scheme of the system is:



The translators obtained work in three steps. The axioms are used to perform proofs of the second step of translation.

The specific goal of this paper is to show, by the way of an example, what the semantics of a realistic programming language looks like in this formalism. We consider a usual Algol-like programming language which allows to declare and use simple variables, arrays and procedures. Simple and mutual recursivities can occur. Procedure parameters are called by value or called by variable.

The characteristic axioms of this language are stated and the semantic functions are given.

AN ABSTRACT NOTION OF ONE-DIMENSIONAL
ARRAY WITH SOME APPLICATIONS ON FORESTS AND TREES.

C. Boehm (Roma)

The non adequacy of Church's tuple concept in λ -calculus as an abstract model for one-dimensional arrays is first illustrated. In fact there is no operation which independently of the 'nature' of the components, can compute the length of each array, or even select its first element; there is moreover no predicate distinguishing the empty array from the non-empty ones. It is probably true that because of such difficulties J. Mc Carthy decided twenty years ago to deviate from the pure λ -calculus by designing the programming language LISP.

Strangely enough, the preceding 'impossibilities' can all be eliminated by a different basic approach to the notion of one dimensional array: this choice enables one also to describe two important operations on arrays, the 'append' operation and the 'rotation to the right' by two non recursive operators.

Operators for handling errors may also be introduced. As a further application of the preceding notion to the case where atoms (nodes) are restricted to be non negative integers, it is possible to develop a consistent notion of ordered forest and tree, giving a semantic counterpart to LISP data structures.

PURE LISP IN LCF, AGAIN !

Malcolm C. Newey (Canberra)

Six years ago the author pushed the Stanford LCF system to its limits with experiments in the verification of some LISP programs. At the time of the workshop, one of these experiments was being repeated to gauge the effectiveness of improvements to LCF. This was the proof of correctness of some Pure LISP functions, notably the EVAL function given by McCarthy.

With an interpretive semantics completed and about 50 useful theorems about PureLISP proved (in the new system) it is concluded that the formal description of this language was more elegant and that deduction had proceeded at about three to four times the pace.

The most significant improvement in expressive power was found to be the much richer type structure - one that brings LCF users more

into line with other people operating on Scott foundations. In particular, the use of the new type building operators made the axiomatisation of the domain of S-expressions much cleaner. The talk uses this domain to illustrate a technique of domain definition using type equations that gives greater confidence in the consistency of the theory being developed.

Another advance in LCF which had a significant effect in this experiment was that of polymorphic types. A general theory of equality had already been developed for arbitrary domains and so the usual theorems about equality were available without cost. Furthermore, this and other uses of the theories feature impart much needed structure and modularity to the enterprise.

In developing the theory of Pure LISP, most time was spent proving theorems and so the most valuable advance, that Edinburgh LCF has contributed to the technology of machine checked proof, is the use of a special purpose programming language. The talk illustrates how, in using ML to code proof recipes, enormous amounts of human effort can be saved. So far, some proofs have been shortened by a factor of five (in terms of keystrokes). However, the theorems proved so far have been quite simple; even larger savings are expected on more complex proofs.

NON DETERMINISTIC COMPUTATIONS IN METRIC SPACES

A. Arnold & M. Nivat (Poitiers, Paris)

In the now standard theory of computation in an ordered domain one proves the equivalence between the definition of the computed function as the smallest fixed point of certain functional and the definition of the same function by means of terminating computation sequences of the program at a given point. This equivalence holds when the computation domain is a flat, or discrete, domain in which different defined values are incomparable: the only converging sequences whose terms are all equal, for sufficiently large n , to the limit of the sequence. In such a domain it is clear that any computed value is the result of some terminating computation sequence.

The situation is entirely different if, following D. Scott, one starts computing in a partially ordered domain which contains infinite ascending chains. A computed value may then be the lub of such a chain and as such can well be the result of no finite computation: a typical example is the domain of real numbers, if basic functions are the four arithmetic operations and the initial values are rational numbers, after any finite amount of time one will have computed only a rational number when the result may well be irrational.

We propose in this situation to give a meaning to successful infinite computation sequences which will be said to produce a result and to define the computed function by stating that its value at a given point is the set of results of both finite terminating and infinite successful computation sequences at that point. Obviously doing so one accepts the idea that a computed function is many valued since there is absolutely no reason why all computation sequences would lead to the same result. But indeed many-valued functions were already considered as the normal output of non deterministic programs.

Our point of view thus amounts to consider deterministic programs as special cases of non deterministic programs with the advantage that our result will hold in the general case of non deterministic programs (this was in fact the original motivation of the whole study).

In order to give a meaning to successful computation sequences we found extremely convenient to replace the order structure on the computation domain by a complete metric topology. (This is not at all to say that one cannot use the structure of a cpo to build a theory in many respects analogous to ours and indeed it has been done).

The results we get to are mainly conditions for the equivalence of this definition of the computed function and a mathematical definition by means of fixed point : it happens that in a very natural way one is lead to consider greatest fixed points rather than smallest. Intuitively this corresponds to the idea that, at the beginning of the computation we only know that the value of the computed function lies in a certain range, a priori the whole computation domain and in the course of the computation this range is reduced (may be to just one value but usually to a set of values). This is dual of the point of view expressed by Dana Scott that an a priori undefined initial value gets more and more defined in the course of the computation. We have borrowed for a large part this idea of decreasing range to L. Nolin (in a uncountable number of discussions).

In the course of our study we consider infinite trees for the following reason : algebraic infinite trees which can be generated by a recursive program scheme are at the basis of the theory called "algebraic semantics" of recursive programs.

The algebraic tree thus attached to a program scheme incorporates the whole semantics of the program in the sense that an interpretation being defined as a morphism, the function computed by the program resulting of the interpretation of the scheme is the morphic image of this algebraic infinite tree. Whence many results concerning classes of interpretation and families of computation domains.

Here infinite trees also play a role, in fact a crucial role. For the link between a semantics defined in an ordered structure and the semantics defined in a topological structure lies in the fact that the set of infinite trees $M^\infty(F, V)$ has both an ordered structure and a topological structure which are closely related (in fact an increasing function is order continuous iff it is continuous for the topology). The free complete F-magma $M^\infty(F, V)$ thus appears as the mother structure in which the phenomena of computation can be better described.

SOME THEORETICAL RESULTS CONCERNING PROLOG

K.R. Apt (Rotterdam)

A program in the programming language PROLOG is a set of definite clauses. A definite clause is a construct of the form $A \leftarrow B_1 \& \dots \& B_n$ where A, B_1, \dots, B_n are atomic formulae and is to be understood as $\forall x_1, \dots, x_k (B_1 \& \dots \& B_n \supset A)$. A program is activated by a goal statement which is a construct of the form $\leftarrow A_1 \& \dots \& A_n$ (understood as $\forall x_1, \dots, x_\ell \neg (A_1 \& \dots \& A_n)$).

From a goal

$$\leftarrow A_1 \& \dots \& A_i \& \dots \& A_n \quad \text{where } n \geq 1, 1 \leq i \leq n$$

with selected atomic formula A_i one derives a goal

$$\leftarrow (A_1 \& \dots \& A_{i-1} \& B_1 \& \dots \& B_m \& A_{i+1} \& \dots \& A_n) \Theta$$

for every clause $A \leftarrow B_1 \& \dots \& B_m$ ($m \geq 0$) such that A_i and A are unifiable with most general unifier Θ . A derivation (computation) successfully terminates if an empty clause is derived. Whether a computation successfully terminates depends on the selection rule and on the search algorithm used to search for an empty clause in the tree of goal statements.

The above can be seen as a special type of a resolution (called SLD-resolution). We prove the soundness and completeness of this resolution: a goal $\leftarrow N$ is a root of a tree containing an empty clause iff $S \cup \{\leftarrow N\}$ is inconsistent. Whether a tree with a goal $\leftarrow N$ as a root contains an empty clause is independent on the selection rule used.

A COMPLETE SET OF ASSERTIONS ON CONCURRENT SYSTEMS

Antoni Mazurkiewicz (Warsaw)

A simple concurrent system consisting of a finite number of sequential (nondeterministic) components is considered. Each component is built up from finitely many actions joined together by their input and output places. Two different components have disjoint sets of places; some actions, however, can be common for two (or more) components. The only synchronization between components is the requirement that an action, common for several components is to be performed by them coincidentally (at the same time). This model is very close to the "path expression" model [Campbell, Lauer]. A configuration of the system is a vector of places, one place is taken from one component. The system can move from one configuration to another producing a history [Shields & Lauer 78] - a vector of strings, each string (an individual history) being a sequence of actions generated by one component.

We say that an assertion set for the system is given, if to each place a predicate (over histories) is assigned. An assertion set is said to be sound, if for any configuration, reachable from a given initial configuration, the conjunction of predicates to its places is true for corresponding histories. An assertion set is said to be complete, if whenever the conjunction of predicates assigned to a configuration is true for a vector of strings, this configuration is reachable from the initial one and the vector of strings forms a corresponding history.

It is shown that for any concurrent system as above a sound and complete assertion set can be constructed in such a way that the following requirements are satisfied:

- (i) (1st locality principle). Only local conditions need to be satisfied by predicates to form an assertion set; such a local condition concerns exclusively predicates assigned to places around an action.
- (ii) (2nd locality principle). Values of (history) variables in the scope of predicates assigned to places of a component are not changed when this component is suspended and other components are allowed to run.

Predicates constructed depend not only on individual history variables but also on "skew" history variables intended to keep so-called "told" and "heard" histories. These variables are the only ghost variables in predicates forming assertion sets.

NONDETERMINISTIC KAHN NETWORKS

David Park (Warwick)

Let Σ be a finite alphabet. (K_Σ, \sqsubseteq) is the domain with elements $K_\Sigma = \Sigma^* \cup \Sigma^\omega$ and with the prefix ordering $u \sqsubseteq v \Leftrightarrow (\exists w)v = uw$. $(P(K_\Sigma), \subseteq)$ is the conventional power-set of K_Σ (not the power-domain). A nondeterministic Kahn network over Σ is a system of defining relations, of the form $W_i \in f_i (W_{i_1} \dots W_{i_{n_i}})$, $1 \leq i \leq N$,

where the functions $f_i : K^{n_i} \rightarrow P(K_\Sigma)$ are associated with computing stations and the variables W_i stand for (histories of communication lines). Such a sequence of relations may be used to specify an asynchronous computing network, with devices communicating through buffers. The deterministic case, that each f_i produces just singleton sets, was presented in [Kahn: IFIP 74 Proceedings 471-5].

There it was convincingly argued that the f_i must be continuous functions in the sense of Scott. The operational behaviour was then obtained abstractly as the least fixpoint μf , for the $f : K_\Sigma^N \rightarrow K_\Sigma^N$ got by combining the f_i . A similar analysis for the general case has long presented an open problem, most importantly when some f_i is a (nondeterministic) "fair merge" of sequences, an interleaving guaranteed on infinite sequences to absorb all of each sequence. This paper presents a solution, arguing in four stages:

1. Conceptual Problems: what is the nondeterministic analog of "continuity" ? Let D, E be bounded-complete ω -algebraic domains. Writing $w \leftarrow X$ for $(\exists x) [w < x \& x \in X]$ and using standard notions from [Plotkin: SIAM J. Comp. 5, 452-487]

Defn: $f : D \rightarrow P(D)$ is coherent iff (i) $x \sqsubseteq y \Rightarrow f(x) \sqsubseteq_M f(y)$
(ii) if $w \leftarrow f(x)$ then $(\exists v) [v < x \& w \leftarrow f(v)]$

Theorem: f is coherent iff (i) above & (ii') if $u \in f(x)$, there exist

$v_0 < v_1 < v_2 < \dots, w_0 < w_1 < w_2 < \dots$ with $\sqcup v_i = x$,
 $\sqcup w_i = u$ and $w_i \leftarrow f(v_i)$, all i .

Theorem: if all values of f are singletons then f is coherent iff $\bar{f}(x) = \{g(x)\}$, all x , for some continuous g .

Defn: the finite behaviour of $f : D \rightarrow P(E)$ is the relation $\{e_1 \dots e_n\} \perp_M f(d)$ between finite elements of D and finite sets of finite elements of E .

Theorem: There is a unique maximal coherent $\bar{f} : D \rightarrow P(E)$ with the finite behaviour of some coherent f , such that $\bar{f}(x) \supseteq f(x)$ for all f with that behaviour. \bar{f} can be obtained from the unique continuous $f_M : D \rightarrow \mathcal{P}_M(E)$ (to the Plotkin powerdomain) determined by the behaviour, by taking $\bar{f}(x)$ as the maximal representative of the equivalence class $f_M(x)$, for each x .

2. Defining 'fair merge': Let $\mathcal{R}^3(K_\Sigma)$ be the Boolean algebra of 3-ary relations on K_Σ ; $\nu X. \mathcal{F}(X)$, $\mu X. \mathcal{F}(X)$, for monotone $\mathcal{F} : \mathcal{R}^3(K_\Sigma) \rightarrow \mathcal{R}^3(K_\Sigma)$ are resp. the maximal and minimal fixpoint operators. The graph of 'fair merge' is:

$$\nu X. (\mu Y. (\mathcal{F}_L(Y) \cup \mathcal{F}_R(\mu Z. (\mathcal{F}_L(X) \cup \mathcal{F}_R(Z))))$$

where $\mathcal{F}_L(W) = \{(\lambda, y, \lambda) \mid y \in K_\Sigma\} \cup \{(\sigma x, y, \sigma z) \mid \sigma \in \Sigma, (x, y, z) \in W\}$

$$\mathcal{F}_R(W) = \{(y, \lambda, \lambda) \mid y \in K_\Sigma\} \cup \{(x, \sigma y, \sigma z) \mid \sigma \in \Sigma, (x, y, z) \in W\}$$

λ denoting the null string .

3. Negative length buffers: $W \in \text{merge}(0^\omega, W)$ appears to have anomalous solutions, e.g. $W = w0^\omega$, any $w \in \Sigma^*$. The anomaly disappears on adopting a nonstandard operational notion - permitting any 'queue' on a communication line to take negative length, in which case it is to be taken as a sequence of predictions by the consumer of the producer's future behaviour, which must conform to them if the computation is to be permitted. The set of tuples satisfying the defining relations (in the sense obtained by writing ' ϵ ' for ' \in ') is shown to be the operational behaviour assuming this notion.

4. The hat trick: to restore the standard operational notion of buffer, one can adjust the functions f_i so that negative queues do not occur. Add a new symbol ' \wedge ' to Σ ; $\hat{\Sigma} = \Sigma \cup \{\wedge\}$. If $w \in K_\Sigma$, $\text{doff}(w)$ is the sequence obtained by deleting ' \wedge ' from w . Given $\bar{f} : K_\Sigma^n \rightarrow P(K_\Sigma)$ (coherent) a coherent $\hat{\bar{f}} : K_{\hat{\Sigma}}^n \rightarrow P(K_{\hat{\Sigma}})$ can be obtained satisfying:

(i) if each w_i is finite, $w \in \hat{\bar{f}}(w_1 \dots w_n)$, then $\text{length}(w) > \text{length}(w_i)$

(ii) if $w \in \hat{\bar{f}}(w_1 \dots w_n)$, then $\text{doff}(w) = \bar{f}(\text{doff}(w_1) \dots \text{doff}(w_n))$.

The intended behaviour, with standard operational notions, is then $\{(doff(w_1) \dots doff(w_N)) | w_i \in f_i(w_{i1} \dots w_{in_i})\}$, assuming each f_i is coherent.

[This device was suggested by Wadge].

AN EXERCISE IN PROOF RULE JUSTIFICATION

J.W. de Bakker (Amsterdam)

Abstract not received.

FAIR OR-TREES AND NON DETERMINISTIC PROGRAM SCHEMES

M. Nivat (Paris)

A non deterministic program scheme (ndrps) S is a system of defining equations $S \quad \varphi_i(v_1, \dots, v_{n_i}) = \tau_i, \quad i = 1, \dots, N$

where $\tau_i \in M(F \cup \Phi \cup \{\underline{or}\}, \{v_1, \dots, v_{n_i}\})$

The choice operator \underline{or} is interpreted operationally by giving the rules

$$\underline{or}(t_1, t_2) \rightarrow t_1 \quad \underline{or}(t_1, t_2) \rightarrow t_2$$

which are applicable at all steps of a computation sequence.

The most natural idea to give a semantics of a ndrps S is to first look at the vector of trees $\vec{T} = \langle T_1, \dots, T_N \rangle$ which are generated by S in $M^\infty(F \cup \{\underline{or}\}, V)$ according to the standard theory of deterministic recursive program scheme. Afterwards one tries to break the \underline{or} 's that is to define a morphism $\theta: M^\infty(F \cup \{\underline{or}\}, V) \rightarrow \mathcal{P}(M^\infty(F, V))$. The morphism θ is easily defined on finite trees by

$$\theta(v) = \{v\}$$

$$\theta(f(t_1, \dots, t_n)) = f(\theta(t_1), \dots, \theta(t_n))$$

$$\theta(\underline{or}(t_1, t_2)) = \theta(t_1) \cup \theta(t_2)$$

And we try to extend it by continuity using on $M^\infty(F \cup \{\underline{or}\}, V)$ the topology induced by the Micielski-Taylor metric d and \overline{d} on $\mathcal{P}(M^\infty(F, V))$ the Hausdorff metric defined from d between closed subsets.

A necessary and sufficient condition to define $\theta(T)$, where T is infinite is that

$\forall \epsilon \exists \eta d(t, T) < \eta \text{ and } d(t', T) < \eta \Rightarrow d(\theta(t), \theta(t')) < \epsilon$

We say that θ has bounded oscillation at T .

It is known that if we define $\theta(T)$, for all such T 's, by

$\theta(T) = \lim \theta(t_n)$ for all t_n which converges towards T

then θ is continuous at T . Results from Arnold, Nivat can be applied namely the set of results of all computations of the ndprs S at $\varphi_1(v_1, \dots, v_{n_1})$ equal to $\theta(T_1)$ if θ has bounded oscillation at T_1 . We are lead to give conditions for θ to have bounded oscillation at T and prove that this is true if T is a fair or-tree i.e. satisfies on all infinite branch of T , for all depth n , there exists an F -symbol of depth greater than n . In other words T does not contain an "infinite or-branch".

From this we easily derive that T_1 is fair if and only if S is a Greibach-scheme which is also the condition for the theorem on the equivalence of operational and greatest fixpoint semantics of Arnold, Nivat.

In conclusion:-the Greibach condition on S stems out from a necessary condition of continuity to extend the natural mapping of or-trees into sets of ordinary trees - the use of the metric topology allows us to write local continuity conditions which appear naturally in the theory.

MODULAR DENOTATIONAL SEMANTICS

Peter Mosses (Aarhus)

Denotational semantics is a fine tool for analysing and describing the fundamental concepts of programming languages. However, denotational semantics (DS) has not become popular for giving formal definitions of the meanings of "realistic" languages: the difficulty of reading and writing the definitions seems to increase much faster than the complexity of the languages. I consider that this is due to the lack of modularity in DS descriptions.

The proposed approach is based on the work of Burstall and Goguen: it is to structure DS by specifying a theory corresponding to each "concept" used. For example, the concept of a store (update, contents, new) can be described by one theory, and that of an environment or symbol-table (bind, find) by another theory. More significantly, the operation of sequencing "actions" can be described without prejudging the issue of whether jumps are to be allowed, thus one can avoid the whole semantic specification being permeated by the choice between the "direct" and "continuation" style.

One might "derive" all the individual theories from the Scott theory of domains. However, I would argue that the reader of a DS wants to

be told (at least some of) the laws which the primitive semantic operations obey. This, together with the desire to build directly on Burstall and Goguen's work on algebraic theories, encourages me to keep to equational algebraic specifications, defining semantic operators implicitly by equations expressing their influences on each other.

I would not claim that such implicit specifications are particularly easy to construct. What makes the effort worthwhile is that it should be possible to re-use (most of) the specifications time and time again, in describing different programming languages. I have worked out some algebraic theories which I believe to have this generality.

Once one has put together the specifications of such primitive semantic concepts as are necessary, what remains is to specify the main semantic functions. Here, I keep to the denotational rule: the meaning of a phrase of a programming language should be equated with some composition of the meanings of its sub-phrases. Thus, close contact with DS is maintained, and one can build on the experience gained in the last 10 years. In fact, I would like a standard DS to be an interpretation of the semantic theory.

Another hoped-for feature of the proposed approach is the ability to handle abstract syntax, context-sensitive constraints, syntactic proof-rules and various sorts of semantics just one formalism: that of (structured) algebraic theories. This, together with the conceptual simplicity of algebraic equations, might make formal specifications accessible to those who need them most: the designers, implementors and users of programming languages.

REASONING ABOUT FAIRNESS IN UNSCHEDULED ACTOR-SYSTEMS

A. Kerp, W. Oberdörster, P. Raulefs (Bonn)

A method for reasoning about fairness in the CSSA-actor model of computation is developed.

In this model, we only make very weak and general assumptions about concurrent computations in actor systems:

- (1) Computation is done by a society of independent, concurrent sequential machines we call actors.
- (2) Actors may communicate by sending messages to each other. An actor may only transmit a message to an actor it is acquainted with. Actor systems form communication networks having actors as nodes and acquaintance links as directed edges. Since actors

may be dynamically created or destroyed, and acquaintance links may be communicated among actors, the communication network formed by an actor may dynamically change throughout the course of a computation.

- (3) Transmitting messages takes a positive, but indefinite amount of time. Computation done by actors takes no time at all.

Although assumption (3) is quite realistic, it very much complicates reasoning about concurrent systems of sequential processes when compared to models assuming synchronized communication as a primitive. This is due to the fact that whenever messages are under way, there is no way of knowing anything about the order of their arrival at target actors, so we have to consider all possibilities. It is our intention to investigate concurrent actor systems under this extremely weak assumption first, and then determine in which way more restricted communication primitives reduce the complexity of the semantical model and the difficulty of reasoning about such systems.

Usually, methods are developed to synchronize a system of concurrent processes by adding schedulers to the system. Distributing the actions done by schedulers among individual processes leads to more efficient yet not necessarily less well-structured solutions. We consider the problem of reasoning about fairness without the presence of explicit schedulers.

Our approach consists in looking at particular combinators between actor systems that allow to form more complicated actor systems. The identification merge combinator maps two actor systems into a new one by identifying two actors of same type (= same actor script) in each system.

Based on a denotational semantics we develop a proof theory for static actor systems (no actors are created/destroyed throughout the course of a computation) that allows to establish fairness of such identification merges.

This method is illustrated at a particularly simple solution of the "dining philosophers"-problem: We present two actor scripts for forks and philosophers s.t. for any $n \geq 2$, a fair system of n philosophers and forks can be uniformly generated. The fairness proof consists of considering philosophers/two-forks - triangles, and (1) deriving the fairness of isolated triangles, followed by (2) deriving the fairness of the identification merge of two triangles, and (3) inferring the fairness of the entire system.

This example suggests a uniform proof method for showing fairness of "regularly" constructed actor systems which we call actor crystals.

A TOPOLOGICAL CLOSURE OF FREE X-CATEGORIES

Günter Hotz (Saarbrücken)

x-Categories were introduced in [Ho] 1965 to develop an algebraic calculus for representations of switching functions by boolean networks. The morphisms of free x-categories represent these networks. Here we generalize this theory to an algebraic calculus of infinite networks, which together with a functorial interpretation represent infinite functions. The infinite networks note how to compute the values of the functions for arguments by means of certain elementary operations. In the case of computable functions one may look at these networks as universal enfoldings of programs to compute the function.

Mathematically we construct a topological closure of the free x-categories. This closure is induced by a partial ordering $<$ of the morphism \mathcal{M} of the free x-category \mathcal{F} . $<$ is defined by "derivations" which are generated by substitutions of variables of the generator set of \mathcal{F} by morphisms of \mathcal{F} with the same domain and codomain - in my lecture I restricted to substitution of the empty word. We get a T₀-space $\overline{\mathcal{M}}$. The operations x and o can uniquely be defined on $\overline{\mathcal{M}}$. We construct in this way a continuous x-category $\overline{\mathcal{F}}$ which contains \mathcal{F} as a sub-x-category. The trace of the topology on the "terminal morphisms" seems to be a Hausdorff space.

Now one finds for explicit recursive equations solutions in $\overline{\mathcal{F}}$. One can use the construction of Wolfgang Weidner given in his dissertation to construct an algebraic closure of \mathcal{F} in $\overline{\mathcal{F}}$. His constructions use a finer topology as our's is. But this topology is too fine to construct a space in which all "enfolding" are convergent.

Literature:

- [Ho] G. Hotz: Eine Algebraisierung des Syntheseproblems von Schaltkreisen I und II, EIK Vol I, pp. 185-205, (209-231), 1965
- [W] W. Weidner: Der algebraische und topologische Abschluß freier x-Kategorien, Dissertation an der Universität des Saarlandes, 1977.

REGULAR ALGEBRAS

Jerzy Tiuryn (Aachen)

The idea laying behind regular algebras is that when dealing with fixed point semantics of programming languages one never needs to

work with complete lattices or cpo's equipped with (directed-) continuous maps, but rather with structures where certain lub's exist, and with maps preserving these lub's. More to the point, there are structures (connected with semantics of non-deterministic procedures) with operations being not continuous but still having a very nice property allowing to solve an arbitrary finite system of fixed-point equations, built up from these operations, within ω steps of iterations. Briefly speaking, ordered algebras with this property are called regular algebras.

The lecture was a kind of a survey of results on regular algebras got by the author since 1975/76. The following topics were mentioned during the lecture:

1. The analogy with continuous algebras.
2. Examples of phenomena leading to non-continuous functions having the above-mentioned properties.
3. Free regular algebras and regular polynomials.
4. Varieties of regular algebras defined by equalities as well as by inequalities between regular polynomials.
5. Connections with rational algebraic theories.

ON *-ALGEBRAS

Klaus Indermark (Aachen)

According to their arities the operations on an algebra A form a heterogeneous (many-sorted) algebra: the derived algebra $D(A)$.

As this construction is reiterated in the algebraic analysis of functions defined by auxiliary recursion on higher functional types, one has to use troublesome indices for the carrier sets of $D^n(A)$. Here, we suggest a simple method to overcome that notational burden. Disregarding arities we allow an operation to have an arbitrary number of arguments. This is no restriction insofar as arities can be simulated by means of $\perp \in A$. On the other hand, many operations share this property, as e.g. projections, composition and resolution operators.

To be precise, let $\text{Ops}(A) := \{f \mid f: A^* \rightarrow A\}$ for any set A .

If Ω is a set of operation symbols (without rank!) and $\varphi: \Omega \rightarrow \text{Ops}(A)$, then $A := \langle A; \varphi \rangle$ is called an Ω -algebra. The class Alg_Ω of Ω -algebras contains for each set X $F_\Omega(X)$, freely generated by X .

Similarly, if A is a cpo and $\text{Ops}(A)$ is restricted to continuous operations, we get the subclass Alg_Ω^C of complete Ω -algebras containing the free complete $F_\Omega^C(X)$ for any generating set X . As in the ranked case, the carrier sets $F_\Omega(X)$ and $F_\Omega^C(X)$ are representable by finite and in-

finite trees. However, labelling by operation symbols is unrestricted.

If $X = \{x_1, x_2, \dots\}$, then any $t \in F_\Omega^C(X)$ represents an operation on each $A \in \text{Alg}_\Omega^C$: the derived operation of t on A .

This derivation operator $\text{derop}_A : F_\Omega^C(X) \rightarrow \text{Ops}(A)$ is defined by

$\text{derop}_A(t)(a_1 \dots a_n) := \overline{[a_1 \dots a_n, \perp]}(t)$ where $[a, \perp](x_i) := \underline{\text{if } i \leq \text{lg}(a)}$
 $\text{then } a_i \text{ else } \perp$ and $\overline{\quad}$ denotes the homomorphic extension.

Next, we consider rational schemes, since they play a central role in the analysis of higher type recursion and, moreover, they serve for studying iterative control structures.

Extending Ω to $R(\Omega) := \Omega \cup \{\Pi_i \mid i \geq 1\} \cup \{\Phi, \text{IR}\}$ we define

$\text{Rat}_\Omega := F_{R(\Omega)}^C$ as the algebra of rational Ω -schemes. For any interpretation $A \in \text{Alg}_\Omega^C$ we construct $\text{Ops}(A) := \langle \text{Ops}(A); \dots \rangle \in \text{Alg}_{R(\Omega)}^C$

where F' means left-composition, Π_i projection, Φ composition and IR resolution of regular equations with parameters. The unique $R(\Omega)$ -homomorphism $h_{\text{Ops}(A)} : \text{Rat}_\Omega \rightarrow \text{Ops}(A)$ describes the semantics of Rat_Ω with respect to A .

Then, we extend $F_\Omega^C(X)$ to $F_\Omega^C(X)^r \in \text{Alg}_{R(\Omega)}^C$ such that derop_A becomes an $R(\Omega)$ -homomorphism. Hence, the semantics $h_{\text{Ops}(A)}$ splits into

$\text{Rat}_\Omega \xrightarrow{\text{tree}} F_\Omega^C(X)^r \xrightarrow{\text{derop}_A} \text{Ops}(A)$. A rational tree $t \in \text{tree}(\text{Rat}_\Omega)$

represents an equivalence class of rational schemes. From their regular and equational normal forms we infer degrees measuring the iteration complexity of a rational tree. We briefly discuss some relationship to the star-height hierarchy of regular languages and the Kosaraju-hierarchy of repeat-exit schemes that has also been studied by G. Cousineau.

CERT-DERI

2, av. Edouard Belin

B.P. 4025

31055 - TOULOUSE cédex, FRANCE

Tél. : (61) 53.11.88

Télex : ONECERT 521596 F

H. GALLAIRE - J.M. NICOLAS

WORKSHOP ON

FORMAL BASES FOR DATA BASES

- Location : ONERA-CERT, TOULOUSE, FRANCE
- Date : December 12-14, 1979
- Topics : Include all formal studies related to data bases. For example (but not exclusively) the following subjects are relevant :
- formal definition of data bases models,
 - theory of relations : decomposition of relations, axiomatization of dependencies, ...
 - inferences in data bases,
 - formal aspects of integrity constraints,
 - formal languages,
 - ...
- Structure : . Half an hour talks and informal discussions,
. Participation limited to 30 people.
- Accommodation : It might be possible to house all the attendees on the ENSAE-CERT campus.
- Deadlines : June 15th, 1979 : one page abstract.
Notification to authors by mid-july.
November 1st, 1979 : the final paper.
- Preprints and proceedings :

Following the organization of the former CERT workshop on "Logic and data bases" :

- Papers will be selected for the workshop only on the basis of their adequation to the topics and within the limit of about 30.
- After reviewing these papers through a standard refereeing process, a book will possibly be published with those selected papers. Contacts with a publisher are taken by Jack Minker.

SOCIAL EVENTS

A number of social events will be held for participants. Tours, visits, half day and full day excursions, cocktail at the beginning and banquet at the closing of the Symposium, are planned.

ACCOMMODATION

Due to heavy demand for hotel accommodation in August in Dubrovnik, participants are advised to make hotel reservations well in advance. Accommodation in hotel Palace can be secured if the registration form together with details concerning the desired accommodation is returned before June 15, 1979.

INVITED SPEAKERS

PATRICK H. WINSTON, Professor, Artificial Intelligence Laboratory, Massachusetts Institute of Technology, Cambridge, Massachusetts 02139 USA.

BERTHOLD K. P. HORN, Professor, Artificial Intelligence Laboratory, Massachusetts Institute of Technology, Cambridge, Massachusetts 02139 USA.

GERALD SUSSMAN, Professor, Artificial Intelligence Laboratory Massachusetts Institute of Technology, Cambridge, Massachusetts 02139 USA.

RANDALL DAVIS, Professor, Artificial Intelligence Laboratory, Massachusetts Institute of Technology, Cambridge, Massachusetts 02139 USA.

RAJ REDDY, Professor, Department of Computer Science, Carnegie-Mellon University, Pittsburgh, Pennsylvania 15213 USA.

ERIK SANDEWALL, Professor, Informatics Laboratory, Uppsala University, Uppsala, Sweden.

MARCO SOMALVICO, Professor, Milan Polytechnic Artificial Intelligence Project, MP-AI Project, 20133 Milano, Italy.

The Program Planning Committee of the Summer Seminar is headed by Patrick H. Winston, M. I. T., Massachusetts, USA.

For registration and details write to

SLOBODAN M. JAUKOVIĆ
Center for Advanced Studies
Kneza Miloša 9/IV
P. O. Box 356
11001 Belgrade, Yugoslavia

BASIC FACTS ABOUT C.A.S. SEMINARS

The Center for advanced Studies (CAS) was established by the Yugoslav Committee for Electronics and Automation as a forum for intellectual contacts among scholars and young research workers of different countries working in the fields of systems analysis and computer science.

One of the most important activities of the Center is the organization of international summer seminars on advanced topics of systems and computer sciences. These seminars, which are held every summer in Dubrovnik, are organized as a two week series of tutorial lectures, case studies, workshops and round table discussions with an international team of 5 to 10 outstanding lecturers and 40 to 80 participants from universities, research institutions and industrial organizations.

Focusing the attention of participants on new ideas and areas of research, the summer seminars contribute towards:

- exchange of views and experiences among leading international experts and scholars in an informal and stimulating ambient;
- orientation and guidance for young research workers in new areas of system and computer sciences;
- transfer of experiences in the application of new concepts and methods through case studies and workshops.

Since its foundation the Center has organized 13 international seminars on the following topics:

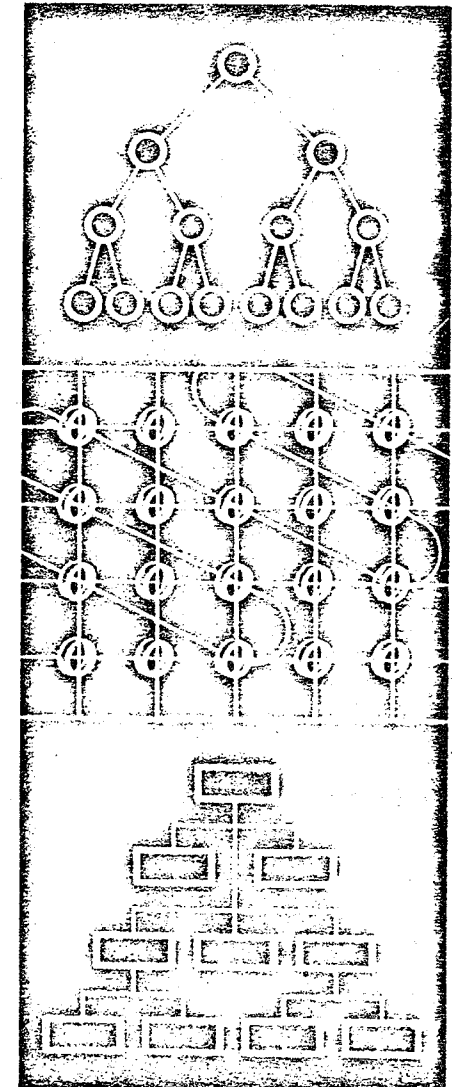
- Optimal Control and Hybrid Computation (1966)
- Large-Scale System Theory and Engineering (1967)
- Mathematical Bio-Sciences (1968)
- Theory of Organisational Systems (1969)
- Artificial Intelligence (1970)
- Organizations and Computers (1971)
- System Approach to Urban Problem-Solving (1972)
- Design Automation of Electronic Systems (1973)
- System Approach to Management and Control of Water Resources (1974)
- System Approach to Environmental Problems (1975)
- Managing Urban Systems (1977)
- Technology in the Health Care System (1978)
- Global and Large-Scale System Models (1978)

These seminars were attended by approximately 700 participants from 20 countries and were conducted under the chairmanship of A. V. BALAKRISHNAN, R. BELLMAN, N. BURAS, VEN TE CHOW, R. KALMAN, L. K. KIRCHMAYER, R. LEVIAN, M. D. MESAROVIC, G. ORLOB, B. RAPHAEL, E. SAVAS, G. SZEGO, H. SIMON, H. WEED and L. ZADEH.

ARTIFICIAL INTELLIGENCE

International Summer Seminar
August 6—17, 1979
Dubrovnik, Yugoslavia

II Announcement



SCOPE OF THE SEMINAR

Artificial Intelligence (AI) has begun to develop about twenty years ago as the discipline which studies the theoretical foundations, the main methodologies, the design criteria and the construction of programs which allow the computer to perform activities that are usually considered as exclusive of the man.

The two week Seminar will be starting with the basic problem of how to obtain, to represent and to manipulate knowledge about a particular domain. Having discussed that, attention will be shifted to the problem solving paradigms considered as the computer's capability to find automatically the solution of a suitably represented problem.

Attention also will be devoted to an objective critique of ongoing applications such as natural language understanding, machine vision, manipulation and robotics. In the area of programming methodology various task oriented languages will be surveyed as well as theorem proving systems, and further possibilities in improving LISP programs.

A challenging question for research workers in the area might be to explore the relationship between AI from one side and biology, psychology, philosophy and theory of computation from the other side.

The work of the Seminar will be organized through tutorial lectures, case studies, workshops and round table discussions with an internationally recognized team of experts and participants from universities, research institutions and industrial organizations.

SPONSORED BY

University of Belgrade, Massachusetts Institute of Technology and the Yugoslav-American Committee for Educational Exchanges.

TIME AND PLACE

The Summer Seminar will be held from Monday, August 6 through Friday, August 17, 1979 in Dubrovnik, Yugoslavia.

ORGANIZED BY

Center for Advanced Studies, Kneza Miloša 9, P. O. Box 356, 11001, Belgrade, Yugoslavia.

REGISTRATION

The number of participants will be limited and priority will be given to those who register before June 15, 1979. Those wishing to participate are requested to complete registration form enclosed. Registration fee in the amount of US \$ 180 (2800 dinars for Yugoslav participants) should be transferred to Beogradska banka in favour of the Yugoslav Committee for ETAN, account No. 60802-620-1-1-72:000-421-01062. Payments in Yugoslav dinars should be transferred to account No. 60802-678-12115.

PROGRAMME OF THE SEMINAR

		Monday, August 6	
		Registration	
		Overview	
	8.30—10.30	Knowledge Representation (for reasoning)	P. WINSTON
	11.00—12.30	Knowledge Representation (for vision)	P. WINSTON
	17.00—20.00		B. HORN
		Tuesday, August 7	
		Programming Methodology	E. SANDEWALL
	9.00—12.00	Programming Methodology	E. SANDEWALL
	17.00—20.00		
		Wednesday, August 8	
		Natural Language	P. WINSTON
	9.00—12.00	Speech Understanding	R. REDDY
	17.00—20.00		
		Thursday, August 9	
		Vision	B. HORN
	9.00—12.00	Vision	R. REDDY
	17.00—20.00		
		Friday, August 10	
		Robotics	M. SOMALVICO
	9.00—12.00	Manipulation	B. HORN
	17.00—20.00	PANEL I	
		Monday, August 13	
		Programming Methodology	G. SUSSMAN
	9.00—12.00	Programming Methodology	G. SUSSMAN
	17.00—20.00		
		Tuesday, August 14	
		Symbol Manipulation a la LISP	P. WINSTON
	9.00—12.00	PANEL II	G. SUSSMAN
	17.00—20.00		
		Wednesday, August 15	
		Production Systems	R. DAVIS
	9.00—12.00	Expert Problem Solving	G. SUSSMAN
	17.00—20.00	Knowledge Engineering	R. DAVIS
		Expert Problem Solving	G. SUSSMAN
		Thursday, August 16	
		Applications of Knowledge Engineering	R. DAVIS
	9.00—12.00	Applications of Knowledge Engineering	R. DAVIS
	17.00—20.00		
		Friday, August 17	
		PANEL III	
	9.00—12.00	Summary-Adjournment	
	17.00		

NEUE PAPIERE

On foundations of reasoning with uncertain facts and vague concepts.

Peter Schefe

Bericht Nr. 56

Fachbereich Informatik

Universität Hamburg

Schlüterstr. 70, 2000 Hamburg 13

SUMMARY

"Fuzzy sets theory" and "fuzzy logic" based on the former have become of rapidly increasing interest. The foundations, however, are still disputed. Especially the definitions of some set operations and logical connectives appear to be somewhat arbitrary. The relations of "fuzziness" to "probability" and "possibility" are not yet clear. This paper contains an outline of a probabilistic foundation of multi-valued ("fuzzy") reasoning. The fundamental concept is "agreement probability". It is shown that some undesirable consequences of "fuzzy logic", e.g., that tautologies of propositional calculus are not preserved can be avoided. A proposal for alternative definitions of "degree of membership" and operations on membership-graded sets are given.

"Fuzziness" is interpreted as a subjectivistic concept, i.e., subjective uncertainty pertaining to the truth of a proposition. An important consequence thereof is that, from a graded agreement associated with a conjecture, an agreement degree pertaining to its negation cannot be computed.

According to this foundation Shortliffe's model of medical diagnosis is reviewed as an application paradigm. There is no fundamental disagreement with Shortliffe's interpretation.

However, Zadeh's "linguistic modelling" is shown to be inadequate. "Fuzziness" of linguistic concepts is interpreted as uncertainty of the applicability of a predicate in a given situation. This leads to the conclusion that the definition of derived concepts, especially, of hedged expressions referring to continuous scales cannot be modelled using Zadeh's fuzzy set operations. Experimental findings of Hersh and Caramazza support an alternative interpretation.

Particularly, Zadeh's conjecture that truth values can be equated with membership degrees is shown to be inadequate. Alternative interpretations of the linguistic phenomena considered and of the sorites paradox are given. Especially the metalinguistic character of the phenomena is emphasized. It is argued that "vagueness" and "uncertainty" should be clearly distinguished as well as "possibility" and "applicability". Suggestions are made how underlying measuring scales and orderings of objects are used in reasoning processes involving vague concepts.

W. Bibel. On Matrices with Connections.

Abstract.

In this paper theorem proving is considered as the problem of verifying that each path through a matrix consisting of a set of clauses can be made complimentary. Introducing connections to such a matrix the following three results are derived from that conceptual basis. First, a simple, one-page proof for the consistency and completeness of the connection graph procedure is given. Second, a macro-simplification rule for the preparatory step of any ATP-method is defined which properly reduces any given matrix whenever it applies, similarly as the deletion or subsumption rule. It can be regarded as a generalization to arbitrary clauses of the well-known fact that sets of two-literal clauses can be decided quickly. Finally, in view of the relation between resolution-based and natural deduction-based methods a constructive transformation is specified which explicitly relates each resolution step to a pair of complementary literals in an axiom of a natural deduction and vice versa. Although the paper is restricted to the ground case it is obvious that all results can be easily lifted to the general case in the usual way.

The full text is available as report No. 79,
Institut für Angewandte Informatik und Formale Beschreibungs-
verfahren, Universität Karlsruhe, 7500 Karlsruhe 1,
Postfach 6380.

Ebenda erhältlich:

W. Bibel. Syntax-directed, semantics-supported program synthesis. (Rept. No. 78)

W. Bibel. Synthese und Analyse von Algorithmen.

I. Deduktion von Algorithmen (Rept. No. 80)

Universität Dortmund, Abt. Informatik
Pstfach 500500, 4600 Dortmund 50

W. Coy, J. Pflüger. Identification of L-Systems in the Limit.
Forschungsbericht Nr. 77, 1979.

Abstract

The principal possibilities of syntactical inference for the main classes of L-languages, especially OL- and IL-languages, are investigated. Sharp bounds on the identifiability by texts and informants (examples and combinations of examples and counter-examples) for the language classes under consideration are derived. Main results are:

- the identifiability of DOL by text alone;
- the non-identifiability by text of POL and DPIL;
- the general non-identifiability of DIL.

The results were contrasted to the known results on Chomsky-languages. These results constitute a frame for the search for practical solutions of the syntactic inference problems for developmental systems.

The notion of an infinite language chain with fixpoint is introduced and used successfully for the proofs of the non-identifiability by texts as well as by informants. This notion simplifies also the proofs for such results within the Chomsky-hierarchy of languages.

Forschungsgruppe CUU. Projekt PROKOP.
Frankfurterstr. 24, 6100 Darmstadt

Arbeitsmaterialien zum interaktiven Problemlösen mit Computerhilfe.
Band 2: Problemaufgaben
Teil 4: Künstliche Intelligenz
von H.-D. Böcker und G. Fischer

Stanford University, Dept. of Computer Science
Attn. Publications Coordinator
Stanford, CA 94305, USA

T. M. Mitchell. Version Spaces: An Approach to Concept Learning.
STAN-CS-78-711. 216pp (Thesis)

M. Georgeff. A Framework for Control in Production Systems.
AIM-322

E. H. Shortliffe, B.G. Buchanan, E.A. Feigenbaum.
Knowledge Engineering for Medical Decision Making: A Review of Computer-
Based Clinical Decision Aids. STAN-CS-79-723.

J. McCarthy. 1st Order Theories of Individual Concepts and Propositions.
AIM-325

J. McCarthy. Ascribing Mental Qualities to Machines.
AIM-326.

R. E. Filman. The Interaction of Observation and Inference.
AIM-327. (Thesis)

MIT AI Publications
MIT AI Lab
545 Tech. Sq., Room 821
Cambridge, MA 02139, USA

H. E. Shrobe, R.C. Waters. Computer Aided Evolutionary Design for Software
Engineering. No. 506 (Jan. 79)

H.E. Shrobe, R.C. Waters, G.J. Sussman. A Hypothetical Monologue Illustrating
the Knowledge Underlying Program Analysis. No. 507 (Jan. 79)

D. Weinreb, D. Moon. Lisp Machine Manual. (Jan 79)

SONSTIGE MITTEILUNGEN

Institut für linguistische Datenverarbeitung (ILDV), Münsingen/Rorschach

Das seit 1976 in Münsingen bei Bern ansässige Institut für linguistische Datenverarbeitung (ILDV) ist Mitte April 1979 nach Rorschach SG (Bodensee) umgezogen. Infolge der erweiterten Tätigkeit trägt es einen neuen (zusätzlichen) Namen:

Institut für nichtnumerische Informationsverarbeitung (INIV)
Thaler Straße 8
Postfach 409
CH-9400 Rorschach SG
Schweiz

Telefon: 071 42 36 38.

Der dem Institut angegliederte Verlag, der Verlag Linguistik, hat seinen Sitz ebenfalls nach Rorschach verlegt.

Der Verlag Linguistik veröffentlicht demnächst ein Buch unter dem Titel "Nichtnumerische Informationsverarbeitung", das unter anderem folgende Themen behandeln wird: künstliche Intelligenz, Computerschach, Computerkunst, Simulation von Entscheidungsprozessen, medizinische und chemische Beratungssysteme.

Ferner ist ein für die Technische Hochschule Zürich verfasster Forschungsbericht erhältlich:

H.E. Bruderer, Simulation von Entscheidungsprozessen. Künstliche Intelligenz. Verlag Linguistik, Rorschach 1978, 63 Seiten, 200.- SFr. (hoher Preis infolge der sehr geringen Auflage, Bezug nur über das Institut für nichtnumerische Informationsverarbeitung).

Professor Nicholas V. Findler, of the State University of New York in Buffalo, will spend his sabbatical year at the Free University of Amsterdam and the University of Amsterdam in 1979-80. He would be glad to visit other universities and to give lectures on various research topics. Some of the lecture titles are as follows:

- (1) A Survey of Artificial Intelligence -- Problems Solved and Outstanding.
- (2) Some Novel Approaches to Machine Learning.
- (3) Studies in Machine Cognition Using the Game of Poker.
- (4) Teaching Strategies On-Line to an Advice Taker/Inquirer System.
- (5) A Heuristic Information Retrieval System Based on Associative Networks.
- (6) The Problems of Time, Retrieval of Temporal Relations, Causality and Co-Existence.
- (7) On the Role of Exact and Non-Exact Associative Memories in Human Machine Information Processing.
- (8) An Interactive Environment for the Simulation of Several Robots Which Can Learn, Plan Their Actions and Co-Exist.

Prof. Findler's addresses are:

Before July 15, 1979:

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

Between Aug. 1, 1979 and July 1, 1980:

Department of Mathematics
Free University
1007 MC Amsterdam
De Boelelaan 1081
Holland Tel.: (020) 548-2410

or

Department of Mathematics
University of Amsterdam
Roetersstraat 15
1018 WB Amsterdam
Holland Tel.: (20) 522-2200