Rostocker Mathematisches Kolloquium

Heft 10

Wilhelm-Pieck-Universität
Rostock

ROSTOCKER MATHEMATISCHES KOLLOQUIUM

Heft 10

1978

Wilhelm-Pieck-Universität Rostock
Sektion Mathematik

Redaktion: Abt. Wissenschaftspublizistik der Wilhelm-Pieck-Universität Rostock, 25 Rostock, Vogelsang 13/14
Fernruf 369 577

Verantwortlicher Redakteur: Dipl.-Ges.-Wiss. Bruno Schrage
Fachredakteur: Doz. Dr. sc. nat. Gerhard Maeß,
Sektion Mathematik

Herausgegeben von der Wilhelm-Pieck-Universität Rostock unter Genehmigungs-Nr. C 48/79

Druck: Ostsee-Druck Rostock, Werk II

Inhalt

Vorwort

Die Tagung "Diskrete Mathematik und ihre Anwendungen in der Mathematischen Kybernetik" fand vom 23. - 26. April 1978 als 10. Arbeitsseminar auf dem Gebiet der Diskreten Mathematik (im Rahmen des Akademieabkommens zwischen der DDR und der UdSSR) statt. Veranstalter der Tagung waren die Mathematische Gesellschaft der DDR und die Sektion Mathematik der Wilhelm-Pieck-Universität Rostock. Die vorangegangenen Arbeitsseminare wurden an folgenden Orten durchgeführt:

1. Universität Rostock, 23. - 29. 10. 1972 (Mitteilungen der MGdDDR, Heft 2/3 - 1973),
2. Friedrich Schiller-Universität Jena, 8. - 13. 10. 1973,
3. Rechenzentrum der Akademie der Wissenschaften der UdSSR Moskau, 4. - 7. 2. 1974,
4. Humboldt-Universität Berlin, 3. - 10. 11. 1974 (EIK 11 (1975), 571 - 660,
5. Wilhelm-Pieck-Universität Rostock, 20. - 26. 4. 1975 (Mitteilungen der MGdDDR, Heft 2/3 - 1977),
6. Rechenzentrum der Akademie der Wissenschaften der UdSSR Moskau, 7. - 14. 12. 1975,
7. Rechenzentrum der Akademie der Wissenschaften der UdSSR Moskau, 6. 6. - 13. 6. 1976,
8. Friedrich-Schiller-Universität Jena, 25. - 30. 10. 1976,
9. Humboldt-Universität Berlin, 23. - 27. 10. 1978.

Schwerpunkte der diesjährigen Tagung waren die folgenden Gebiete (in Klammern die Anzahl der Vorträge):

Funktionensysteme und diskrete Analysis (Kombinatorik) (16),
Kompliziertheitstheorie (5),
Graphentheorie (7),

Automatentheorie und Sprachen (4),
Algorithmentheorie (4),
Erkennungstheorie (4).
Insgesamt wurden 42 Vorträge gehalten, davon 16 Hauptvorträge. Die meisten Vorträge stellten neuere oder neueste Forschungsergebnisse vor. Daneben gab es aber auch entsprechend der Planung der Veranstaltung Überblicksvorträge. Zum Abschluß der Tagung fand eine eineinhalbstündige Diskussion der Teilnehmer zu dem Thema "Anwendungen der Diskreten Mathematik" statt. Die 96 Teilnehmer (davon 25 aus der UdSSR, der VR Polen, der CSSR und der SR Vietnam) kamen aus dem Hochschulwesen und von Akademieeinrichtungen.

Auf der Tagung zeigte sich, daß ursprüngliche Schwierigkeiten in der Profilbestimmung der Diskreten Mathematik durch die Arbeit der letzten Jahre in erheblichem Maße abgebaut wurden. Dafür spricht auch, daß bei aller Spezifik des jeweiligen Tagungsortes die Thematik der bisherigen 10 Tagungen ziemlich konstant war.

Im Rostocker Mathematischen Kolloquium werden in zwei Heften Manuskripte über auf der Tagung gehaltene Vorträge veröffentlicht. Zu einigen Vorträgen der Tagung steht uns kein Manuskript zur Verfügung, da diese Ergebnisse an anderer Stelle publiziert werden oder wurden.

Prof. G. Burosch

Rostock. Math. Kolloq. 10, 7 - 10 (1978)

Andreas Albrecht

Über die Kompliziertheit der Realisierung Boolescher Funktionen durch Schemata aus Funktional- und Verbindungselementen

Es wird die Kompliziertheit der Realisierung Boolescher Funktionen durch Schemata aus Funktionalelementen für den Fall betrachtet, daß die Anzahl der Verbindungen der Funktionalelemente und deren Lage in der Ebene die Kompliziertheit der Realisierung beeinflussen. Die entsprechende Konkretisierung, die hier verwendet wird, hat erstmals S. S. Kravcov in /1/ untersucht. Wir betrachten Rechtecke, die vollständig in Quadrate der Kantenlänge 1 eingeteilt sind. Jedes Quadrat kann mit einem der drei Funktionalelemente (FE) bzw. einem der zwei Verbindungselemente (VE) belegt werden (s. Abb. 1). Die Belegung muß jedoch nicht vollständig sein. Für die drei FE gibt es in Abhängigkeit von der Lage der Richtungspfeile zur unteren Kante vier verschiedene Möglichkeiten der Einsetzung in ein Quadrat.

Als F-V-Schemata werden diejenigen rechteckigen Schemata mit einer Belegung aus FE und VE bezeichnet, die einen gerichteten azyklischen Graphen repräsentieren, wenn die FE als Knoten interpretiert und für die Verbindungen der FE untereinander und für die Zuführung von Variablen zu den FE gerichtete Kanten eingeführt werden. Der gerichtete azyklische Graph muß den Bedingungen genügen, die ein Schema aus Funktionalelementen (s. /2/) erfüllt. Eine ausführliche Definition der F-V-Schemata findet man in /1/.
Entsprechend der Zuordnung von Variablen aus $X^n = \{x_1, x_2, \ldots, x_n\}$

zu den Eingangsknoten des F-V-Schemas S und der Auswahl eines Ausganges realisiert S eine Boolesche Funktion $(f(x_1,x_2,\ldots,x_n)$. Das F-V-Schema bezeichnen wir mit S_f. $h(S_f)$ sei die Höhe und $l(S_f)$ die Länge von S_f.

Definition 1:
$$L_G(S_f) = h(S_f)l(S_f) ;$$
$$L_F(S_f) = \text{Anzahl der FE in } S_f ;$$
$$L_V(S_f) = L_G(S_f) - L_F(S_f) .$$

Die Funktion $\varphi: Nz \to Nz^+$ sei monoton wachsend und $\sum_n$ sei die Menge aller F-V-Schemata, die Boolesche Funktionen aus $\mathcal{P}^n$ realisieren. Wir definieren folgende Shannonfunktion:

Definition 2:

$$L_G(n / \varphi(n)) = \begin{cases} \max_f \min L_G(S_f) & \text{, falls es für jedes } f \in \mathcal{P}^n \text{ ein } S_f \in \sum_n \\ L_F(S_f) \leq \varphi(n) & \text{mit } L_F(S_f) \leq \varphi(n) \text{ gibt;} \\ \infty & \text{andernfalls.} \end{cases}$$

Kravcov hat in /1/ gezeigt, daß $L_G(n/\text{ const.}2^n) \asymp 2^n$ gilt (allerdings ohne besondere Berücksichtigung der Anzahl der FE). Wir untersuchen den Verlauf von $L_G(n/\varphi(n))$ in Abhängigkeit von $\varphi(n)$ für $\frac{2^n}{n} \lesssim \varphi(n) \lesssim \text{const. } 2^n$.
Eine ausführliche Darstellung der in dieser Mitteilung aufgezeigten Resultate findet man in /7/.

Obere Abschätzung

Die Realisierung einer beliebigen Booleschen Funktion durch ein F-V-Schema erfolgt mit Hilfe einer Kombination der von Lupanov entwickelten asymptotisch-optimalen Synthesemethode /2/ mit der von Kravcov in /1/ angegebenen Synthese.
Die Funktionen $\psi, \theta: Nz^+ \longrightarrow R^+$ seien monoton wachsend.

Satz 1: Für jedes $\varepsilon > 0$ und $\psi(n), \theta(n)$ mit $\psi(n) \leq n$, $\frac{\theta(n)}{\psi(n)} \longrightarrow 0$ und $\theta(n) \geq (1+\varepsilon)\log \psi(n)$ gilt für hinreichend große n

$$L_G(n/\frac{2^n}{\psi(n)}(1+\delta(n))) \leq 0\left(\frac{2^{n+\psi(n)}}{(\psi(n)-\theta(n))2^{\theta(n)}}\right)$$

wobei $\delta(n) = 0\left(\max\left\{\frac{\theta(n)}{\psi(n)-\theta(n)}, \frac{1}{\psi^{\varepsilon/2}\, 2^{n-\psi}}\right\}\right)$.

Untere Abschätzung

Für die untere Abschätzung wird die Anzahl nichtisomorpher Graphen mit m Knoten abgeschätzt, die F-V-Schemata mit einer bestimmten Anzahl von Verbindungselementen entsprechen. Diese Abschätzung beruht auf der Kenntnis der Anzahl spezieller ebener paarer Graphen mit einer von der Anzahl der Verbindungselemente im F-V-Schema abhängigen Anzahl von Überschneidungen. Daraus gewinnt man durch die übliche "Mächtigkeitsuntersuchung" den

Satz 2: $\psi: Nz^+ \longrightarrow R^+$ sei monoton wachsend mit $\psi(n) \leq n$. Es gibt ein $c \in R^+$, so daß

$$L_G\left(n \,/\, \left[\frac{2^n}{\psi(n)}\right]\right) \geq c \cdot \frac{2^n + \psi(n)}{\psi(n)}, \quad n \longrightarrow \infty.$$

Dabei geht für jedes $\varepsilon > 0$ der Anteil der Funktionen mit

$$L_G\left(n \,/\, \left[\frac{2^n}{\psi(n)}\right]\right) \leq (1-\varepsilon) \cdot c \cdot \frac{2^n + \psi(n)}{\psi(n)}$$

gegen Null bei $n \longrightarrow \infty$.

Bemerkung: Überträgt man die Ergebnisse der Arbeit /3/ auf den zweidimensionalen Fall, kann daraus direkt die Aussage von Satz 2 für $\psi(n) \sim n$ abgeleitet werden (siehe auch /4/).

Literatur

/1/ Кравцов, С.С., Реализация функций алгебры логики в одном классе схем из функциональных и коммутационных элементов, Сб. "Проблемы кибернетики", вып. 19, М., Наука, 1967, 285-292.

/2/ Лупанов, О.Б., О синтезе некоторых управляющих систем, Сб. "Проблемы кибернетики", вып. 10, М., Наука, 1963, 63-97.

/3/ Колмогоров, А.Н., Барздинь, Я.М., О реализации сетей в 3-мерном пространстве, Сб. "Проблемы кибернетики", вып. 19, М., Наука, 261-268.

/4/ Ачасова, С.М., Бандман, О.Л., Матричный метод синтеза комбинационных схем и логических преобразователей конечных автоматов, Изв. АН СССР, сер. Техн. киб., № 6, 1975, 99-106.

/5/ Шкаликова, Н.А., О сложности реализации некоторых функций клеточными схемами, Сб. работ по мат. кибернетики, М., ВЦ АН СССР, 1976.

/6/ Альбрехт, А., О схемах из клеточных элементов, Сб. "Проблемы кибернетики", вып. 33, М., Наука, 1978, 209-214.

/7/ Albrecht, A., Zur Kompliziertheit der Realisierung Boolescher Funktionen durch F-V-Schemata, eingereicht in EIK.

eingegangen: 15.9.1978

Anschrift des Verfassers:

Dipl.-Math. Andreas Albrecht
Humboldt-Universität zu Berlin
Sektion Mathematik
1086 Berlin
PSF 1297

Rostock. Math. Kolloq. 10, 11 - 22 (1978)

Andreas Brandstädt

Über den Einfluß eines zusätzlichen Einweg-Eingabebandes bei Entscheidungen und Aufzählungen mit beschränktem Regime sowie beschränkter Rückkehr bzw. beschränkter dualer Rückkehr auf einbändrigen Turingmaschinen

1. Einleitung

Die vorliegende Arbeit vergleicht die Klassen der entscheidbaren und aufzählbaren Mengen auf gewöhnlichen einbändrigen einköpfigen Turingmaschinen mit bzw. ohne Einweg-Eingabeband im Falle beschränkter Kompliziertheit bezüglich drei verschiedener Maße:
Das allgemein bekannte Regimekompliziertheitsmaß (vgl. /5/) ist durch die Maximalzahl von Feldgrenzenüberschreitungen einer Berechnung (maximale Spurlänge) gegeben. Das Rückkehrmaß wurde 1975 von Wechsung (/6/, vgl. auch /7/) definiert und beinhaltet die maximale Länge der Restspuren von der ersten Änderung an. In Anlehnung an eine Arbeit von Hibbard läßt sich die duale Rückkehr definieren als maximale Länge der Anfangsspuren bis zur letzten Änderung (vgl. /3/), d.h., ist die duale Rückkehr eines Berechnungsprozesses durch f(n) beschränkt, so darf auf keiner Zelle nach dem f(n)-ten Besuch dieser Zelle ihr Inhalt noch verändert werden.
Während für viele Maschinenkonzepte und Kompliziertheitsbeschränkungen entscheidbare Mengen auch aufzählbar sind und für das Regime bekannt ist, daß die in beschränktem Regime auf einbändrigen Turingmaschinen (ohne Eingabeband) entscheidbaren Mengen ebenso wie die aufzählbaren Mengen genau die Klasse der regulären Mengen liefern sowie die auf dem gleichen Maschinenmodell in beschränkter Rückkehr entscheidbaren bzw. aufzählbaren Mengen genau die Klasse der kontextfreien Sprachen ergeben (/7/, /2/), zeigen wir, daß sich diese Situation

bei Hinzunahme eines Einweg-Eingabebandes ändert: Die Klasse der aufzählbaren Mengen bleibt erhalten, aber die Klasse der entscheidbaren Mengen vergrößert sich. In diesen Fällen gibt es also mehr entscheidbare als aufzählbare Mengen.

2. Grundkonzepte und Bezeichnungen

Wir setzen die Kenntnis des Konzeptes der Turingmaschine voraus und verwenden folgende Bezeichnungen:

DTM (TM) deterministische (nichtdeterministische) einbändrige einköpfige Turingmaschine

1 DTM (1TM) - deterministische (nichtdeterministische) einbändrige einköpfige Turingmaschine mit zusätzlichem Einweg-Eingabeband (Einweg-Turingmaschine)

Die hier verwendeten Kompliziertheitsmaße erhalten die Bezeichnung: Regime - C, Rückkehr - V, duale Rückkehr - H.

Wir sagen, daß die TM M eine Sprache L in durch f(n) beschränkter Kompliziertheit entscheidet, falls w genau dann Element von L ist, wenn eine akzeptierende Berechnung existiert (d.h. eine solche Berechnung, auf der M in einen Finalzustand gelangt), für die $\Phi_M(w) \leq f(|w|)$ ist. Entsprechend wird Entscheidbarkeit für 1TM sowie deterministische Maschinen definiert. R 1TM - C (f) ist dann die Klasse aller durch 1TM in Regime $\leq f$ entscheidbaren Sprachen. Entsprechend kann man die Klassen für TM bzgl. V und H definieren.

Bei der Aufzählbarkeit ist die Situation insofern etwas anders, als wir uns hier bei der Kompliziertheitsschranke auf die Länge des Ergebniswortes beziehen.

Die TM M zählt L mit leerem Band und in durch f(n) beschränkter Kompliziertheit auf, falls w genau dann Element von L ist, wenn eine Berechnung von M existiert, die, mit leerem Band beginnend, in $\Phi_M(w) \leq f(|w|)$ in einen Finalzustand gelangt und gleichzeitig genau das Wort w erzeugt hat.

Die TM M zählt L in durch f(n) beschränkter Kompliziertheit auf, falls w genau dann Element von L ist, wenn ein Wort u existiert, von dem ausgehend M wie oben w erzeugt. Entsprechend wird die Aufzählbarkeit im Falle von 1 TM definiert.

G_λ TM - C(f) ist die Klasse aller durch TM mit leerem Band in Regime $\leq$ f aufzählbaren Sprachen, G TM - C(f) die Klasse aller durch TM in Regime $\leq$ f aufzählbaren Sprachen. Entsprechend werden die Klassen für 1TM sowie V und H definiert.

G TM - C(const) $=_{Df} \bigcup_{k=1}^{\infty}$ G TM - C(k) .

(entsprechend für R, 1 TM und V bzw. H).

G TM - C(const) sind die mit beschränktem Regime auf TM aufzählbaren Sprachen.

REG$=_{Df}$ Klasse der regulären Mengen, CF$=_{Df}$ Klasse der kontextfreien Mengen.

3. Aufzählungen und Entscheidungen mit beschränktem Regime auf 1 TM bzw. 1 DTM

Es ist gut bekannt, daß man auf einbändrige Turingmaschinen mit beschränktem Regime genau die regulären Sprachen aufzählen bzw. entscheiden kann (/5/):

R DTM - C(const) = R TM - C(const) =

G DTM - C(const) = G TM - C(const) = REG.

Wir untersuchen nun den Einfluß des Einweg-Eingabebandes auf Entscheidungen bzw. Aufzählungen mit beschränktem Regime:

Satz 1: a) G 1 DTM - C(const) = G 1DT - C(const) = REG

b) R 1 DTM - C(const) $\supsetneq$ REG.

Beweis: Zu b) Man überlegt sich leicht, daß REG $\subseteq$ R 1DTM - C(const) gilt. Weiterhin ist $\{a^n b^n : n \in \mathbb{N}\} \in$ R 1DTM - C (2), aber nicht regulär.

Zu a) (Beweisidee von K. Wagner): Wir zeigen, daß jede durch 1 DTM (bzw. 1TM) in beschränktem Regime aufzählbare Menge bereits durch DTM (bzw. TM) in der gleichen Regimeschranke aufzählbar ist.

Dazu nehmen wir an, L sei durch die 1 DTM M in Regime $\leq$ c aufzählbar. Wir konstruieren eine DTM M' mit folgenden Eigenschaften: Das Band von M' ist in c+1 Spuren unterteilt. Die ersten c Spuren sollen als Eingabespuren zur Simulation des Eingabeverhaltens der 1 DTM M und die letzte Spur zur Simulation des Arbeitsbandverhaltens von M dienen. Wird ausgehend

vom Wort x durch M ein Wort $y \in L$ erzeugt, so reicht es, ein Wort x' zu konstruieren, so daß M' auf x' ebenfalls y erzeugt. M betritt jede Zelle des Arbeitsbandes höchstens c-mal. Bei jedem Betreten einer Arbeitsbandzelle durch den Arbeitskopf K2 von M liest der Eingabekopf K1 von M gleichzeitig ein Eingabesymbol. Während K2 auf einer Zelle verweilt, kann K1 möglicherweise mehrere Eingabesymbole lesen; offenbar kann man sich aber auf den Fall beschränken, daß K1 bei unbewegtem K2 nur beschränkt viele ($\leq c'$) Eingabesymbole liest, denn anderenfalls kann man das Eingabewort x zu $\hat{x}$ verkürzen, wobei M auf $\hat{x}$ ebenfalls y liefert und $\hat{x}$ die Beschränktheitsbedingung hinsichtlich c' erfüllt. Verteilt man nun die beim Berechnungsprozeß von M auf x verarbeiteten Eingabesymbole gemäß dem Arbeitsbandverhalten von M auf die c Spuren bei M', so daß x' eine endliche Folge von (c+1)-Tupeln ist mit der Eigenschaft, daß M' in den ersten c Spuren einer Zelle gerade nacheinander die Eingabesymbole vorfindet, die in der entsprechenden Arbeitsbandzelle von M eingelesen werden, so liefert M' auf x' als Ergebnis in der (c+1)-ten Spur y, wenn man vereinbart, daß M' auf einer Zelle als Eingabesymbol das oberste nichtmarkierte Symbol wählt, in der (c+1)-ten Spur M' simuliert und beim Verlassen der Zelle das gerade "gelesene" Eingabesymbol markiert. Das Zustandsverhalten von M' entspricht ebenfalls dem von M, so daß M' in einen Finalzustand gelangt, wenn das bei M der Fall ist und umgekehrt. Alle durch M aufgezählten Wörter werden also auch durch M' aufgezählt. Wird umgekehrt durch M' ein Wort aufgezählt, so kann man aus dem Eingabewort von M' eines konstruieren, welches bei M zum gleichen Ergebnis führt, da M' lediglich M simuliert.
Die Simulation erhöht das Regime nicht, und die gleiche Konstruktion ist sowohl für deterministische als auch für nichtdeterministische Maschinen möglich. Weiterhin ist GTM - C(const) = REG, und da man offenbar auch jede reguläre Sprache in beschränktem Regime auf 1DTM erzeugen kann, folgt die Behauptung des Satzes.
Für größere Regimeschranken ändert sich jedoch diese Situation:

Satz 2: Für $f \geq id$ und beliebige $\varepsilon > 0$ gilt

$$R\ 1DTM - C(f) \subseteq G\ 1DTM - C((1+\varepsilon)f)$$

bzw. $R\ 1TM - C(f) \subseteq G\ 1TM - C((1+\varepsilon)f)$.

Beweis: Wir nehmen an, daß $L \in R\ 1DTM - C(f)$ und $L \neq \emptyset$ ist. Will man L aufzählen, so reicht es offenbar, für Eingabewörter x diese Wörter auf das Arbeitsband zu übernehmen (z.B. in eine erste Spur), und zwar werden jeweils $\left[\frac{1}{\varepsilon}\right]$ Einleseschritte gleichzeitig erledigt, indem der Eingabeabschnitt dieser Länge in der ersten Spur aufgeschrieben wird, und gleichzeitig wird der Entscheidungsprozeß von L in der zweiten Spur des Bandes ausgeführt. Wird x akzeptiert, so löscht die zu konstruierende Maschine M die zweite Spur, während die erste Spur am Ende x enthält. Wird x im deterministischen Fall abgelehnt, so löscht M alles und gibt ein festes x_0, $x_0 \in L$, aus. Für nichtdeterministische Maschinen erübrigt sich dieser Fall. Da man für deterministisches Regime nicht weiß, ob linearer speed-up möglich ist, ist es also in diesem Falle auch unbekannt, ob man auf die Vergrößerung durch ε verzichten kann. In gewissem Sinne ist die Schranke id in Satz 2 auch die bestmögliche Schranke, denn der nachfolgende Satz besagt, daß es Sprachen gibt, die zwar in R 1DTM - C(const) liegen, aber zu ihrer Erzeugung lineares Regime benötigen:

Satz 3: 1. $L_0 =_{Df} \{w \# w^R : w \in \{0,1\}^*\} \in R\ 1DTM - C\ (2)$

2. $Vf\ (f \prec id \longrightarrow L_0 \notin G\ 1TM - C(f))$

Beweis: Der Beweis erfolgt mit Hilfe einer Variante der Spurenmethode für Einweg-Turingmaschinen.

Wir nehmen an, die 1(D)TM ist in ihrer Arbeitsweise folgendermaßen normiert:

1. Das Ergebnis der Aufzählung befindet sich rechts vom Startpunkt des Arbeitsbandkopfes.
2. Die 1(D)TM beendet am rechten Ende des Ergebniswortes ihre Arbeit.

Ist x eine Zelle des Arbeitsbandes, so bezeichne $\varrho(x)$ die rechte Randspur von x, d.h. die Folge aller Zustände, in denen x nach rechts verlassen bzw. betreten wird, in ihrer zeitlichen Reihenfolge. Analog dazu sei $\lambda(x)$ die linke Rand-

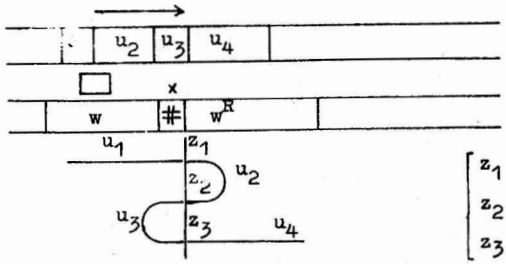
spur von x und zu einem Wort $w = x_1 \ldots x_n$ sei $\varrho(w) = \varrho(x_n)$ und $\lambda(w) = \lambda(x_1)$.
Ist M eine 1(D)TM und k die Zahl ihrer Zustände sowie f(n) die maximale Spurlänge auf Wörtern der Länge n, so existieren höchstens $k^{f(n)}$ verschiedene Spuren. Ist $f < id$, so existiert also ein n mit $k^{f(n)} < 2^n$.
Zum Beweis des Satzes benötigen wir noch das folgende

Lemma: Ist M eine 1(D)TM, die in durch f beschränktem Regime die Sprache L aufzählt, so existiert eine 1(D)TM M', die in jedem Takt ihren Eingabekopf bewegt und ebenfalls L in Regime $\leq f$ aufzählt.

Beweis des Lemmas: M' arbeitet folgendermaßen: bleibt der Eingabekopf K1 von M auf y stehen, so prüft M', ob der nächste Buchstabe des Eingabewortes y ist. Ist das der Fall, so führt M' denselben Befehl wie M aus; ist der nachfolgende Buchstabe von y verschieden, so geht M' in einen Zustand über, aus dem M' nie zu einem Finalzustand gelangt. Man sieht nun leicht, daß M' dieselbe Menge wie M aufzählt. Insbesondere hat M' das gleiche Arbeitsbandverhalten wie M, d.h. auch kein größeres Regime als M. □

Wir nehmen an, M sei eine 1(D)TM, die L_0 in Regime $f < id$ aufzählt. Dann gibt es Worte $w,w' \in \{0,1\}^*$ mit der Eigenschaft: $w \neq w'$ und $\varrho(w\#) = \lambda(w'^R)$, da 2^n Worte der Länge n über 0,1 existieren. O.B.d.A. kann vorausgesetzt werden, daß M die im Lemma formulierte Eigenschaft besitzt, daß K1 sich in jedem Takt bewegt.
Es sei u ein Eingabewort bei Erzeugung von $w \# w^R$, das folgende Faktorisierung besitzt: $u = u_1 \ldots u_{m+1}$, wobei u_1 das bis zum ersten Verlassen der Zelle x mit der Endbeschriftung # gelesene Eingabestück, u_2 das nach dem ersten Verlassen von x bis zur ersten Rückkehr zu x gelesene Eingabestück usw. sind:

Bsp.



Weiterhin sei v ein Eingabewort mit der Faktorisierung $v = v_1 \dots v_{m+1}$ analog zur Faktorisierung von u (da ja $\lambda(w'^R) = \varrho(w \#)$ sein soll).

Bildet man nun das Eingabewort $t = u_1\, v_2\, u_3\, v_4 \dots v_{m+1}$, so erzeugt L auf t die Ausgabe $w \# w'^R$:

L beginnt mit dem Lesen von u_1 und arbeitet links von x wie bei Eingabe von u bis zum ersten Verlassen von x.

Durch die Spurengleichheit arbeitet L danach auf v_2 wie bei v usw. und erzeugt daher $w \# w'^R$, das nicht in L_o liegt.

Also kann L die Sprache L_o nicht im Regime $f \prec id$ erzeugen.

Man überlegt sich leicht, daß für beliebige f die Inklusionen

(1) R DTM - C(f) $\subseteq$ G DTM - C (f+2) und

(2) G DTL - C(f) $\subseteq$ G 1DTM - C (f+1) gelten.

(analog für den nichtdeterministischen Fall).

Man kann den Satz 3, Punkt 2 also auch für die Klassen R(D)TM - C(f) und G(D)TM - C(f), $f \prec id$, beweisen.

Für den Fall $f \succeq id$ lassen sich die Beziehungen der gesamten Klassen noch präzisieren:

Es gilt offenbar:

R DTM - C(f) $\subseteq$ R 1DTM - C(f+1) und für beliebiges $\varepsilon > 0$

R 1DTM - C(f) $\subseteq$ R DTM - C((1+ε)f)

(analog für den nichtdeterministischen Fall) sowie

G 1DTL - C(f) $\subseteq$ G DTL - C((1+ε)f).

Für $f \succeq id$ ist es also bis auf konstanten Faktor der Schrankenfunktion ohne Einfluß, ob man mit oder ohne Einweg-Eingabeband aufzählt (bzw. entscheidet).

Die Frage, ob die Inklusion bei Satz 2 echt ist, bleibt allerdings offen, ebenso hinsichtlich der Inklusionen (1) und (2).

4. Aufzählungen und Entscheidungen mit beschränkter Rückkehr bzw. beschränkter dualer Rückkehr auf 1 TM bzw. 1 DTM

In /7/ zeigte Wechsung, daß RTM-V(const)=CF ist.
In /2/ wies Chytil nach, daß auch G_λTM-V(const)=CF gilt.
Aufbauend auf diesen Resultaten untersuchen wir für beschränktes V und H den Einfluß eines Einweg-Eingabebandes.
Weiterhin werden Verallgemeinerungen der obigen Resultate angegeben. Der nachfolgende Satz baut auf Satz 1 auf:

Satz 4: a) G 1TM - V(const) $\subseteq$ GTM - V(const)
b) G 1TM - H(const) $\subseteq$ GTM - H(const)

Beweis: a) Die zusätzliche Schwierigkeit gegenüber dem Beweis von Satz 1 ist hier die, daß nicht die Länge der Gesamtspuren, sondern nur die Länge der Restspuren auf den Arbeitsbandzellen durch ein c beschränkt werden kann. Man kann also durch Unterteilung des Bandes der TM in Spuren im wesentlichen nur die Eingabebuchstaben einer Zelle nach ihrer ersten Änderung erfassen.
Es sei u ein Eingabewort, auf dem die 1TM M in Rückkehr $\leq$ c das Wort w erzeugt. Um w mit einer TM M' in Rückkehr $\leq$ c zu erzeugen, wählen wir folgendes Eingabewort u' :
Das Band von M' ist in c+1 Spuren unterteilt.
u' umfaßt die gesamte bei der Erzeugung von w durch M aktive Bandzone. Auf jeder Zelle, die bei dieser Erzeugung jemals geändert wurde, stehen in den ersten c Spuren die Eingabesymbole dieser Zelle nach der ersten Änderung in ihrer zeitlichen Reihenfolge. Die (c+1)-te Spur dient wieder als Arbeitsspur zur Simulierung des Arbeitsbandverhaltens von M. M' arbeitet folgendermaßen:
Es beginnt im Anfangszustand z_0 von M, wählt nichtdeterministisch einen hypothetischen Eingabebuchstaben und führt danach die M entsprechende Zustandsüberführung und Laufrichtung aus. Befindet sich M' infolge dieser Überführung auf einer Zelle, in der die ersten c Spuren nicht leer sind, so kann es sich nichtdeterministisch dafür entscheiden, diese Zelle zu aktivieren, falls der Eingabebuchstabe der ersten Spur der aktuelle

Eingabebuchstabe ist (d.h. falls der im letzten Takt vorhandene Eingabebuchstabe mit dem der ersten Spur übereinstimmt oder K1 von M bewegt wurde).
M' verhält sich in diesem Falle auf der (c+1)-ten Spur wie M auf dem Arbeitsband. Danach erfolgt eine Markierung des Symbols der ersten Spur, damit das Symbol der zweiten Spur als nächstes Eingabesymbol dieser Zelle fungiert. Gelangt M' danach auf Zellen, deren erste c Spuren leer sind bzw. die noch nicht aktiviert wurden, so erfolgt wieder die nichtdeterministische Wahl von Eingabesymbolen usw..
Es ist klar, daß auf diese Weise M' genau dieselbe Sprache aufzählt wie M, und zwar in der gleichen Rückkehrschranke.
b) Im Falle von H verfahren wir analog zu a).
Die ersten c Spuren jeder Zelle erhalten die Eingabesymbole bis zum c-ten Besuch dieser Zelle durch M, und bei jedem Besuch durch M' wird das gerade gelesene Eingabesymbol (der obersten nichtmarkierten Spur) markiert. Sind alle c ersten Spuren einer Zelle markiert, so rät M' nichtdeterministisch Eingabesymbole usw..
Liefert auch für beschränktes H und V auf 1 TM die Entscheidbarkeit mehr als die Aufzählbarkeit?
$L_1 =_{Df} \{w \# w^R \# w : w \in \{0,1\}^*\}$ ist nicht kontextfrei und $L_1 \in R$ 1DTM - V (3) sowie $L_1 \in R$ 1DTM - H(1), (es gilt sogar $L_1 \in R$ 1DTM - C(3)).

Satz 5: G 1TM - V(const) = GTM - V(const) = GTM - H(const) = $= G_\lambda$ TM - H(const) = G1TM - H(const) = CF

Beweis: Wir zeigen die folgenden Inklusionen:
1.) G1TM - V(const) $\subseteq$ GTM - V(const) gilt nach Satz 4.
2.) GTM - V(const) $\subseteq$ GTM - H(const) :
Wir nutzen dazu eine in /8/ verwendete Beweismethode aus, indem man bei der Simulation den Berechnungsweg umkehrt. Ist M eine TM, die L mit beschränktem V aufzählt, so ist die Länge der Restspuren durch ein c beschränkt, und zu $w \in L$ existiert ein u, auf dem M in normierter Startsituation im Anfangszustand beginnt und als Ergebnis w liefert. M' beginnt seine Arbeit mit der nichtdeterministischen Erzeugung eines w, auf dem es

beginnt, mögliche Berechnungen von M umzukehren, indem es in einem Finalzustand startet, und gibt w aus, falls es bei diesem Prozeß auf einem u in die Startsituation, d.h. insbesondere den Anfangszustand von M gelangt.

3.) GTM - H(const) $\subseteq$ R TM - H(const) :

Für $L \in$ GTM - H(const) ist eine Entscheidung auf einer TM L' in beschränktem H zu konstruieren. Dazu unterteilen wir das Arbeitsband von M' in zwei Spuren, und das Eingabewort w kommt auf die obere Spur. Auf der unteren Spur wird nichtdeterministisch ein Wort v erzeugt, auf dem die L aufzählende Maschine M die Abarbeitung von v beginnt.

Kommt M in einen Finalzustand, so prüft M mit zwei Durchläufen, ob das über dem Ergebnis stehende Wort w identisch mit dem Ergebnis ist. Wenn ja, so wird w akzeptiert.

4.) RTM - H(const) = CF ist nach Hibbard (/3/)) bekannt, ebenso CF = G_λ TM - V(const) (vgl. Chytil /2/).

5.) G_λ TM - V(const) $\subseteq$ G 1TM - V(const) ist klar, da G_λ Aufzählung mit anfangs leerem Band bedeutet.

6.) RTM - H(const) $\subseteq$ G_λ TM - H(const) :

Man erzeugt zunächst nichtdeterministisch ein w und prüft dann mit dem Entscheidungsverfahren, ob w akzeptiert wird. Wenn ja, so wird w aufgezählt.

7.) G_λ TM - H(const) $\subseteq$ G 1TM - H(const) ist ebenfalls klar (wie bei 5.).

8.) G1TM - H(const) $\subseteq$ GTM - H(const) gilt nach Satz 4.

Aus Satz 5 folgt insbesondere:

$L_1 \notin$ G1TM - V(const) und $L_1 \notin$ G 1TM - H(const).

Vermutlich gilt sogar $L_1 \notin$ G1TM - V(f) für $f \prec id$.

Leicht einzusehen ist jedoch, daß für $f \geq id$ ein Analogon des Satzes 2 gilt:

$$R\ 1TM - V(f) \subseteq G\ 1TM - V((1+\varepsilon)f) \quad \text{und}$$
$$R\ 1TM - H(f) \subseteq G\ 1TM - H((1+\varepsilon)f) .$$

Ich danke Herrn Dr.sc. Klaus Wagner für einen Hinweis zum Beweis von Satz 1.

Literatur

/1/ Brandstädt, A. und Saalfeld, D., Eine Hierarchie beschränkter Rückkehrberechnungen auf on-line Turingmaschinen, EIK 13 (1977), 571-583.

/2/ Chytil, M. P., Return Complexity of Language Generation, II. Sympos. "Algorithmische Komplizierth., Lern- u. Erk. Proz." Jena, 1976

/3/ Hibbard, T. N., A Generalization of Context-Free Determinism, Inf. and Control 11 (1967), 196-238.

/4/ Peckel, J., On a Deterministic Subclass of Context-Free Languages, MFCS 1977 Lecture Notes in Comp Science 53 (1977), 430-434.

/5/ Трахтенброт, Б.А., Сложность алгоритмов и вычислений, Новосибирск, 1967.

/6/ Wechsung, G., Characterization of Some Classes of Context-Free Languages in Terms of Complexity Classes, MFCS 1975 Lecture Notes in Computer Science 32, 457-461.

/7/ Wechsung, G., Kompliziertheitstheoretische Charakterisierung der kontextfreien und linearen Sprachen, EIK 12 (1976) 6, 289-300.

/8/ Wechsung, G., Brandstädt, A., A Relation between Space, Return and Dual Return Complexities, erscheint 1979 in Theoretical Computer Science.

eingegangen: 15.9.1978

Anschrift des Verfassers:

Dr. Andreas Brandstädt
Friedrich-Schiller-Universität Jena
Sektion Mathematik
DDR - 69 Jena
Schillerstraße UHH

Rostock. Math. Kolloq. 10, 23 - 36 (1978)

Klaus Denecke

Schwache Automorphismen präprimaler Algebren, die arithmetische Varietäten erzeugen

0. In /1/ wird eine Klassifizierung aller präprimalen Algebren vorgenommen und als eine ihrer Eigenschaften bewiesen, daß es nur eine Klasse präprimaler Algebren mit nichtidentischen Automorphismen gibt, das sind die Algebren der Form $S^k_{s(x)}$.
Die Automorphismengruppe einer solchen präprimalen Algebra ist zyklisch und von Primzahlordnung. Hierbei werden präprimale Algebren nur bis auf Äquivalenz, also unabhängig von einer konkreten Signatur untersucht.
Dieses Vorgehen legte es nahe, den Begriff des schwachen Automorphismus zu verwenden, der von A. Goetz /6/ und E. Marczewski /12/ eingeführt wurde, und schwache Automorphismen präprimaler Algebren zu untersuchen.
In /3/ wird gezeigt, daß eine präprimale Algebra entweder konstantiv und einfach ist, d.h. jedes Element der Trägermenge ist nullstellige Operation der Algebra, oder eine arithmetische Varietät erzeugt, d.h. die Kongruenzenverbände aller Algebren der Varietät sind distributiv und die Kongruenzen sind paarweise vertauschbar. (/13/)
In dieser Arbeit werden die schwachen Automorphismen der präprimalen Algebren charakterisiert, die arithmetische Varietäten erzeugen und bewiesen, daß die schwachen Automorphismen dieser Algebren gerade die unären eineindeutigen algebraischen Operationen dieser Algebren sind. Für beliebige präprimale Algebren wird gezeigt, daß die Automorphismengruppe Normalteiler in der Gruppe der eineindeutigen unären Operationen ist. Die Gruppe der schwachen Automorphismen einer präprimalen Algebra ist isomorph zur Gruppe der Klonautomorphismen.

1. $\underline{A} = \langle E_k, F \rangle$ mit $E_k = \{0,1,\ldots,k-1\}$ $k \geq 1$ sei eine endliche Algebra, $F \subseteq \underline{F}(E_k)$, wobei $\underline{F}(E_k)$ die Klasse aller Funktionen ist, die auf E_k definiert sind, d.h. $\underline{F}(E_k) = \bigcup_{n=0}^{\infty} F_n(E_k)$ mit $F_n(E_k) = \{f: E_k^n \longrightarrow E_k\}$.

$\overline{F}$ bezeichne die aus F durch Superposition von Funktionen entstehende Klasse von Funktionen aus $\underline{F}(E_k)$, die alle Projektionen $e_i^n \in F_n(E_k)$ mit $e_i^n(x_1,\ldots,x_n) = x_i$ (i=1,...,n) enthalten soll. Dann ist $\overline{F}$ Polynomialklasse oder Operationenklon von $\underline{A}$.

Zwei Algebren heißen äquivalent, wenn sie den gleichen Operationenklon erzeugen.

Aut($\underline{A}$) sei die Automorphismengruppe von $\underline{A}$.

Zu jeder Algebra gehört eine Funktionenalgebra, deren Trägermenge $\overline{F}$ ist und deren Operation die Komposition von Funktionen ist. Ein Automorphismus dieser zu $\underline{A}$ gehörigen Funktionenalgebra heißt Klonautomorphismus von $\underline{A}$. Aut($\overline{F}$) sei die Gruppe der Klonautomorphismen von $\underline{A}$.

Eine Permutation τ der Trägermenge E_k der Algebra $\underline{A} = \langle E_k, F \rangle$ heißt ein schwacher Automorphismus, wenn die Abbildung

(1) $\alpha : f \longrightarrow f^{\alpha} = \tau \cdot f \cdot \tau^{-1}$

eine eineindeutige Abbildung der Menge $\overline{F}$ (aller algebraischen Operationen von $\underline{A}$) auf $\overline{F}$ ist, wobei $\tau \cdot f \cdot \tau^{-1}$ definiert ist durch

(2) $\tau \circ f \circ \tau^{-1}(x_1,\ldots,x_n) = \tau(f(\tau^{-1}(x_1),\ldots,\tau^{-1}(x_n)))$.

Es ist klar, daß jeder Automorphismus von $\underline{A}$ auch ein schwacher Automorphismus von $\underline{A}$ ist.

Aut*($\underline{A}$) sei die Gruppe der schwachen Automorphismen von $\underline{A}$.

Die durch den schwachen Automorphismus τ induzierte Abbildung $\alpha : f \longrightarrow f^{\alpha} = \tau \cdot f \cdot \tau^{-1}$ ist ein Klonautomorphismus von $\underline{A}$.

Die durch (1) definierten Klonautomorphismen heißen nach /9/ innere Automorphismen der zu $\underline{A}$ gehörigen Funktionenalgebra.

$Q(\underline{A})$ sei die Menge der unären eineindeutigen algebraischen Operationen der Algebra $\underline{A}$. Da $\underline{A}$ endlich ist, bildet $Q(\underline{A})$ bzgl. der Nacheinanderausführung der Operationen ebenfalls eine Gruppe.

Lemma 1.1: /14/ Jede unäre eineindeutige algebraische Operation

der endlichen Algebra $\underline{A} = \langle E_k, F \rangle$ ist ein schwacher Automorphismus von $\underline{A}$: $Q(\underline{A}) \subseteq Aut^*(\underline{A})$.

Lemma 1.2: /4/ $Q(\underline{A})$ ist Normalteiler von $Aut^*(\underline{A})$.

Die Abbildung, die jedem schwachen Automorphismus τ von $\underline{A}$ den zugehörigen inneren Klonautomorphismus α zuordnet: $\tau \longrightarrow \alpha$, ist ein Gruppenhomomorphismus von $Aut^*(\underline{A})$ in $Aut(\overline{F})$ mit dem Kern $Aut(\underline{A})$, d.h. die Automorphismen von $\underline{A}$ sind gerade die schwachen Automorphismen, die auf das Einselement von $Aut(\overline{F})$ abgebildet werden.

Theorem 1.3: /14/ Die Abbildung $\tau \longrightarrow \alpha$ ist ein Homomorphismus von $Aut^*(\underline{A})$ in $Aut(\overline{F})$ mit dem Kern $Aut(\underline{A})$.

Corollar 1.4: $Aut(\underline{A})$ ist ein Normalteiler von $Aut^*(\underline{A})$.

Corollar 1.5: $Aut^*(\underline{A})/Aut(\underline{A})$ ist isomorph zu einer Untergruppe von $Aut(\overline{F})$.

2. Definition 2.1: Eine Algebra $\underline{A} = \langle E_k, F \rangle$ $(k \geq 1)$ ist genau dann präprimal, wenn $\overline{F}$ eine maximale Klasse in $\underline{P}(E_k)$ ist, d.h. $\overline{F} \subset \underline{P}(E_k)$ und für jede Funktion $f \notin \overline{F}$, $f \in \underline{P}(E_k)$ gilt: $\{\overline{F \cup f}\} = \underline{P}(E_k)$.

In /1/ wird, ausgehend von der Rosenbergschen Klassifizierung aller maximalen Klassen in $\underline{P}(E_k)$, eine vollständige Klassifizierung aller präprimalen Algebren (bis auf Äquivalenz) vorgenommen. Ein Ergebnis dieser Klassifizierung ist (/3/):

Theorem 2.1: Eine präprimale Algebra ist entweder konstantiv und einfach (jedes Element von E_k ist nullstellige Operation von $\underline{A}$) oder erzeugt eine arithmetische Varietät (alle Algebren der Varietät $V_{\underline{A}} = HSP\underline{A}$ haben distributive Kongruenzenverbände und die Kongruenzen sind paarweise vertauschbar).

Die Algebren der zweiten Art gehören folgenden Klassen an:
$\underline{S^k_{s(x)}} = \langle E_k, F_d \rangle$, $\overline{F}_d = S^k_{s(x)}$ ist die Klasse aller Funktionen, die bzgl. einer Permutation s von E_k mit Zyklen gleicher Primzahllänge ohne invariante Elemente selbstdual sind: für alle $f \in S^k_{s(x)}$ gilt: $f(s(x), s(y), \ldots) = s(f(x,y,\ldots))$.

$\underline{T^k_{A,o}} = \langle E_k, F_s \rangle$ $\quad F_s = T^k_{A,o}$ ist die Klasse aller Funktionen, die eine Teilmenge $\emptyset \neq A \subset E_k$ bewahren.

$\underline{U^k_\Theta} = \langle E_k, F_h \rangle$ $\quad F_h = U^k_\Theta$ ist die Klasse aller Funktionen, die eine Äquivalenzrelation Θ auf E_k bewahren. Θ ist dabei verschieden von der Null- und der Einheitsrelation.

Ein Ergebnis aus /2/ ist:

Theorem 2.2: Jede nichttriviale Automorphismengruppe einer präprimalen Algebra ist zyklisch und von Primzahlordnung. Dabei sind die präprimalen Algebren der Form $S^k_{s(x)}$ die einzigen mit nichttrivialen Automorphismengruppen, ihre Automorphismengruppen sind zyklisch und von Primzahlordnung.

3. Wir untersuchen schwache Automorphismen und Klonautomorphismen von Algebren der Form $S^k_{s(x)}$. In /2/ wird bewiesen, daß $\mathrm{Aut}(S^k_{s(x)}) = \langle s \rangle$ die von s erzeugte zyklische Gruppe von Primzahlordnung ist. $Q(S^k_{s(x)})$ ist die Klasse der eineindeutigen unären Operationen von $S^k_{s(x)}$.
$\mathrm{Aut}^*(S^k_{s(x)})$ ist die Gruppe der schwachen Automorphismen, $\mathrm{Aut}(S^k_{s(x)})$ die Gruppe der Klonautomorphismen, S_k die symmetrische Gruppe der Elemente $E_k = \{0,1,\ldots,k-1\}$.
Nach Corollar 1.4. ist $\mathrm{Aut}(S^k_{s(x)})$ Normalteiler in $\mathrm{Aut}^*(S^k_{s(x)})$. Ferner ist nach Lemma 1.2. $Q(S^k_{s(x)})$ Normalteiler in $\mathrm{Aut}^*(S^k_{s(x)})$. $Q(S^k_{s(x)})$ enthält genau die Elemente von S_k, die mit s vertauschbar sind, daher ist $Q(S^k_{s(x)})$ der Normalisator von s in S_k. Es ist klar, daß $Q(S^k_{s(x)})$ auch der Normalisator von $\langle s \rangle = \mathrm{Aut}(S^k_{s(x)})$ in S_k ist. Der Normalisator von $\mathrm{Aut}(S^k_{s(x)})$ ist aber die maximale Untergruppe von S_k, in der $\mathrm{Aut}(S^k_{s(x)})$ Normalteiler ist.

Nach Lemma 1.1 gilt: $Q(S^k_{s(x)}) \subseteq Aut^*(S_{s(x)})$.

Wegem $Aut(S^k_{s(x)}) \subseteq Q(S^k_{s(x)}) \subseteq Aut^*(S_{s(x)}) \subseteq S_k$ folgt daher $Q(S^k_{s(x)}) = Aut^*(S^k_{s(x)})$ oder $Aut^*(S^k_{s(x)}) = S_k = Q(S^k_{s(x)})$, da im zweiten Fall $Aut(S^k_{s(x)})$ Normalteiler in S_k ist. Damit erhalten wir:

Lemma 3.1: Für jede präprimale Algebra der Form $S^k_{s(x)} = \langle E_k, F_d^2 \rangle$ gilt: $Q(S^k_{s(x)}) = Aut^*(S^k_{s(x)})$, d.h. jeder schwache Automorphismus ist eine eineindeutige unäre algebraische Operation.

Corollar 3.2: Die Gruppe der schwachen Automorphismen von $S^k_{s(x)}$ ist der Normalisator von s in S_k.

Es muß noch geklärt werden, in welchem Fall $Q(S^k_{s(x)}) = S_k$ ist. Offenbar stimmt der Normalisator von $\langle s \rangle$ genau dann mit der ganzen Gruppe überein, wenn $\langle s \rangle$ Normalteiler in S_k ist. S_k hat für $k \geq 5$ genau die Normalteiler S_k, A_k, E_k, wo A_k die alternierende Gruppe ist. Ein Erzeugendensystem für die alternierende Gruppe sind die dreigliedrigen Zyklen $(01r) : A_k = \langle (012)\ (013)\ (014)\ \ldots\ (01k-1) \rangle$. Da zwei dieser Zyklen nicht kommutativ multiplizierbar sind, kann A_k für $k > 3$ nicht zyklisch sein, d.h. für $k \geq 5$ kann nicht $Aut^*(S^k_{s(x)}) = S_k$ gelten.

Für k = 2 gilt: s=(01): $Aut(S^2_{s(x)}) = \langle (01) \rangle$, $Q(S^2_{s(x)}) = \langle (01) \rangle$, also $Aut(S^2_{s(x)}) = Q(S^2_{s(x)}) = Aut^*(S^2_{s(x)}) = S_2$.

Für k=3 kommt als s nur in Frage: s=(012), dann ist :

$Aut(S^3_{s(x)}) = \langle (012) \rangle = A_3$, $Q(S^k_{s(x)}) = Aut^*(S^3_{s(x)}) = S_3$,

da $Aut(S^3_{s(x)}) = A_3$ Normalteiler in S_3 ist.

Für k=4 kommen folgende s in Frage:

$s_1 = (01)(23)$, $s_2 = (02)(13)$, $s_3 = (03)(12)$.

In jedem der drei Fälle ist $Aut(S^4_{s(x)})$ isomorph zur zyklischen Gruppe zweiter Ordnung, die aber nicht Normalteiler in S_4 ist. Folglich gilt für k=4: $Aut(S^k_{s(x)}) \subset S_4$.

Damit haben wir folgendes Ergebnis:

Lemma 3.3: Nur für k=2 und k=3 gilt $Aut^*(S^k_{s(x)}) = S_k$.

D. Lau hat in /8/ bewiesen, daß jeder Klonautomorphismus von $S^k_{s(x)}$ ein innerer Klonautomorphismus ist. Beim Beweis wird analog vorgegangen wie beim Beweis entsprechender Aussagen in /11/.
Es folgt:

Lemma 3.4: Für jede Algebra $S^k_{s(x)}$ gilt: $Aut^*(S^k_{s(x)})$ ist isomorph zu $Aut(S^k_{s(x)})$, d.h. die Gruppe der schwachen Automorphismen ist isomorph zur Gruppe der Klonautomorphismen.

Aus Corollar 3.2. und Theorem 2.2. ergibt sich folgende Verallgemeinerung für beliebige präprimale Algebren:

Theorem 3.5: Sei $\underline{A}$ eine beliebige präprimale Algebra, dann ist $Aut(\underline{A})$ ein Normalteiler von $Q(\underline{A})$.

Für präprimale Algebren der Form $S^k_{s(x)}$ folgt das Theorem aus Corollar 3.2. Für andere präprimale Algebren aus der Tatsache, daß maximale Klassen in $\underline{F}(E_k)$ alle Projektionen und damit auch die identische Funktion enthalten, daher ist der identische Automorphismus ein schwacher Automorphismus aus $Q(\underline{A})$.

5. Wir untersuchen schwache Automorphismen und Klonautomorphismen von präprimalen Algebren der Form $T^k_{A,o} = \langle E_k, F_s \rangle$ mit $F_s = T^k_{A,o}$.
Algebren dieser Form sind semiprimal (/1/), d.h. alle Funktionen $f \in \underline{F}(E_k)$, die alle Unteralgebren der Algebra bewahren, sind Polynomiale der Algebra.
Daher sind der dreistellige Diskriminator t mit

$t(x,y,z) = \begin{cases} x, & \text{wenn } x \neq y \\ z, & \text{wenn } x = y \end{cases}$ und der duale Diskriminator d mit

$d(x,y,z) = \begin{cases} x, & \text{wenn } x = y \\ z, & \text{wenn } x \neq y \end{cases}$ Polynomiale von $T^k_{A,o}$, d.h. Funktionen aus $T^k_{A,o}$.

Es gilt: $d(x,y,z) = t(x,t(x,y,z),z)$.

Lemma 4.1: Für jede präprimale Algebra der Form $\underline{T^k_{A,o}} = \langle E_k, P_s \rangle$ gilt: $Q(\underline{T^k_{A,o}}) = Aut^*(\underline{T^k_{A,o}})$, d.h. jeder schwache Automorphismus ist eine eineindeutige unäre algebraische Operation.

Beweis: Es ist $Q(\underline{T_{A,o}}) \leq Aut^*(\underline{T^k_{A,o}})$. Es muß bewiesen werden, daß nur die eineindeutigen unären Operationen aus $Q(\underline{T^k_{A,o}})$ schwache Automorphismen von $\underline{T^k_{A,o}}$ sind.

Sei τ eine Permutation von E_k, die A nicht bewahrt, dann gibt es ein $a \in A$ und ein $c \notin A$ mit $\tau(a) = c$. Sei $\tau^{-1}(a)=b$, wobei b irgendein Element $b \neq a$ aus E_k ist, wir wählen f mit $f(x)\equiv a$, diese konstante Operation ist Element von $\underline{T^k_{A,o}}$. Dann gilt: $\tau(f(\tau^{-1}(a))) = \tau(a) = c \notin A$. Also ist $\tau \cdot f \cdot \tau^{-1}$ für keine Permutation τ von E_k, die A nicht bewahrt, eine Operation von $\underline{T^k_{A,o}}$.

Nach Corollar 1.5. gilt: $Aut^*(\underline{T^k_{A,o}})/_{Aut(\underline{T^k_{A,o}})}$ ist isomorph zu einer Untergruppe von $Aut(\underline{T^k_{A,o}})$.

Wir wollen untersuchen, welche Klonautomorphismen innere Klonautomorphismen sind.

Dazu untersuchen wir zunächst, wie die konstanten Operationen und die Operationen aus $Q(\underline{T^k_{A,o}})$ bei einem beliebigen Klonautomorphismus α abgebildet werden.

Sei α ein Automorphismus von $Aut(\underline{T^k_{A,o}})$. Dann ist α ein Automorphismus der einstelligen Operationen von $\underline{T^k_{A,o}}$, da bei einem Automorphismus die Stellenzahl nicht größer werden kann, ebenso ein Automorphismus der nullstelligen Operationen, d.h. der Konstanten aus E_k. Jedem α ist also eindeutig eine Permutation der Elemente von A zugeordnet: $a \longrightarrow a^\alpha$. Weiter gilt bei α: für $e(x)=x$ ist $e^\alpha(x) = e(x) = x$.

Sei r eine eineindeutige unäre Operation aus $Q(\underline{T^k_{A,o}})$, dann ist $r(r^{-1}(x)) = e(x)$ und $r^\alpha((r^{-1})^\alpha(x)) = e^\alpha(x) = e(x)$.

Es folgt: $(r^{-1})^\alpha(x) = (r^\alpha)^{-1}(x)$, d.h., die Operationen aus $Q(\underline{T^k_{A,o}})$ gehen bei α wieder in Operationen aus $Q(\underline{T^k_{A,o}})$ über.

Jede Permutation aus $Q(\underline{T^k_{A,o}})$ läßt sich eindeutig in ein Produkt elementfremder Zyklen zerlegen: $r=z_1 . z_2 \ldots . z_1$.

Dabei kommen nach oben Gesagtem in einem beliebigen Zyklus entweder nur Elemente aus A oder nur Elemente aus $E_k \setminus A$ vor. Bei Anwendung von α gilt: $r^\alpha = z_1^\alpha . z_2^\alpha ..... z_1^\alpha$. Da bei einem Automorphismus aus $Q(\underline{T^k_{A,o}})$ die Ordnung der Elemente erhalten bleibt, folgt, daß r die gleiche Anzahl von Fixpunkten hat wie r^α.
Als nächstes untersuchen wir die Bilder von d und t bei α.

Lemma 4.2: Für jeden Klonautomorphismus α von $\underline{T^k_{A,o}}$ gilt: $d = d^\alpha$, $t = t^\alpha$.

Beweis: Jede Identität von $\underline{T^k_{A,o}}$ hat die Form f=g mit f,g $\in T^k_{A,o}$. Daraus folgt $f^\alpha = g^\alpha$, da α ein Automorphismus ist.
Nach /5/ sind d und t in einer Varietät durch folgende Identitäten bestimmt:
t ist genau dann Diskriminator einer Algebra $\underline{A}$, wenn gilt:
a. $t(x,z,z)=x \quad t(x,y,x)=x \quad t(x,x,z)=z$
b. $t(x,t(x,y,z),y)=y$
c. für jedes Operationssymbol f von $V_{\underline{A}}$ ist $t(x,y,f(z_1,\ldots,z_k)) = t(x,y,f(t(x,y,z_1),\ldots,t(x,y,z_k)))$, wo f k-stellig ist.
d ist genau dann dualer Diskriminator von $\underline{A}$, wenn gilt:
a. $d(x,z,z)=z \quad d(x,y,x)=x \quad d(x,x,z)=x$
b. $d(x,y,d(x,y,z))=d(x,y,z)$
c. $d(z,d(x,y,z),d(x,y,w))=d(x,y,z)$
d. für jedes Operationssymbol f von $V_{\underline{A}}$ ist $d(x,y,f(z_1,\ldots,z_k)) = d(x,y,f(d(x,y,z_1),\ldots,d(x,y,z_k)))$, wo f k-stellig ist.
Die entsprechenden Identitäten gelten auch für d^α und t^α, z.B. $t^\alpha(e^\alpha(x),e^\alpha(x), e^\alpha(z)) = e^\alpha(z)$, d.h. $t^\alpha(x,x,z) = z$.
Daher sind d^α (bzw. t^α) ebenfalls Diskriminator (bzw. dualer Diskriminator) von $\underline{T^k_{A,o}}$. Da Diskriminator und dualer Diskriminator von $\underline{T^k_{A,o}}$ eindeutig bestimmt sind, gilt für jeden Klonautomorphismus α von $T^k_{A,o}$: $d^\alpha = d$, $t^\alpha = t$.
Wir beweisen jetzt das

Lemma 4.3: Für jede Algebra $\underline{T^k_{A,o}}$ gilt: $Aut^*(\underline{T^k_{A,o}})$ ist isomorph zu $Aut(T^k_{A,o})$, d.h., die Gruppe der schwachen Automorphismen ist isomorph zur Gruppe der Klonautomorphismen.

Beweis: Ist card A=1, so gilt für alle $f \in T^k_{A,o}$: $f(a,\ldots,a)=a$. In /10/ wird bewiesen: Wenn T_a die abgeschlossene Klasse aller f ist mit der Eigenschaft: wenn $x_1=a \vee x_2=a \vee \ldots \vee x_n=a$, so folgt $f(x_1,\ldots,x_n)=a$ oder f ist konstant gleich a, so sind alle Automorphismen einer abgeschlossenen Klasse, die T_a enthält, innere Automorphismen. Die inneren Automorphismen des Klons entsprechen eineindeutig den schwachen Automorphismen der Algebra. Da für die Funktionen aus T_a ebenfalls gilt: $f(a,\ldots,a) = a$, so ist $T_a \subset T^k_{A,o}$ mit $A=\{a\}$. Damit gilt das Lemma 4.3. für card A = 1.

Sei card A ≠ 1 und $\text{card}(E_k \setminus A) \neq 2$. Dann gibt es für jedes c eine Operation f aus $Q(T^k_{A,o})$ mit genau einem Fixpunkt $c \notin A$: $f(c)=c$.

Nach dem vor Lemma 4.2. Gesagten hat dann auch f^α genau einen Fixpunkt $d \notin A$, $f^\alpha(d) = d$ und f^α ist eine eineindeutige unäre Operation aus $Q(T^k_{A,o})$.

Wir betrachten $d(x,f(x),a) = \begin{cases} x, & \text{wenn } f(x)=x, \text{ d.h. } x=c, c \notin A \\ a, & \text{wenn } f(x)\neq x, \text{ d.h. } x=c \end{cases} = u_{a_c}(x)$.

Für jedes feste $a \in A$ kann jedem $c \notin A$ umkehrbar eindeutig ein $d(x,f(x),a)$ zugeordnet werden, wobei $f \in Q(T^k_{A,o})$ mit genau einem Fixpunkt $c \notin A$ ist.

Bei Anwendung eines Klonautomorphismus α gilt nach Lemma 4.2.: $d^\alpha(e^\alpha(x),f^\alpha(x),a^\alpha) = d(x,f^\alpha(x),a^\alpha) = \begin{cases} x, & \text{wenn } f^\alpha(x)=x, \text{d.h. } x=d \\ a, & \text{wenn } f^\alpha(x)\neq x, \text{d.h. } x\neq d \end{cases}$

$d \notin A$, $a^\alpha \in A$. Es ist also $u^\alpha_{a_c}(x) = u_{a^\alpha_{c^\alpha}}(x)$.

Die Zuordnung $u_{a_c} \leftrightarrow u_{a^\alpha_{c^\alpha}} = u^\alpha_{a_c}$ ist eineindeutig. Da zu jedem u_{a_c} umkehrbar eindeutig ein (a,c), $a \in A$, $c \notin A$ gehört, entspricht der Abbildung $u_{a_c} \leftrightarrow u^\alpha_{a_c}$ eindeutig die Abbildung $(a,c) \leftrightarrow (a^\alpha,c^\alpha)$, $a \in A$, $c \notin A$. Jedem Klonautomorphismus α ist eindeutig eine Abbildung $(a,c) \leftrightarrow (a^\alpha,c^\alpha)$, $a \in A$, $c \notin A$ zugeordnet, wobei $a \leftrightarrow a^\alpha$ eine Permutation von A, $o \leftrightarrow o^\alpha$ eine Permutation von $E_k \setminus A$ ist. Diese Abbildungen sind aber gerade die Permutationen aus $Q(T^k_{A,o})$.

Umgekehrt ist jeder Permutation aus $Q(T^k_{A,o})$ eindeutig ein Klonautomorphismus zugeordnet.

Aus Lemma 4.1. folgt dann die Behauptung.
Sei jetzt $\mathrm{card}(E_k \setminus A) = 2$. Die Elemente aus $E_k \setminus A$ seien c und d. Wir betrachten folgende Funktion aus $T^k_{A,o}$:
$g_{a_c}(x) = \begin{cases} a, & \text{wenn } x \in A \\ c, & \text{wenn } x \notin A. \end{cases}$
Für jedes feste $a \in A$ haben wir folgende eineindeutige Abbildung $g_{a_c}(x) \leftrightarrow c$, $g_{a_d}(x) \leftrightarrow d$.
Wir beweisen, daß bei einem Automorphismus α aus $\mathrm{Aut}(T^k_{A,o})$ $g_{a_c}(x)$ in eine Funktion entsprechender Art übergeht.
Sei $b \in A$, dann ist $g_{a_c}(b) = a$ und $g^{\alpha}_{a_c}(b^{\alpha}) = a^{\alpha}$. Da sich jede Konstante aus A als Bild bei einem Klonautomorphismus α auffassen läßt, ist $g^{\alpha}_{a_c}(x) = a^{\alpha}$, wenn $x \in A$.
Sei $x \notin A$, dann ist $g_{a_c}(x) = c$ für $x = c$ und $x = d$. Angenommen es wäre $g^{\alpha}_{a_c}(d) \neq g^{\alpha}_{a_c}(c)$. Sei f diejenige Funktion aus $Q(\underline{T^k_{A,o}})$, die alle Elemente von A als Fixelemente hat, sowie c mit d vertauscht, so geht f bei Anwendung jedes α in sich über. Wir haben $g_{a_c}(f(x)) = g_{a_c}(x)$, aber $g^{\alpha}_{a_c}(f^{\alpha}(x)) = g^{\alpha}_{a_c}(f(x)) \neq g^{\alpha}_{a_c}(x)$, denn es ist $f(c) = d$ und $g^{\alpha}_{a_c}(f(c)) = g^{\alpha}_{a_c}(d) \neq g^{\alpha}_{a_c}(c)$ nach Voraussetzung. Es muß also $g^{\alpha}_{a_c}(d) = g^{\alpha}_{a_c}(c)$ sein.
Wäre $g^{\alpha}_{a_c}(c) \in A$, so würde sich ergeben: $g^{\alpha}_{a_c}(g^{\alpha}_{a_c}(x)) \equiv a^{\alpha}$, aber $g_{a_c}(g_{a_c}(x)) \equiv g_{a_c}(x)$. Es folgt $g^{\alpha}_{a_c}(x) = g_{a^{\alpha}_c}(x)$, wobei $c^{\alpha} = c$ oder $c^{\alpha} = d$ ist. Auf diese Weise gehört zu jedem α eindeutig eine Abbildung $a \leftrightarrow a^{\alpha}$, $a \in A$, $a^{\alpha} \in A$ und $c \leftrightarrow c^{\alpha}$, $c \notin A$, $c^{\alpha} \notin A$ der Elemente von A und von $E_k \setminus A$ auf sich selbst. Wie im Fall $\mathrm{card}(E_k \setminus A) \neq 2$ folgt dann die Behauptung.
Das Lemma 4.3. gestattet folgende Verallgemeinerung für beliebige präprimale Algebren:
Nach Theorem 2.1. ist eine präprimale Algebra entweder konstantiv und einfach oder erzeugt eine arithmetische Varietät. Algebren der Form $\underline{U^k_{\delta}}$ sind ebenfalls konstantiv, aber nicht einfach.
I. A. Malzev bewies in /11/, daß alle Automorphismen einer abgeschlossenen Klasse von Funktionen $F \subseteq \underline{F}(E_k)$, die alle Konstanten aus E_k enthält, innere Automorphismen sind, d.h,

Lemma 4.4: Für eine konstantive Algebra ist die Gruppe der schwachen Automorphismen isomorph zur Gruppe der Klonautomorphismen.

Aus Theorem 2.1., Lemma 3.4., Lemma 4.3, Lemma 4.4. folgt schließlich:

Theorem 4.5: Für eine beliebige präprimale Algebra $\underline{A} = \langle E_k, F \rangle$ gilt: $Aut^*(\underline{A})$ ist isomorph zu $Aut(\overline{F})$, d.h., die Gruppe der schwachen Automorphismen ist isomorph zur Gruppe der Klonautomorphismen.

(Dieses Theorem wurde von D. Lau in /8/ unter Verwendung des Begriffs innerer Automorphismus formuliert.)

5. Wir untersuchen schwache Automorphismen von Algebren der Form $\underline{U}_\theta^k$.

Lemma 5.1: Für jede präprimale Algebra der Form $\underline{\quad}^k = \langle E_k, F_h \rangle$ gilt: $Q(\underline{U}_\theta^k) = Aut^*(\underline{U}_\theta^k)$, d.h., jeder schwache Automorphismus ist eine eineindeutige unäre algebraische Operation von $\underline{U}_\theta^k$.

Beweis: Wir zeigen, daß es keinen schwachen Automorphismus τ von $\underline{U}_\theta^k$ gibt, der nicht eineindeutige unäre algebraische Operation von U_θ^k ist.

Sei τ eine Permutation, die nicht Operation von $\underline{U}_\theta^k$ ist, d.h. es gibt zwei Elemente a,b, $a \neq b$ mit $a \equiv b(\theta)$, aber $\tau(a) \not\equiv \tau(b)(\theta)$. Sei $\tau^{-1}(a) = d$, $\tau^{-1}(b) = r$, dann ist $r \neq d$. f sei die durch $f(x) = \begin{cases} a, & \text{wenn } x=d \\ b, & \text{sonst} \end{cases}$ definierte Funktion aus U_θ^k. Dann ist $\tau(f(\tau^{-1}(a)) = \tau(f(d)) = \tau(a) \not\equiv \tau(b) = \tau(f(r)) = \tau(f(\tau^{-1}(b)))$, obwohl $a \equiv b(\theta)$ ist, das bedeutet $\tau \cdot f \cdot \tau^{-1}$ ist keine Funktion aus U_θ^k und folglich kein schwacher Automorphismus.

Aus Lemma 3.1., Lemma 4.1. und Lemma 5.1. folgt:

Theorem 5.2: Für eine präprimale Algebra $\underline{A}$, die eine arithmetische Varietät erzeugt, gilt:

$Q(\underline{A}) = Aut^*(\underline{A})$, d.h. jeder schwache Automorphismus ist eine eineindeutige unäre algebraische Operation von $\underline{A}$.

Es gibt auch konstantive einfache präprimale Algebren, für die die Aussage des Theorems 5.2. zutrifft, z.B. $\underline{T^k_{N,k-1}} = \langle E_k, F_1 \rangle$ mit $F_1 = T^k_{N,k-1}$: Klasse, die alle Funktionen einer Variablen und alle Funktionen, die einen Wert auslassen, enthält.

Andererseits gibt es auch präprimale Algebren, für die Theorem 5.2. nicht gilt. Hierzu folgendes Beispiel:

Wir betrachten die Algebra $\underline{A} = \langle \{0,1,2\}, \max(x,y), \min(x,y), m_1, m_2, 0, 1, 2 \rangle$ mit $m_1(x) = \begin{cases} 0, & \text{wenn } x=0 \\ 2, & \text{sonst} \end{cases}$ und $m_2(x) = \begin{cases} 0, & \text{wenn } x=0 \text{ oder } x=1 \\ 2, & \text{sonst} \end{cases}$, max und min beziehen sich auf die Ordnungsrelation, die durch $0 < 1 < 2$ gegeben ist.

Diese Algebra ist präprimal, denn $\langle \max(x,a); \min(x,y); m_1(x); m_2(x); 0; 1; 2 \rangle$ erzeugt die Klasse der bzgl. $0 < 1 < 2$ monotonen Funktionen (/7/).

Sei $\tau = (01)$, $\tau \notin Q(\underline{A})$. Es gilt: $\tau(m_1(\tau^{-1}(x))) = m_2(x)$, $\tau(m_2(\tau^{-1}(x))) = m_1(x)$, $(\max(\tau^{-1}(x), \tau^{-1}(y))) = \min(x,y)$ $\tau(\min(\tau^{-1}(x), \tau^{-1}(y))) = \max(x,y)$. Daher ist τ ein schwacher Automorphismus von $\underline{A}$.

Literatur

/1/ Denecke, K., Algebraische Fragen nichtklassischer Aussagenkalküle, Dissertation, Päd. Hochschule Potsdam, 1978.

/2/ - Präprimale Algebren, Preprint 1978.

/3/ - Präprimale Algebren, die arithmetische Varietäten erzeugen, Preprint 1978.

/4/ Dudek, J., Glazek, K., Some remarks on weak automorphismus, Colloquia Mathematica Societatis Janos Bolyai, 17. Contributions to universal algebra, Szeged, 1975.

/5/ Fried, E. und Pixley, A. F., The dual discriminator function in universal algebra, Preprint, to appear in Acta Math., Szeged.

/6/ Goetz, A., On weak isomorphisms and weak homomorphisms of abstract algebras, Coll. Math. 14 (1966), 163-167.

/7/ Яблонский, С.В., Функциональные построения в к-значной логике, Труды мат. инст. им. В.А. Стеклова, т.51, изд-во АН СССР, 1950.

/8/ Lau, D., Automorphismen auf den maximalen Klassen der k-wertigen Logik, Preprint 1977.

/9/ Мальцев, А.И., Итеративные алгебры и многообразия Поста, алгебра и логика, 5 - 2, 5-24 (1966).

/10/ - Итеративные алгебры Поста, Новосибирск, 1976.

/11/ Мальцев, И.А., Конгруенции и автоморфизмы на клетках алгебр Поста, алгебра и логика, II,6, 666-672, (1972).

/12/ Marszewski, E., Independence in abstract algebras, results and problems, Coll. Math. 14 (1966), 169-180.

/13/ Pixley, A. F., Completeness in arithmetical algebras, Algebra Universalis, Vol. 2, fasz. 2, 1972.

/14/ Senft, J. R., On weak automorphisms of universal algebras, Dissertationes mathematicae, Warsaw, 1970.

eingegangen: 15.9.1978

Anschrift des Verfassers:

K. Denecke
DDR - 4308 Thale
Bert.-Brecht-Str. 55

Rostock. Math. Kolloq. 10, 37 - 41 (1978)

Volker Esperstedt

Fehlerwahrscheinlichkeit bei der Decodierung von Convolutionscodes mittels linearer Filter

Die vorliegende Arbeit knüpft an die Untersuchungen in /1/ und /2/ über die Anwendung linearer Filter bei der Decodierung von Convolutionscodes an. Untersucht wird hier die Wahrscheinlichkeit von Decodierungsfehlern, die bei solchen Entstöralgorithmen auftreten können.
Es sei Y ein endlicher Körper und $[Y^{\omega},+,Y]$ der lineare Raum aller Folgen vom Typ ω über Y. Ein Convolutionscode L über Y (vgl. /3/) ist dann ein linearer, regulärer Unterraum in $[Y^{\omega},+,Y]$. Wir setzen voraus, daß der Übertragungskanal symmetrisch und ohne Gedächtnis sei und daß additiv gestört werde, d.h. $\eta' = \eta + \sigma$, $\eta \in L$, $\sigma \in S$, wobei $S \subseteq Y^{\omega}$ eine Menge von Störfolgen sei. Als $[L,S]$-Filter bezeichnen wir solche Entstörabbildungen, die jede Folge $\eta' \in L+S$ eindeutig in die Summe $\eta' = \eta + \sigma$, $\eta \in L$, $\sigma \in S$ zerlegen und die Codefolge η liefern. Außerdem sollen diese Abbildungen in einem endlichen Automaten realisiert werden können. Existiert für eine Menge S von Störfolgen ein $[L,S]$-Filter, so wollen wir diese Menge als zulässig für den Code L nennen. Die Zulässigkeit von S erfordert u.a. die Regularität von S, was wir später ausnutzen werden. Als lineare Filter bezeichnen wir schließlich $[L,S]$-Filter mit $L+S=Y^{\omega}$ (vgl. /1/).
Wenden wir uns nun dem Problem der Decodierung von Convolutionscodes zu. Hierfür existiert bereits eine Reihe praktikabler und gut analysierter Decodierungsverfahren. Zu diesen zählen u.a. die Algorithmen der sequentiellen Decodierung /4/, der Viterbi-Algorithmus /3/ und die Feedback- oder Syndromdecodierung /5/. Untersucht man nun die Arbeitsweise der linearen Filter, so stellt man fest, daß sie eine verallgemeinerte Feedback-Decodierung darstellen. Der endliche Automat, der das

Filter realisiert, arbeitet sukzessive die empfangene Folge $\eta' \in Y^{\omega}$ ab. Er zerlegt in jedem Schritt ein Teilwort der Länge τ von η' in die Summe von Teilwörtern gleicher Länge einer Code- und einer Störfolge und gibt das Teilwort der Codefolge aus. Hierbei berücksichtigt er nicht nur das in den vorangegangenen Schritten bereits bestimmte Anfangsstück der Codefolge η wie bei der Feedback-Decodierung, sondern auch das bisher bestimmte Anfangsstück der Störfolge ζ. Da L und S regulär sind, erfordert dies nur ein endliches Gedächtnis. Die Zerlegung in jedem Schritt erfolgt analog wie bei der Standard-Decodierung von Gruppencodes der Länge τ (vgl. /6/). Als Code fungiert hierbei die Menge $A_{\tau}(L)$ aller Anfangswörter von L der Länge τ. Als Repräsentanten der Nebenklassen von $A_{\tau}(L)$ im linearen Raum $[Y^{\tau},+,Y]$ aller Wörter der Länge τ über Y dienen die Elemente aus $A_{\tau}(S_j)$, d.h. die Anfangswörter der Länge τ jenen Zustandes S_j von S, der gerade im betreffenden Takt des Automaten berücksichtigt wird (vgl. /2/). Ein Kriterium für die Leistungsfähigkeit von Decodierungsverfahren ist die Fehlerwahrscheinlichkeit. Unter einem Decodierungsfehler zum Zeitpunkt t verstehen wir dabei das Ereignis E_t, bei dem im t-ten Schritt des Decodierungsalgorithmus erstmals eine falsche Entscheidung getroffen wird. Für die Filter heißt das, daß ein Teilwort einer solchen Codefolge ausgegeben wird, die verschieden von der gesendeten ist. Auf Grund der Linearität von L ist es für die Filterdecodierung unwesentlich, welche konkrete Codefolge gesendet wurde. Die Fehlerwahrscheinlichkeit läßt sich deshalb auf folgende Weise angeben:

$$P(E_t) = \sum_i p_{io}(t) P(S_i),$$

wobei $p_{io}(t)$ die Wahrscheinlichkeit dafür ist, daß nach genau t-Takten des Automaten der Zustand S_i von S aus dem Anfangszustand S_o erreicht wurde. $P(S_i)$ sei die Wahrscheinlichkeit eines Decodierungsfehlers des Automaten, wenn der Zustand S_i im entsprechenden Takt berücksichtigt wird.
Die Wahrscheinlichkeiten $p_{io}(t)$ lassen sich nun folgendermaßen berechnen. Es seien S_i für i=0,1,...,N die Zustände der zu-

lässigen, regulären Menge S von Störfolgen. Wir setzen voraus, daß die Folgen $\eta' \in Y^{\omega}$ gleichverteilt seien. Weiterhin sei $\{X_t\}_{t=0}^{\infty}$ eine Folge von Zufallsgrößen mit Werten aus $\{0,1,\ldots,N\}$, wobei $X_t=i$ genau dann gelte, wenn im t-ten Abarbeitungsschritt einer Folge $\eta' \in Y^{\omega}$ durch den Automaten der Zustand S_i erreicht wird. Da der Übergang in den i-ten Zustand zum Zeitpunkt t nur von η' und dem im Zeitpunkt t-1 unmittelbar vorangegangenen Zustand S_j abhängig ist, bilden diese Zufallsgrößen eine Markov-Kette, d.h. für die bedingten Wahrscheinlichkeiten $p_{ij}^{(t)}$ gilt:

$$p_{ij}^{(t)} =_{Df} P(X_t = i \mid X_0=k_0,\ldots,X_{t-2}=k_{t-2},X_{t-1}=j) = P(X_t = i \mid X_{t-1}=j)$$

für alle $t > 0$ und $i,j,k_1 \in \{0,1,\ldots,N\}$.

Sei t_0 die kleinste Zahl, so daß nach t_0 Arbeitsschritten des Automaten bei entsprechender Eingabe η' jeder beliebige S-Zustand erreicht werden kann. Dann bildet die Folge $\{X_t\}_{t=t_0}^{\infty}$ eine homogene Markov-Kette, da auf Grund der zeitinvarianten Arbeitsweise des Automaten die Übergangswahrscheinlichkeiten $p_{ij}^{(t)}$ unabhängig von t sind, d.h. $p_{ij}^{(t)} = p_{ij}$ für $t \geq t_0$. Die Anfangsverteilung ist dann durch $P(X_t = j) = p_{j0}(t_0)$ gegeben, wobei $p_{j0}(t_0)$ die Wahrscheinlichkeit des Überganges aus dem Anfangszustand S_0 in den Zustand S_j nach t_0 Takten des Automaten sei. Mit $P = (p_{ij})_{i,j=0}^{N}$ bezeichnen wir die Matrix der Übergangswahrscheinlichkeiten der S-Zustände nach einem Takt. Dann läßt sich die Matrix $P(t) = (p_{ij}(t))_{i,j=0}^{N}$ der Übergangswahrscheinlichkeiten nach t Takten wie folgt berechnen $P(t)=p^{t-t_0}$ für $t \geq t_0$.

Wie wir sehen, setzt also die Berechnung der unmittelbaren Größen $p_{ij}(t)$ die Kenntnis der Übergangswahrscheinlichkeiten p_{ij} voraus. Diese lassen sich nun wie folgt angeben. Es sei $U(S_j)$ die Menge von Teilwörtern aller möglichen Folgen $\eta' \in Y^{\omega}$, die der Automat entstört, wenn der S-Zustand S_j berücksichtigt wird. Mit $U(S_j \rightarrow S_i)$ wollen wir die Menge jener Teilwörter aus $U(S_j)$ bezeichnen, die den Übergang in S_i im

nächsten Takt des Automaten bewirken. Unter den genannten Voraussetzungen an den Kanal und die empfangenen Folgen η' gilt dann

$$p_{ij} = \frac{\sum\limits_{y \in U(S_j \to S_i)} P(y \mid 0^\tau)}{\sum\limits_{y \in U(S_j)} P(y \mid 0^\tau)}, \quad \text{wobei } P(y \mid 0^\tau) = \prod_{i=1}^{\tau} p(y_i \mid 0)$$

für $y = (y_1, \ldots, y_\tau)$ ist.

Die Mengen $U(S_j)$ und $U(S_j \to S_i)$ lassen sich effektiv angeben, worauf hier aber nicht weiter eingegangen werden soll. Die Wahrscheinlichkeiten $P(S_i)$ von Decodierungsfehlern, wenn die Zustände S_i berücksichtigt werden, lassen sich schließlich analog wie bei der Standard-Decodierung von Gruppencodes berechnen.

In Abhängigkeit vom Kanal und dem verwendeten Code ist es natürlich vorteilhaft, solche linearen Filter für die Decodierung zu verwenden, bei denen $P(E_t)$ minimal ist. Vergleicht man die linearen Filter mit der Feedback-Decodierung nach dem Maximum-Likelihood-Prinzip, so kann man bei günstiger Wahl der zulässigen Menge S von Störfolgen eine Fehlerwahrscheinlichkeit erzielen, die nicht schlechter, in speziellen Fällen sogar besser als die durch die Feedback-Decodierung erreichte ist.

Literatur

/1/ Lindner, R., Staiger, L., Algebraische Codierungstheorie - Theorie der sequentiellen Codierungen, Berlin, 1977.

/2/ Esperstedt, V., Staiger, L., Lineare Filter und ihre Anwendung bei der Decodierung von Convolutionscodes, erscheint in: Schriftenr. d. WBZ f. MKR, TU Dresden, Sekt. Math.

/3/ Viterbi, A. J., Convolutional Codes and Their Performance in Communication Systems, IEEE Trans. Com. Technology, vol. COM-19, Nr. 5, (1971).

/4/ Forney, G. D., Convolutional Codes: III Sequential Decoding, Information and Control 25, (1974).

/5/ Massey, J. L., Threshold Decoding, Cambridge, Mass., 1963.

/6/ Peterson, W. W., Weldon, E. J., Error-Correcting Codes, Second Edition, Cambridge, Mass., 1972.

eingegangen: 15.9.1978

Anschrift des Verfassers:

Volker Esperstedt
Friedrich-Schiller-Universität Jena
Sektion Mathematik
DDR - 69 Jena

Rostock. Math. Kolloq. 10, 43 - 48 (1978)

Hans-Dietrich Hecker

Bemerkungen zu Enumerationssystemen und Zeitmaßen

Ausgehend von Ansätzen von P. Young und anderen für Enumerationen in der Menge aller rekursiv-aufzählbaren Mengen natürlicher Zahlen haben wir in /3/ analoge Fragestellungen im Zusammenhang mit Approximationen von Funktionen durch endliche Funktionstabellen untersucht. Wir werden hier den Begriff des Enumerationssystems bezüglich einer speziellen Standardklasse (vgl. /5/) von rekursiv-aufzählbaren Mengen entwickeln und dabei Resultate von Young und unsere Resultate aus /3/ verallgemeinern. Wir verwenden übliche Bezeichnungen, wie sie etwa im Buch von A. I. Malzew /7/ zu finden sind. Speziell bezeichne $\langle , \rangle$ eine eineindeutige rekursive Abbildung (Cantornumerierung) von $N \times N$ auf N und analog $\langle , \ldots , \rangle$ eine solche von N^n auf N, von der wir der Einfachheit halber für alle natürlichen Zahlen $x_1, \ldots, x_n$ und alle $m \leq n$ die Gültigkeit der Beziehung $\langle\langle x_1, \ldots, x_m\rangle, x_{m+1}, \ldots, x_n\rangle = \langle x_1, \ldots, x_n\rangle$ voraussetzen. Mit $\mathfrak{T}$ bezeichnen wir die Postnumerierung der rekursiv-aufzählbaren Mengen. Wir verwenden außerdem einige Begriffsbildungen aus der Numerierungstheorie.

Ein System $\mathbb{K}$ von rekursiv-aufzählbaren Mengen ist genau dann eine spezielle Standardklasse (SSK), wenn es eine allgemeinrekursive Funktion g gibt, für die gilt:

a) Ist $\mathfrak{T}(x) = \mathfrak{T}_x \in \mathbb{K}$, so ist $\mathfrak{T}_{g(x)} = \mathfrak{T}_x$ $(x \in N)$

b) Für jedes $x \in N$ ist $\mathfrak{T}_{g(x)} \in \mathbb{K}$ und es gilt $\mathfrak{T}_{g(x)} \subseteq \mathfrak{T}_x$.

Das System $\mathbb{P}$ der (Mengen der Cantornummern der Elemente der Graphen der) einstelligen partiell-rekursiven Funktionen fassen wir als Teilmenge des Systems $\mathbb{R}$ aller rekursiv-aufzählbaren Zahlenmengen auf. Dann bilden sowohl $\mathbb{P}$ als auch $\mathbb{R}$ wichtige Beispiele für spezielle Standardklassen. Weitere Beispiele kann man sich leicht aus einer Charakterisierung der SSK von Lachlan (vgl. /2/, II) konstruieren. Grundlegend

für unseren Ansatz ist der folgende Satz. Dabei sei $\mathbb{K}$ eine SSK und ν_0 eine starke Aufzählung des Systems $\mathbb{E} = \mathbb{E}(\mathbb{K})$ aller endlichen Mengen in $\mathbb{K}$ (d.h. es gilt für die Abbildung ν_{fin} aus /2/ die Beziehung $\nu_0 \leq \nu_{fin}$).

Satz 1: Ist die r.a. Menge $R \in \mathbb{K}$ unendlich, so existiert eine ν_0- berechenbare Folge $(F_n)_{n \in \mathbb{N}}$ von Mengen aus $\mathbb{E}$, für die für alle $n \in \mathbb{N}$ $F_n \subsetneq F_{n+1}$ und $\bigcup \{F_n \mid n \in \mathbb{N}\} = R$ gilt.

Eine zweistellige allgemein-rekursive Funktion E heiße Enumerationstechnik von $\mathbb{K}$ bzgl. ν_0, falls gilt:

a) Für alle $i,n \in \mathbb{N}$ ist $\nu_0(E(i,n)) \subseteq \nu_0(E(i,n+1))$

b) Für alle $R \in \mathbb{K}$ existiert ein $i \in \mathbb{N}$ mit $R = R_i = \bigcup_{n=0}^{\infty} \nu_0(E(i,n))$.

Für $\nu_0(E(i,n)) = \bigcup_{\nu=0}^{n} \nu_0(E(i,\nu))$ schreiben wir auch R_i^n.

Die Enumerationstechnik E heiße Gödelenumerationstechnik, wenn durch $\nu(i) := R_i$ eine berechenbare Hauptnumerierung (Gödelnumerierung) von $\mathbb{K}$ gegeben wird.

Wir nennen schließlich ein Tupel $\mathcal{K} = (\mathbb{K}, \nu_0, E, \nu)$ ein Enumerationssystem, wenn gilt:

a) $\mathbb{K}$ ist SSK.

b) Die Numerierung ν_0 des Systems $\mathbb{E}$ ist auf ν_{fin} reduzierbar.

c) E ist eine Gödelenumerationstechnik.

d) ν ist die durch E gegebene Gödelnumerierung von $\mathbb{K}$.

Wir wissen, daß alle Gödelnumerierungen einer SSK untereinander isomorph sind. Wir versuchen, dieses Resultat für Enumerationssysteme zu verschärfen. Dazu betrachten wir zunächst die folgenden Ordnungen $<_i^{\mathcal{K}}$ der Mengen $\nu(i)$ $(i \in \mathbb{N})$:

Für $x,y \in \nu(i)$ setzen wir $x <_i^{\mathcal{K}} y$, falls x in einem früheren Enumerationsschritt als y in der Enumeration i erhalten wird.

Wir definieren genauer:

$$x <_i^{\mathcal{K}} y :\Longleftrightarrow \mu n(x \in \nu_0(E(i,n)) < \mu n(y \in \nu_0(E(i,n)) \quad \text{oder}$$

$$(\mu n(x \in \nu_0(E(i,n)) = \mu n(y \in \nu_0(E(i,n)) \quad \text{und } x < y).$$

Wir fragen für zwei Enumerationssysteme einer SSK $\mathbb{K}$ nach der Existenz eines Isomorphismus, der die gerade definierten Ordnungen der enumerierten Mengen "erhält". Unter einer zusätzlichen Einschränkung gelingt uns tatsächlich der Nachweis

für die Existenz solcher Isomorphismen.
Wir nennen ein Enumerationssystem $\mathcal{K}=(K,\nu_0,E,\nu)$ stark, wenn zu jedem $i \in N$ ein $j \in N$ existiert, für das $<_i = <_j$ und damit $\nu(i)=\nu(j)$ gilt und für das für alle $n \in N$ die Ungleichung

$$|\nu_0(E(j,n+1)) \setminus \nu_0(E(j,n))| \leq 1 \quad \text{erfüllt ist.}$$

Dann gilt folgendes Ordnungsrekursionstheorem:

Satz 2: Sei $\mathcal{K}$ ein starkes Enumerationssystem, dann gibt es zu jeder allgemein-rekursiven Funktion f^2 eine allgemein-rekursive Funktion ξ^1, so daß für alle $x,y \in N$ die Beziehung

$$<^{\mathcal{K}}_{f^2(\xi^1(y),y)} = <^{\mathcal{K}}_{\xi^1(y)} \quad \text{gilt.}$$

Wir beschränken uns jetzt auf starke Enumerationssysteme, was durch die Tatsache gerechtfertigt wird, daß wir von Satz 2 überraschenderweise auch die Umkehrung zeigen konnten:

Satz 3: Jedes Enumerationssystem, für das obiges Ordnungsrekursionstheorem gilt, ist stark.

Es läßt sich nun unter Benutzung unseres Satzes 2 für starke Enumerationssysteme ein "Ordnungstranslationstheorem" und ein "Ordnungspaddinglemma" beweisen, woraus dann folgender Satz folgt:

Satz 4: Seien $\mathcal{K}=(K,\nu_0,E,\nu)$ und $\mathcal{K}^+=(K,\nu_0,E^+,\nu^+)$ zwei starke Enumerationssysteme. Dann existiert eine rekursive Permutation t von N, so daß für alle $y \in N$ die Beziehung $<^{\mathcal{K}}_y = <^{\mathcal{K}^+}_{t(y)}$ gilt.

Offen bleibt die Frage, ob der Satz 4 für beliebige E-Systeme richtig ist, d.h. auch für solche Systeme, für die das Ordnungsrekursionstheorem nicht gilt.
Als nächstes erinnern wir an den Begriff des Zeitmaßes Φ^2 nach M. Blum für Gödelnumerierungen $(\varphi_i)_{i \in N}$ der Menge der einstelligen partiell-rekursiven Funktionen. Jedes Zeitmaß ist eine partiell-rekursive Funktion und es gilt für alle $i,x \in N$:

a) $\Phi^2(i,x)\downarrow \iff \varphi_i(x)\downarrow$

b) $\{<i,x,y> \mid \Phi^2(i,x) = y\}$ ist rekursiv.

Wir verallgemeinern diesen Begriff und nennen eine zweistellige partiell-rekursive Funktion Φ^2 ein verallgemeinertes Zeitmaß für eine spezielle Standardklasse $\mathbb{K}$ mit einer Gödelnumerierung ν, wenn für alle $i,x \in \mathbb{N}$ gilt

a) $\Phi(i,x)\downarrow \iff x \in \nu(i)$

b) $\{\langle i,x,y\rangle \mid \Phi(i,x) = y\}$ ist rekursiv.

Sei Φ^2 ein verallgemeinertes Zeitmaß für $(\mathbb{K},\nu)$ und sei f^2 eine allgemein-rekursive im zweiten Argument monoton wachsende Funktion. Sei $M_\Phi(f,i,n) := \{x \mid \Phi^2(i,x) \leq n \text{ und } x<n+f(i,n)\}$. Man zeigt übrigens leicht, daß es zu jedem Paar $(\mathbb{K},\nu)$ ein verallgemeinertes Zeitmaß gibt. Es gilt weiter folgender Satz:

Satz 5: Ist $\nu_0 \leq \nu_{fin}$ eine starke Numerierung der Menge aller endlichen Mengen in $\mathbb{K}$. Sei Φ^2 ein verallgemeinertes Zeitmaß für $(\mathbb{K},\nu)$. Dann gibt es eine Gödelenumerationstechnik E, so daß für alle $n,i \in \mathbb{N}$ die Beziehungen $\nu_0(E(i,n)) \supseteq M_\Phi(f,i,n)$ und $\bigcup_{\mu=0}^{\infty} M(f,i,\mu) = \bigcup_{\mu=0}^{\infty} \nu_0(E(i,\mu)) = \nu(i)$ gelten.

Satz 6: Zu jeder Gödelenumerationstechnik E für eine SSK $\mathbb{K}$ und zu jeder Numerierung $\nu_0 \leq \nu_{fin}$ der Menge der endlichen Mengen in $\mathbb{K}$ gibt es eine im zweiten Argument monotone rekursive Funktion f und ein verallgemeinertes Zeitmaß Φ, so daß für alle $i,n \in \mathbb{N}$ $\nu_0(E(i,n)) = M_\Phi(f,i,n)$ gilt.

Aus dem Satz 6 folgt, daß man sich jede Gödelenumerationstechnik in der beschriebenen Weise aus einem verallgemeinerten Zeitmaß definiert denken kann. Sehen wir uns speziell die Verhältnisse für $\mathbb{K} = \mathbb{P}$ an. Fassen wir $\mathbb{P}$ wie oben als Teilmenge von $\mathbb{R}$ auf, welche Beziehungen bestehen dann zwischen Zeitmaßen und verallgemeinerten Zeitmaßen? Jedes Zeitmaß ergibt sofort ein verallgemeinertes Zeitmaß. Wenn jedoch Ψ ein verallgemeinertes Zeitmaß für $(\mathbb{P},\varphi)$ ist, dann ist die Funktion Φ :

$$\Phi(i,x) := \begin{cases} \Psi(i,x,y), & \text{falls } \varphi_i(x) = y \\ \text{undefiniert} & \text{sonst} \end{cases} \quad \text{i.a. kein Zeitmaß}$$

für $(\mathbb{P},\varphi)$. Die Menge $\{\langle i,x,y\rangle \mid \Phi(i,x)=y\}$ kann sogar jeden r.a. Entscheidbarkeitsgrad annehmen. Wir bemerken noch, daß man sich jede E-Technik E für $\mathbb{P}$ aus einem Zeitmaß (Φ,φ) in der

Form $\nu_0(E(i,n))=\{\langle x,\varphi_i(x)\rangle \mid \Phi(i,x)\leq n \;\&\; x<n+f(i,n)\}$ mit einer geeigneten im 2. Argument wachsenden rekursiven Funktion f gegeben denken kann. Umgekehrt kann man sich jedes "raumkompliziertheitsähnliche" Maß (Φ,φ) aus einer GET E in der Form

$$\Phi(i,x) = \mu n(x \in l(\nu_0(E(i,n))) \qquad (+)$$

gegeben denken (l liefert dabei die Menge der linken Komponenten der Numerierung $\langle,\rangle$). Genauer gilt sogar folgender Satz:

Satz 7: Ein Blummaß (Φ,φ) läßt sich genau dann durch eine GET in der Form (+) definieren, wenn es eine Funktion $h \in A^2$ gibt (d. h. h ist rekursiv), für die für alle i,x die Beziehung $h(i,\Phi(i,x)) > x$ gilt. Dabei bezeichne ν_0 eine beliebige starke Numerierung aller endlichen Funktionstabellen.

Weitere Resultate über Enumerationssysteme, die insbesondere Charakterisierungen von Mengen betreffen, die in einem Enumerationssystem innerhalb einer gemeinsamen rekursiven Schranke enumeriert werden können, werden in einem Artikel in der "Zeitschrift für mathematische Logik und Grundlagen der Mathematik" erscheinen.

Literatur

/1/ Blum, M., A machine-independent theory of the comp exity of recursive functions, J. Assoc. Comp. Mach., 14 (1967), 332 - 336.

/2/ Eršov, Ju. L., Theorie der Numerierungen I/II, Zeitschr. Math. Logik und Grundl. d. Math. 19/21 (1973/1975) 289 - 388/473 - 584.

/3/ Hecker, H. D., Zur Enumerationskomplexität von Funktionstabellen, Vortrag auf der Tagung "Diskrete Math. und Anwendungen in der Math. Kybernetik", Berlin 1977.

/4/ Helm, J., Meyer, A., Young, P., On orders of translations and enumerations, Pacific J. Math., 46, 1 (1973), 185 - 195.

/5/ Lachlan, A. H., On standard classes of r.e.sets, Zeitschr. Math. Logik und Grundl. d. Math. 10 (1964), 23 - 42.

/6/ Young, P., Toward a theory of enumerations, J. Assoc. Comp. Mach., Bd. 16 (1969), 328-341.

/7/ Malzew, A. I., Algorithmen und rekursive Funktionen, DVdW Berlin, 1974.

eingegangen: 15.9.1978

Anschrift des Verfassers:

Dr. H.-D. Hecker
Ernst-Moritz-Arndt-Universität
Sektion Mathematik
DDR - 22 Greifswald
Jahnstr. 15 a

Rostock. Math. Kolloq. 10, 49 - 52 (1978)

Peter Heinrich

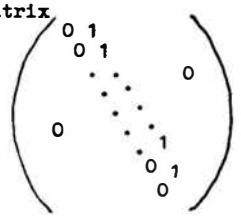
Zur Charakterisierung der Jordanschen Normalform der Adjazenzmatrix eines gerichteten zyklenfreien Graphen

Sind A, B zwei n-reihige Matrizen, dann sind diese bekanntlich ähnlich genau dann, wenn sie die gleiche Jordansche Normalform besitzen. Wählt man die zueinander ähnlichen Matrizen A, B speziell als Adjazenzmatrizen gerichteter Graphen G_1, G_2 (wir sagen dann, daß G_1, G_2 zueinander ähnlich sind), so entsteht sofort das folgende Problem: Welche gemeinsamen graphentheoretischen Eigenschaften besitzen G_1 und G_2, falls G_1 und G_2 zueinander ähnlich sind? Solche Untersuchungen wurden vom Verfasser durchgeführt; auf Grund der beschränkten Platzkapazität soll hier nur das Ergebnis im Fall gerichteter zyklenfreier Graphen genannt und an einem kleinen Beispiel veranschaulicht werden; auf Beweise muß dabei verzichtet werden.
Ist G ein gerichteter Graph, so soll G zyklenfrei heißen genau dann, wenn der aus G durch Vernachlässigung der Kantenorientierung jeder Kante von G entstehende ungerichtete Graph kreisfrei ist und G selbst keine Schlingen enthält. Weiter soll ein gerichteter zyklenfreier Graph GN ein Normalformgraph des Typs $(v_1,\ldots,v_r)$ heißen genau dann, wenn gilt:

1. GN besitzt genau r Komponenten $K_1,\ldots,K_r$.
2. Jede Komponente von GN ist ein (gerichteter) Weg.
3. v_i ist die Länge des Weges K_i ; $i = 1,\ldots,r$.

Ist GN ein Normalformgraph des Typs $(v_1,\ldots,v_r)$, so können wir die (v_i+1)-reihige Matrix

$A_i =$



als Adjazenzmatrix von K_i auffassen ($i = 1,\ldots,r$) und damit die verallgemeinerte Diagonalmatrix $N = (A_1, A_2, \ldots, A_r)$ als Adjazenzmatrix von GN.
Sei G ein beliebiger gerichteter zyklenfreier Graph. Unter einem Weg-System W von G der Länge l verstehen wir ein System von Wegen $w_1,\ldots,w_s$ von G mit folgenden Eigenschaften:
1. $w_1,\ldots,w_s$ sind paarweise knotenfremd.
2. $l = l(w_1) + l(w_2) + \ldots + l(w_s)$, wobei $l(w_i)$ die Länge des Weges w_i in G ist ($i = 1,\ldots,s$).

Wir wollen uns nun unserer Problemstellung zuwenden. Sei G ein gerichteter zyklenfreier Graph mit $n \geq 1$ Knoten. Dann gilt folgender

Satz: Sei l_{max} die Länge eines Weg-Systems größter Länge in G, k_o die Minimalzahl von Komponenten, die ein System der Länge l_{max} in G besitzt und $l_1,\ldots,l_{k_o-1}$ die Maximallängen eines Weg-Systems von G mit $1,\ldots,k_o-1$ Komponenten. Dann ist die Adjazenzmatrix des Normalformgraphen GN des Typs $(v_1,\ldots,v_r)$ die Jordansche Normalform der Adjazenzmatrix von G genau dann, wenn gilt:
1. $r = n - l_{max}$
2. $v_1 = l_1$
3. $v_i = l_i - l_{i-1}$ für $i = 2,\ldots,k_o-1$
4. $v_{k_o} = l_{max} - l_{k_o-1}$
5. $v_i = 0$ für $i = k_o + 1,\ldots,r$.

Nun gilt auch die Umkehrung obigen Satzes in folgender Form:

Satz: Ist G ein gerichteter zyklenfreier Graph mit n Knoten und N die Jordansche Normalform der Adjazenzmatrix von G (dabei soll N als obere Dreiecksmatrix geschrieben sein), so gilt:
1. N ist Adjazenzmatrix eines Normalformgraphen GN.
2. Hat GN den Typ $(v_1,\ldots,v_r)$ und ist dabei die Numerierung der v_i so gewählt, daß $v_1 \geq v_2 \geq \ldots \geq v_{k_o} > v_{k_o+1} = \ldots = v_r = 0$, so gilt:

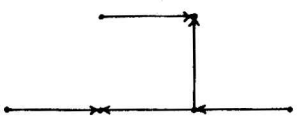
a) Die größte Länge eines Weg-Systems in G ist $l_{max} = n-r$.

b) Die Minimalzahl von Komponenten, die ein Weg-System der Länge l_{max} in G besitzt, ist k_o .

c) Die Maximallänge l_t, die ein System mit t Komponenten für $t = 1,\ldots,k_o-1$ in G besitzt, ist $l_t = v_1 + \ldots + v_t$, d.h. $l_1 = v_1$, $l_t = v_t + l_{t-1}$ für $t = 1,\ldots,k_o-1$.

Beide Sätze zusammen lösen damit in unserem Spezialfall das Problem vollständig.

Beispiel: Sei G der folgende Graph:



Man erkennt sofort: $l_{max} = 3$, $k_o = 2$, $l_1 = 2$. Dann ist $v_1 = 2$, $v_2 = 1$, $r = 3$, $v_3 = 0$. Damit ist die Adjazenzmatrix des Normalformgraphen GN vom Typ (2,1,0), also die Matrix

$$N = \begin{pmatrix} 010 & 00 & 0 \\ 001 & 00 & 0 \\ 000 & 00 & 0 \\ 000 & 01 & 0 \\ 000 & 00 & 0 \\ 000 & 00 & 0 \end{pmatrix},$$

die gesuchte Jordansche Normalform.

Umgekehrt hat jeder gerichtete zyklenfreie Graph, dessen Adjazenzmatrix gerade N als Jordansche Normalform besitzt, die charakteristischen Größen $l_{max} = 3$, $k_o = 2$ und $l_1 = 2$ und der Graph GN vom Typ (2,1,0) ist ein besonders einfacher Repräsentant in der Klasse aller zu obigem G ähnlichen Graphen.

Literatur

/1/ Chen, W. K., Applied Graph Theory, N.-H. Publishing Company, 1971.

/2/ Heinrich, P., Eine Beziehung zwischen den Weg-Kreis-Systemen in einem gerichteten Graphen und den Unterdeterminanten seiner charakteristischen Matrix, erscheint in den Mathemat. Nachr.

/3/ Heinrich, P., Eine Charakterisierung der Jordanschen Normalform spezieller Klassen gerichteter Graphen, unveröffentl. Manuskript.

/4/ Sachs, H., Beziehungen zwischen den in einem Graphen enthaltenen Kreisen und seinem charakteristischen Polynom, Publ. Math. (Debrecen) 11 (1963), 119-134.

eingegangen: 15.9.1978

Anschrift des Verfassers:

Dr. rer. nat. Peter Heinrich
Bergakademie Freiberg
Sektion Mathematik
DDR - 92 Freiberg
Bernhard-von-Cotta-Str. 2

Rostock. Math. Kolloq. 10, 53 - 62 (1978)

Albrecht Hübler, Reinhard Klette, Rolf Lindner

Elementare Operationen auf Binärbildern

Ein aktuelles Kernproblem bei der Schaffung schneller digitaler Bildinterpretationssysteme ist die Entwicklung parallel arbeitender Prozessoren für Transformationen diskreter Bilder. Ergebnisse erster Erkundungssysteme zur Parallelarbeit, wie etwa die Computer der ILLIAC-Reihe, STARAN oder TSE-Computer, haben zu einer beschleunigten Hardware- und Software-Entwicklung auf dem Gebiet der Parallelabarbeitung geführt. In der vorliegenden Arbeit werden, Definitionen und Ergebnisse aus /2/, /3/, /4/, /5/ referierend, parallele Operationen auf Binärbildern diskutiert.
Die Punkte eines gleichmäßigen, orthogonalen d×d-Rasters sind entweder schwarz (Code 0, Untergrundpunkt) oder weiß (Code 1, Objektpunkt) gefärbt. Es sei

$$\underline{B} := \left\{ x = (x_{rs})_{1 \le r,s \le d} : \forall r \forall s \left(x_{rs} \in \{0,1\} \right) \right\}$$

die Klasse aller B i n ä r b i l d e r . Für x aus $\underline{B}$ bezeichnet x_{rs} oder x(r,s) die Färbung von x im Punkt (r,s). Wir vereinbaren $x_{rs} := 0$, falls (r,s) ein Punkt außerhalb des d×d-Rasters ist. Die B i l d g r ö ß e d sei fixiert; für ein gewisses $e \ge 2$ sei $d = 2^e$. Gegenwärtig sind Bildgrößen $d = 2^e$ mit $5 \le e \le 8$ von besonderem praktischen Interesse. Für diese Bildgrößen erreichten Laborsysteme zur parallelen Verarbeitung von Binärbildern gegenüber sequentiell arbeitenden Systemen wesentliche Beschleunigungen bei der Ausführung von Binärbildtransformationen /8/.
Ein Binärbild x heißt U n t e r b i l d eines Binärbildes y ($x \le y$), falls stets $x_{rs} \le y_{rs}$ erfüllt ist. Diese Relation definiert auf $\underline{B}$ eine BOOLEsche Algebra. Es sei $et(x,y) := \inf\{x,y\}$ und $vel(x,y) := \sup\{x,y\}$. Das Komplement von x in $[\underline{B}, \le]$ sei mit non(x) bezeichnet. Die Funktionen et,vel,non (kurz $\wedge$, $\vee$, $^-$) sind erste Beispiele BOOLEscher Operatoren auf $\underline{B}$.

1. Operatoren

Es sei $\underline{F}$ die Klasse aller Operatoren auf $\underline{B}$; ein Operator $f \in \underline{F}^{(n)}$ bildet eindeutig von $\underline{B}^n$ in $\underline{B}$ ab. Durch elementare Betrachtungen bestimmt man die Mächtigkeit von $\underline{F}^{(n)}$ zu

$$\text{card } \underline{F}^{(n)} = 2^{d^2} \cdot 2^n \cdot d^2 \quad , \text{ für } n \geq 0.$$

Die Klasse BOOL := $[\{non,et,vel\}]$ aller BOOLEschen Operatoren auf $\underline{B}$ ist echt in $\underline{F}$ enthalten; es gilt

$$\text{card BOOL}^{(n)} = 2^{2^n} \quad , \text{ für } n \geq 0.$$

Die alleinige Verwendung BOOLEscher Operatoren ist für die anstehenden Aufgaben der Bildverarbeitung unzureichend. Für die Bearbeitung vielfältigster Aufgabenstellungen der digitalen Bildverarbeitung ist die Anwendung gewisser geometrischer Operatoren erforderlich.

Verschiebungen: Die Operatoren slr, sll, slu, sld $\in \underline{F}^{(1)}$ bewirken die Verschiebung eines Binärbildes um je eine Spalte oder Zeile nach rechts, links, oben oder unten. Die aus dem d×d-Raster herausgeschobenen Spalten oder Zeilen werden vergessen, die frei werdenden Spalten oder Zeilen sind jeweils schwarz gefärbt. Es ist zum Beispiel

slr(x)(r,s):=if $1 \leq r \leq d$ and $2 \leq s \leq d$ then x(r,s-1) else 0 fi .

Es sei SLD := $\{slr,sll,slu,sld\}^+$ die Klasse aller Mehrfachverschiebungen.[1]

Satz 1: /4, Hübler, Klette/ card SLD = $(\frac{1}{3} d^3 + \frac{1}{2} d^2 + \frac{1}{6} d)^2$.

Zyklische Verschiebungen: Mit cyr, cyl, cyu, cyd bezeichnen wir zyklische Verschiebungen um jeweils eine Spalte oder Zeile nach rechts, links, oben oder unten. Die herausgeschobenen Spalten oder Zeilen werden für die auf der Gegenseite frei werdende Spalte oder Zeile wieder hereingeschoben. Es gilt zum Beispiel

cyr(x)(r,s):=if s=1 then x(r,d) else slr(x)(r,s) fi .

---

1 $A^+ := A^* - \{\lambda\}$, wobei λ das leere Wort bezeichnet.

Es sei CYC := $\{cyr,cyl,cyu,cyd\}^+$ die Klasse aller zyklischen Verschiebungsoperatoren.

Satz 2: /4/ card CYC = d^2 .

Drehungen: Der Operator rot bezeichnet die Drehung eines Binärbildes um 90^0 im mathematisch positiven Sinn, rot(x)(r,s) = x(d-s+1,r). Dann ist ROT := $\{rot\}^+ = \{id,rot,rot^2,rot^3\}$.

Invertierungen: Durch die vertikale Invertierung ver (horizontale Invertierung hor) werden Binärbilder spaltenweise (zeilenweise) invertiert. Es gilt ver(x)(r,s) = x(r,d-s+1) und hor(x)(r,s) = x(d-r+1,s). Die Spiegelung dia(x)(r,s) := x(s,r) eines Binärbildes an der Hauptdiagonalen entspricht der Matrizentransposition. Es sei INV := $\{ver,hor,dia\}^+$ die Menge aller mehrfachen Invertierungen. Diese Operatorenklasse besteht aus genau acht Operatoren: id, ver, hor, dia, rot, rot^2, rot^3 und rot·hor .

Vergrößerungen und Verkleinerungen: Der Vergrößerungsoperator mag vergrößert die Färbung der Bildmitte $\{(r,s) : \frac{d}{4}+1 \leq r,s \leq \frac{3}{4}d\}$ unter Beibehaltung der Färbungsstruktur auf das gesamte d×d-Raster. Die Färbung eines Bildpunktes der Bildmitte wird hierbei auf je vier Bildpunkte abgebildet. Der Verkleinerungsoperator dmg bildet die Färbung des d×d-Rasters auf die Bildmitte ab; ein Rasterpunkt wird genau dann weiß gefärbt, falls von den vier entsprechenden Urbildpunkten mindestens zwei weiß gefärbt sind. Außerhalb der Bildmitte ist ein Binärbild dmg(x) stets schwarz gefärbt. Für die Klasse MDM := $\{mag,dmg\}^+$ aller mehrfachen Vergrößerungen/Verkleinerungen gilt (Es ist e = $\log_2 d$).

Satz 3: /4/ card MDM = $\frac{1}{3}e + \frac{1}{2}e^2 + \frac{1}{6}e^3$.

Geometrische Operatoren: GEO := $\{slr,sll,slu,sld,cyr,cyl,cyu,cyd,rot,ver,hor,dia,mag,dmg\}^+ = \{slr,cyr,rot,ver,mag,dmg\}^+$ bezeichnet die Klasse aller geometrischen Operatoren über B. Es ist GEO $\subsetneq \underline{P}^{(1)}$ und GEO $\cap$ BOOL = $\{k_0,id\}$; die Operatoren non und k_1 aus $BOOL^{(1)}$ sind keine geometrischen Operatoren. Verschiebungen und zyklische Verschiebungen um 2^m , $0 \leq m \leq e$, Zeilen oder Spalten (slr $2^m(x)$,cyr $2^m(x)$,...) sowie die Operatoren

rot, ver, hor, dia, mag und dmg werden e l e m e n t a r e geometrische Operatoren genannt. Die Menge OP bestehe aus genau allen elementaren geometrischen Operatoren sowie den BOOLEschen Operatoren non, et, vel. Die Operatoren der Menge OP werden als Elementaroperationen eines parallel arbeitenden Bildcomputers betrachtet; ihre Hardware-Realisierung ist abgesichert /8/.

2. Lokale Operatoren

Die Realisierung vieler für die Bildverarbeitung interessanter Transformationen (Konturbestimmung, Entstörung u.ä.m.) entspricht der Ausführung einer lokalen Operation /1/. Der Wert eines lokalen Operators in einem Punkt (r,s) ist ausschließlich von der Färbung in einer (für den jeweiligen Operator fixierten) Umgebung des Punktes (r,s) abhängig. Wir betrachten lokale Operatoren, deren Werte durch eine jeweils fixierte BOOLEsche Funktion gebildet werden.
Ein n-stelliger Operator f heiße im Sinne von /5/ genau dann l o k a l e r O p e r a t o r d e r O r d n u n g m ($m \geq 0$), falls eine Umgebung V und eine BOOLEsche Funktion f' existieren, die folgenden Forderungen genügen:

1. $V = [(r_1,s_1),(r_2,s_2),\ldots,(r_m,s_m)]$;
2. $f' : \{0,1\}^{m \cdot n} \rightarrow \{0,1\}$;
3. für $1 \leq r,s \leq d$ ist $f(x_1,x_2,\ldots,x_n)(r,s) = f'(x_1(r+r_1, s+s_1),x_1(r+r_2,s+s_2),\ldots,x_1(r+r_m,s+s_m),x_2(r+r_1,s+s_1),x_2(r+r_2, s+s_2),\ldots,x_n(r+r_m,s+s_m))$. Für Punkte (r,s) außerhalb des d×d-Rasters ist $f(x_1,x_2,\ldots,x_n)(r,s) = 0$.

Wir sagen, daß der Operator f durch die Umgebung V und die BOOLEsche Funktion f' erzeugt wird und schreiben $f = \langle f',V \rangle$. Für $m \geq 0$ sei LOC_m die Klasse aller lokalen Operatoren der Ordnung m. Die Klasse BOOL ist zum Beispiel in LOC_1 enthalten. Der nicht in BOOL liegende Operator slr ist gemäß $slr = \langle id, [(0,-1)] \rangle$ aus LOC_1. Der Operator $cyr = \langle vel, [(0,d-1),(0,-1)] \rangle$ liegt in LOC_2 und nicht in LOC_1. Es sei $LOC := \bigcup_{m \geq 0} LOC_m$

die Klasse aller lokalen Operatoren über $\underline{B}$.

Satz 4: /4/ (i) Für $m<(2d-1)^2$ gilt $LOC_m \subset LOC_{m+1}$.
(ii) Für $m \geq (2d-1)^2$ ist bereits $LOC_m = LOC$.
(iii) Die Klasse LOC ist echt in der Klasse $\underline{F}$ aller Operatoren enthalten; es gilt zum Beispiel

$$\text{card } LOC^{(1)} = \text{card } \underline{F}^{(1)} / \; 2^{(d-1)\cdot 2^{(d-1)^2}\cdot(d\cdot 2^d - d+1)}.$$

3. Programme zur Parallelverarbeitung von Binärbildern

Als Bildcomputer wird eine Maschine mit direktem Zugriff verwendet; der Speicher besteht aus Registern des Vektor-, Index- und Matrixtyps. Dieser Bildcomputer stellt ein Hybrid-System einer Vektormaschine /6/ mit einer Matrixmaschine /3/ dar. Es bezeichnen a,b,c stets Vektorregister, i,j,k stets Indexregister und x,y,z stets Matrixregister. Die Register können indiziert sein; es mögen im Speicher potentiell unendlich viele Register jedes Typs bereitstehen. In den Vektorregistern steh jeweils ein Vektor aus $\{0,1\}^d$. Der Wert $\langle a \rangle$ eines Vektorregisters ist jeweils diejenige natürliche Zahl n, deren Dualdarstellung von rechts nach links geschrieben in a momentan gespeichert ist. Folglich gilt immer $\langle a \rangle \leq 2^d-1$. Indexregister speichern je einen Vektor aus $0^* \cup 0^*10^*$ der Länge e+1. Mithin ist stets $\langle i \rangle = 0$ oder $\langle i \rangle = 2^m$, mit $0 \leq m \leq e$. In den Matrixregistern steht jeweils ein Binärbild. Speichert ein Matrixregister x ein gewisses Binärbild w, so stehen in der untersten Zeile von x die Werte $w_{d,1}, w_{d,2}, \ldots, w_{d,d}$ in dieser Anordnung. Als Ein- oder Ausgaberegister können Vektor- oder Matrixregister Verwendung finden. Alle nicht als Eingaberegister verwendeten Register sind zu Beginn einer Berechnung mit 0 geladen.

Elementare Instruktionen

int a = n; **int** i = 2^m;	Konstantenzuweisungen;
image x = w;	
a:=b; i:=j; x:=y;	direkte Zuweisungen;
a:=x;	unterste Zeile von x wird in a übertragen;

x:=a;	a wird in die unterste Zeile von x übernommen, Rest von x ist gleich 0;
a:=$\overline{b}$; a:=bʌc; a:=bvc; a:=slr i(b); a:=sll i(b);	Verschiebungen um ⟨i⟩ Bits nach rechts/links;
i:=slr(j); i:=sll(j);	Verschiebungen um je ein Bit nach rechts/links;
x:=op_1(y); x:=op_2(y,z);	$op_1, op_2 \in OP$, Verschiebungsdistanzen werden durch Indexregister angegeben;

Die Zeit einer Berechnung ist die Anzahl der auszuführenden elementaren Instruktionen plus die Anzahl der auszuführenden

Elementaren Testinstruktionen

if a=0 then ...; if a≠0 then ...; if i=0 then ...; if i≠0 then ...;

Die Ausführung von Tests if x=0 then ...; oder if x≠0 then benötigt jeweils e Zeiteinheiten (vgl. /8/).
Der Raum einer Berechnung ist die Anzahl der Bits jener Register, die während der Berechnung aktiv verwendet werden.
Der Bildcomputer berechnet gemäß einem eingegebenen Programm gewisse Funktionen f: $\underline{V}^{n_1} \times \underline{B}^{m_1} \longrightarrow \underline{V}^{n_2} \times \underline{B}^{m_2}$, wobei $\underline{V} := \{0,1\}^d$ sei. Eine durch den Bildcomputer berechnete Funktion f liegt in der K o m p l i z i e r t h e i t s k l a s s e $O(T(d)) - OP' - O(S(d))$, falls f durch alleinige Verwendung von Operationen aus $OP' \subseteq OP$ und beliebigen Operationen auf Vektor- oder Indexregistern in einer Zeit $O(T(d))$ und gleichzeitig im Raum $O(S(d))$ berechnet wird.

Beispiel: Es sei eine gewisse Funktion dup : $\underline{V} \longrightarrow \underline{B}$ zu berechnen; im Binärbild dup(a) stehe in jeder Zeile der Inhalt des Vektorregisters a. Folgendes Programm realisiert diese Berechnung /4/:

```
begin vector a; index i; image y,z; read a; int i=1; y:=a;
      M: if i=0 then print y else z:=slu i(y); y:=yvz;
         i:=sll(i); goto M fi;
end
```

Diesem Programm zufolge liegt die Funktion dup in der Kompliziertheitsklasse $O(e) - \{vel, slu\ i\} - O(d^2)$.

Satz 5: /4/, /5/ Die Operatorenklasse $LOC_m^{(1)}$ ist enthalten in der Klasse $O(\max\{m \cdot e, 2^m/m\}) - \{non, et, slr, i, sll\ i, slu\ i, sld\ i\} - O(m \cdot d^2)$.

Für konkrete lokale Operatoren ist die Zeitabschätzung dieses Satzes oftmals bedeutend zu verbessern. Wir betrachten ein wichtiges Beispiel, die

Konturbestimmung: In einem Binärbild seien gewisse Figuren abgebildet, deren Konturbild bestimmt werden soll. Die Konturbestimmung kann zu den Standardaufgaben der Bildverarbeitung gezählt werden, da eine Reihe weiterer Primärmerkmale (zum Beispiel der Umfang der Figuren) in enger Beziehung zur Kontur stehen. - Durch $h(x)(r,s) := x(r,s+1) + x(r,s-1) + x(r+1,s) + x(r-1,s)$ wird die Summe der Helligkeitswerte in der 4-Nachbarschaft eines Punktes (r,s) bezüglich eines Binärbildes x bestimmt. Im Sinne von /7/ sei

$$kontur(x)(r,s) := \begin{cases} 1 & , \text{ falls } x(r,s)=1 \ \&\ h(x)(r,s) \leq 3 \\ 0 & , \text{ sonst.} \end{cases}$$

Der Operator kontur : $\underline{B} \longrightarrow \underline{B}$ ist aus LOC_5 und Satz 5 zufolge in einer Zeit $O(e)$ auf dem Bildcomputer zu realisieren. Durch eine direkte Analyse des Operators kontur erhält man demgegenüber die Möglichkeit einer Realisierung in konstanter (!) Zeit $O(1)$. Das folgende einfache Programm berechnet den Operator kontur /2/:

```
begin image x,y,z; read x; y:=slr(x); z:=x∧y; a:=sll(x),
      z:=z∧y; y:=slu(x); z:=z∧y; y:=sld(x); z:=z∧y;
      z:=z̄; y:=z∧x; print y
end
```

Der Operator kontur liegt demzufolge bereits in der Kompliziertheitsklasse $O(1) - \{non, et, slr, sll, slu, sld\} - O(d^2)$.

Die Konturbestimmung im Sinne obiger Definition (erweiterte Verfahren sind zum Beispiel für die Ausgabe gewisser Kontur-

koordinaten anzugeben) würde auf einem streng sequentiell arbeitenden System eine Zeit $O(d^2)$ erfordern. Der Beschleunigungsfaktor einer Realisierung auf dem dargelegten Computermodell gegenüber einer Realisierung auf sequentiell arbeitenden Systemen ist vom Grad der jeweiligen Parallelisierungsmöglichkeiten abhängig. Eine Aufgabe mit einem gegenüber der Konturbestimmung geringeren Beschleunigungsfaktor ist die

Bestimmung minimaler weißer Pfade: Der zu realisierende Operator wpath : $\underline{B} \longrightarrow \underline{B}$ soll folgendes leisten:

1. Existieren in x weiße Pfade von der ersten zur d-ten Spalte, so sind in wpath(x) alle derartigen Pfade minimaler Länge weiß gefärbt; in der ersten und d-ten Spalte von wpath(x) sind genau die Ein- und Ausgänge dieser Pfade weiß gefärbt.
2. Existiert in x kein solcher weißer Pfad von links nach rechts, so ist wpath(x) = $\underline{0}$.

Die Länge derartiger weißer Pfade ist durch $\frac{d}{2}(d-1)+2$ nach oben beschränkt. - Wir beschreiben den zur Realisierung von wpath in /2/ angegebenen Lösungsweg: Im Binärbild mask sei die erste Spalte weiß und die restlichen Spalten seien schwarz. Mittels x $\wedge$ mask werden in x die potentiellen Pfadanfänge ausgeblendet. Unter Verwendung der Operatoren slr,sll,slu,sld,et und vel verfolgen wir schrittweise und parallel alle möglichen Fortsetzungen dieser weißen Pfadanfänge. Dabei wird jeweils ein Test ausgeführt, ob eine dieser Fortsetzungen (oder mehrere gleichzeitig) bereits die d-te Spalte erreicht hat. Dieser Test ist bei Verwendung von mask,rot,et und einem Null-Test für ein Vektorregister in konstanter Zeit auszuführen. Fällt dieser Test zum ersten Mal positiv aus, so haben wir in der d-ten Spalte genau alle Ausgänge der gesuchten minimalen Pfade markiert. Wir führen das geschilderte Verfahren auf dem derart bereits reduzierten Bild ein zweites Mal durch, diesmal in der d-ten Spalte beginnend, und erhalten beim erstmaligen Erreichen der ersten Spalte die Markierung aller Eingänge der gesuchten minimalen Pfade. STOP. Führt das geschilderte Verfahren beim ersten Durchlauf von links nach rechts nach $\frac{d}{2}(d-1)+2$ Schritten zu keinem positiven Resultat (d-te Spalte erreicht), so wird

$\underline{0}$ ausgegeben. STOP. Der Bildcomputer benötigt derart für die Realisierung des beschriebenen Operators wpath höchstens $O(d^2)$ Schritte. Der Algorithmus verwendet dabei einen Raum aus $O(d^2)$; vier Matrixregister und ein Vektorregister sind ausreichend. Ein sequentiell arbeitendes System würde bei einer Realisierung des Operators wpath auf Grund der notwendigen Weglängenvergleiche der verschiedenen Pfade für gewisse Binärbilder mindestens $O(d^3)$ Schritte benötigen.

4. Berechenbarkeit und Realisierbarkeit

Das Operationensystem des Bildcomputers ist reichhaltig genug, jede Abbildung $f : \underline{V}^{n_1} \times \underline{B}^{m_1} \longrightarrow \underline{V}^{n_2} \times \underline{B}^{m_2}$ zu berechnen. (Eine prinzipiell andere Fragestellung ist dagegen, nach der Berechenbarkeit bei variabler Bildgröße zu fragen.) Für die besonders interessierende Abbildungsklasse $\underline{F}$ aller Operatoren auf $\underline{B}$ gilt

Satz 6: (Hübler/Klette) Jeder n-stellige Operator, $n \geq 1$, liegt bereits in der Kompliziertheitsklasse $O((2^{n \cdot d^2} \cdot \log d)/n)$ - {non, et, slr i, sll i, slu i, sld i} - $O(n \cdot d^4)$.

Zum Beweis dieses Satzes wurden n-stellige Operatoren durch lineare Programme dargestellt. Als eine Folgerung dieses Beweises ist die Beziehung

$$\underline{F} = [\{non, et, slr, sll, slu, sld\}]$$

herzuleiten. Eine einfache Basis für $\underline{F}$ ist zum Beispiel die Operatorenmenge {non,et,slr,rot}.
Das im Beweis von Satz 6 vorzunehmende vollständige Aufspalten der Binärbildzuordnungen ohne jede Redundanzzusammenfassung führt bereits für kleine Bildgrößen d zu praktisch nicht zu bewältigenden Speicherplatz- und Rechenzeitbedürfnissen. Ein Verzicht auf vollständiges Aufspalten baut auf der Kenntnis über gewisse Redundanzeigenschaften einer Abbildung auf; die Zusammenfassung von Situationen führt zu einem Programm zur Berechnung einer Abbildung, das ein vollständiges Aufspalten überflüssig macht. Je besser eine solche Zusammenfassung gelingt, um so eher kann die betrachtete Abbildung als

r e a l i s i e r b a r angenommen werden. Es wäre bereits von Interesse, die Klasse aller in polynomialer Zeit realisierbaren Operatoren näher zu charakterisieren.

Literatur

/1/ Andrews, H. C., L. H. Enloe (Edts.), Digital Picture Processing. Proc. IEEE, spec. issue, 60 (1972) July.

/2/ Hübler, A., Algorithmen für konkrete Bildtransformationsaufgaben in TSE-RASP. FSU Jena BASM-77/4, Nov. 1977.

/3/ Klette, R., Fast Matrix Multiplication by Boolean RAM in Linear Storage. Proc. MFCS'78, Zakopane, Sept. 1978.

/4/ Klette, R., Anzahlbetrachtungen zu elementaren einstelligen Operatoren auf Binärbildern. FSU Jena BASM-78/10, Aug. 1978 (eingereicht für EIK).

/5/ Lindner, R., Zur TSE-RASP-Realisierung von Klassen homogener Image-Funktionen. FSU Jena BASM-77/2, Okt. 1977.

/6/ Pratt, V. R., L. J. Stockmeyer, A Characterization of the Power of Vector Machines. J. Comput. Syst. Sci. 12 (1976) 198-221.

/7/ Rosenfeld, A., J. S. Weszka, Picture Recognition. Communication and Cybernetics 10 (Edt. K. S. Fu), Springer 1976, 135-166.

/8/ Schaefer, D. H., J. P. Strong, Tse Computers. Goddard Space Flight Center Report X-943-75-14, Jan. 1975.

eingegangen: 15.9.1978

Anschrift des Verfassers:

Prof.Dr.sc. Rolf Lindner
Dr. Reinhard Klette
Dipl.-Math. Albrecht Hübler
Friedrich-Schiller-Universität Jena
Sektion Mathematik
DDR - 69 Jena
Universitätshochhaus 17.OG

Rostock. Math. Kolloq. 10, 63 - 69 (1978)

Klaus P. Jantke

Universal Methods for Identification of Total Recursive Functions

Introduction

The current work is built upon the fundamental results in the theory of inductive inference, developed by Gold, Barzdin and others. The basic idea of this mathematical theory is the well known black box situation: Given a black box with a total recursive function inside. We are allowed to experiment on the black box, that means we are able to put natural numbers x in the black box, the function f inside produces the values $f(x)$ and the black box returns these values. The intention is to guess the unknown function inside in a recursive way.
Let be given an unknown total recursive function f. Provided that we can look step by step at the sequence $f(0),f(1),\ldots$, the intention is to guess this function after some finite time using only the given sequence. As descriptions of recursive functions we use Gödelnumbers with respect to a fixed Gödelnumbering φ.
We give now an exact definition using the following notation: P and R denote the class of all partial recursive respectively total recursive functions. For any total recursive function f and any natural number n $f[n]$ denotes a code of the finite sequence $f(0),\ldots,f(n)$.

Definition 1: A class U of total recursive functions is said to be identifiable in the limit
iff There is a function $F \in P$ such that for all $f \in U$ holds:
(1) For all natural numbers n $F(f[n])$ is defined.
(2) $L(f) := \lim F(f[n])$ exists.
(3) $\varphi_{L(f)} = f$.
GN denotes the set of all function classes U identifiable in

the limit.

A recursive function F in the sense of definition 1 is said to be a strategy for (the identification of) U.

As introduction we want to regard three different approaches to the subject of this paper.

(1) It is a well known general mathematical question, having a set of single solvable problems, to construct an universal algorithm solving all of these problems in the case, that the algorithm "knows", what problem have been given. Our single problems described below are function classes identifiable in the limit.

(2) As an inner question in the theory of inductive inference it is interesting to explain, what kind of partial recursive functions are the mysterious strategies mentoined in the definition 1.

(3) In the theory of automatic program synthesis one approach consists in constructing programs from lists of behavior examples (another approach in the program construction using a system of example computations or computation traces covering a realization of the guessed program). Because it is, in fact, doubtfull that an user has exactly one and only one fixed target program, in real programming problems there is a set of programs suitable as solutions. Therefore following this idea a single programming problem defines a special function class (identifiable in the limit). Because a theory of automatic programming is senseless for only a single problem, the question is to find algorithms solving many those problems. That is the question to identify all function classes of a certain family in GN.

Summarizing the mentoined approaches the subject of our current work is to define and to study functions of two variables as universal identification algorithms in the following sense: If a fixed description of a function class identifiable in the limit is given as the first argument, the resulting function of one variable is a strategy for the described class.

Descriptions of function classes

In almost all publications in inductive inference or automatic synthesis of programs the regarded classes of total recursive functions are assumed to be effectively enumerable. As finite descriptions of those effectively enumerable classes we use Gödelnumbers of total recursive functions enumerating these classes. Therefore we define:

Definition 2: A class U of total recursive functions is said to be effectively enumerable
iff There is a total recursive function h such that
$U \subseteq \{\varphi_{h(i)} / \ i=1,2,\ldots\} \subseteq R$
NUM is the set of all effectively enumerable classes in R.
I_{NUM} is the set of all Gödelnumbers of enumerations h in the definition above.
For all $i \in I_{NUM}$ U_i denotes the class $\{\varphi_{h(i)} / \ i=1,2,\ldots\}$.
It is well known that NUM is a proper subset of GN (see /1/). Therefore we need a generalization of definition 2 to describe function classes identifiable in the limit but not effectively enumerable.

Definition 3: For all $U \in GN$ a number i is said to be a possible description of U
iff $h = \varphi_i \in R$ and $U \subseteq \{\varphi_{h(i)} / \ i=1,2,\ldots\} \cap R$.
For any natural number i U_i denotes the class $\{\varphi_{h(i)} / i=1,..\} \cap R$. $I_{GN} := \{i / \ U_i \in GN\}$.
Firstly we give a trivial result to motivate our later definition of strategic operators.

Theorem 1: There is a partial recursive function F^{Σ} of 2 variables such that for all $i \in I_{NUM}$ $\lambda z F^{\Sigma}(i,z)$ is a strategy for U_i.

Proof: This theorem is trivial, because the well known method of identification by enumeration (see /9/) can be choosen as F^{Σ}. We will define it precisely:

$$F^{*}(i,f[n]) := \begin{cases} \varphi_i(\mu j(\varphi_{\varphi_i(j)}[n]=f[n])) \\ \text{not defined, if there is a number } j \text{ such that for all } j' \leq j \text{ holds } \varphi_{\varphi_i(j')}[n] \neq f[n] \text{ and there is a number } j' \leq j \text{ and } m \leq n, \text{ where } \varphi_{\varphi_i(j')}[m] \text{ not defined.} \end{cases}$$

It is clear that F^{*} is a partial recursive function and the domain of $\lambda z F^{*}(i,z)$ is infinite if and only if $i \in I_{NUM}$. It is quite clear that for any $i \in I_{NUM}$ $\lambda z F^{*}(i,z)$ is a recursion-theoretic formalization of the identification by enumeration in U_i enumerated by φ_i. This completes the proof.

Strategic operators

In the paper /7/ of Podnieks "strategic operators" are defined as special Turingmachines for prediction only in effectively enumerable function classes. We will use this name in a more general case to denote universal recursive identification methods.

Definition 4: Let I be an arbitrary set of natural numbers and F is any partial recursive function. Assume that $I \subseteq I_{GN}$.
F is said to be a strategic operator on I
iff For all $i \in I$ F is a strategy for U_i.

The function F^{*} defined in the proof of theorem 1 is such a strategic operator on I_{NUM}. The next natural question here is, whether there exists a strategic operator on I_{GN} or not. We give an answer:

Theorem 2: There is no strategic operator on the set of all numbers i in I_{GN} with U_i contains exactly one element.

Sketch of proof: For an exact proof see /6/. The undecidability of the halting problem is the key reason that this theorem holds. Let us suppose that any partial recursive function F of 2 variables is presented. Using the fixed point theorem for total recursive functions (see /8/) it is possible to

construct those enumerations of function classes containing exactly one element that F is unable to decide, whether the first enumerated function is total recursive or not. This construction is dependend of F in a recursive way.

As a simple conclusion of our theorem 2 we see, that there is no strategic operator on I_{GN}. Is there at least a strategic operator on a sufficiently large description set containing description of all function classes identifiable in the limit? The next theorem will give a positive answer. Furthermore it can be constructed a strategic operator being a proper extension of the identification by enumeration method.

Theorem 3: There is a description set $I \subseteq I_{GN}$ and a function F of 2 variables such that

(1) I is recursive.
(2) $I \cap I_{NUM} = \emptyset$
(3) For all $U \in GN$ there is an $i \in I$ with $U \subsetneq U_i$.
(4) F is a strategic operator on $I \cup I_{NUM}$.
(5) $F/_{I_{NUM}} = F^{*}$

Sketch of proof: The fundamental steps to prove this theorem are the following: All partial recursive functions can be regarded as identification strategies. There are recursive procedures (see /10/, /11/) to construct for all partial recursive functions F an enumeration of a function class containing all total recursive functions identifiable by F. Furthermore all these enumerations can be extended to enumerate at least one proper partial recursive function. Instead of constructing those enumerations, we are able to compute Gödelnumbers in such a way, that the resulting set of all descriptions obtained from a strategy is recursive. Suppose that I have been constructed. F will be defined step by step as a parallel procedure in the first computation testing whether i is an element of I or not and in the second computation trying to realize the principle of identification by enumeration on φ_i.

In /6/ a stronger conception of strategic operators is formulated. Theorems being analogous to the theerems in our current work are proved there.

Literature

/1/ Barzdin, J. M., Inductive Inference of Automata, Functions and Programs (russ.), Intern. Math. Congress, Vancouver, 1974.

/2/ Gold, E. M., Limiting Recursion, J. Symb. Logic 30, 1965, 28-48.

/3/ Gold, E. M., Language Identification in the Limit, Inf. and Control 10, 1967, 447-474.

/4/ Biermann, A. W., The Inference of Regular LISP Programs from Examples, Duke University, March 1977.

/5/ Biermann, A. W. and Krishnaswamy, R., Constructing Programs from Example Computations, IEEE Trans. on Software Eng., Vol. 2, No. 3, 1976, 141-153.

/6/ Jantke, K. P., Leistungsfähigkeit und Kompliziertheit universeller Verfahren zur Erkennung allg.-rek. Funktionen, Diss. A, Berlin, 1978.

/7/ Podnieks, K. M., Predicting Strategies of Limited Complexity, (russ.). In /14/, 89-102.

/8/ Rogers, H., Theory of Recursive Functions and Effective Computability, McGraw-Hill, New York, 1967.

/9/ Solomonoff, R. J., A Formal Theory of Inductive Inference, Inf. and Control 7, 1964, 1-22 and 224-254.

/10/ Wiehagen, R., Zur Theorie der algorithmischen Erkennung, Diss. B, Berlin 1978.

/11/ Wiehagen, R. and Jung, H., Rekursionstheoretische Charakterisierung von erkennbaren Klassen rekursiver Funktionen, EIK 13, 1977, 385-397.

/12/ Theory of Algorithms and Programs (russ.), University of Riga (USSR), 1974.

/13/ Theory of Algorithms and Programs (russ.), University of Riga (USSR), 1975.

/14/ Theory of Algorithms and Programs (russ.), University of Riga (USSR), 1977.

eingegangen: 15.9.1978

Anschrift des Verfassers:

Dipl.-Math. Klaus P. Jantke
Humboldt-Universität zu Berlin
Sektion Mathematik
Bereich Math. Kybernetik
DDR - 1086 Berlin
PSF 1297

Rostock. Math. Kolloq. 10, 71 - 81 (1978)

Immo O. Kerner

Diskrete Arithmetik

Anwendern von Computern ist bestens bekannt, daß die dabei gültige Arithmetik sich durchaus von der in der Analysis (auch in der Numerischen Mathematik) benutzten und begründeten unterscheidet. Der Grund dafür liegt in der Tatsache, daß man in der Analysis - und allen darauf beruhenden oder davon abgeleiteten Teilen der Mathematik - das Kontinuum der reellen Zahlen verwendet, während man sich auf Computern mit einem endlichen und diskreten Zahlraum, einem Teilraum der rationalen Zahlen begnügen muß. Von algebraischer Sicht kann dieser Raum nicht durch übliche Strukturen (z.B als linear geordneter Körper mit den für reelle Zahlen geltenden Verträglichkeitseigenschaften zwischen der algebraischen Struktur und der Ordnungsstruktur) beschrieben werden. Bereits mit der Addition als Verknüpfung liegt schon keine Halbgruppe vor, da das Assoziativgesetz nicht allgemein erfüllt ist. Man benötigt also weiter entwickelte mathematische Strukturen, um die reale Computerarithmetik algebraisch zu erfassen.

Rasterstruktur und Rundung

Geführt von der Praxis oder den realen Verhältnissen in existierenden Computern geht man von einer Menge R in üblicher Zahldarstellung, meist Gleitkomma und Skalierungsfaktor, aus. Für das kartesische Produkt $R \times R$ als Operandenmenge einer dyadischen Operation liegt kein zweidimensionales Kontinuum vor, sondern ein "Raster", welches jedoch einige Gesetzmäßigkeiten beinhaltet. Das Ausführen einer Operation

1. ist nicht unbeschränkt möglich (Überlauf)
2. liefert durch Rundung i.a. einen Rasterpunkt, der nicht äquivalent ist mit dem im Kontinuum gewonnenen Resultatpunkt.

Es gibt dazu bereits eine zahlreich gewordene Literatur.
Aus ihr entnimmt man /1/, /2/:
Definition eines Rasters:
Es sei $\{M, \leq\}$ eine halbgeordnete Menge, L(a) und U(a) mit $a \in M$ seien die Mengen der unteren (lower) und oberen (upper) Schranken für ein Element a. Eine Teilmenge $R \subseteq M$ heißt "unteres oder oberes Raster" von $\{M, \leq\}$, wenn

$$\forall a(a \in M,\ L(a) \cap R \neq \emptyset)$$
$$\forall a\ \exists x\ \forall b(a \in M,\ x \in L(a) \cap R,\ b \in L(a) \cap R,\ b \leq x)$$

bzw. entsprechend

$$\forall a(a \in M,\ U(a) \cap R \neq \emptyset)$$
$$\forall a\ \exists x\ \forall b(a \in M,\ x \in U(a) \cap R,\ b \in U(a) \cap R,\ x \leq b)$$

erfüllt ist.
D.h., die Mengen $L(a) \cap R$ haben jeweils ein größtes und die Mengen $U(a) \cap R$ haben jeweils ein kleinstes Element. Die erste Forderung macht deutlich, daß es zum Computerzahlraum noch immer Differenzen gibt. Denkt man bei M an das Kontinuum reeller Zahlen und bei R an den Computerzahlenraum, so ist die Forderung nicht für alle a erfüllt. Sie ist allerdings erfüllt, wenn M ein solcher Teilraum des reellen Zahlraumes ist, so daß die minimalen und maximalen Elemente von M und R sich jeweils decken.
Ist R sowohl unteres als auch oberes Raster von M, so heißt es "R ist Raster von M". Der Übergang von M zu R wird durch die Abbildung "Rundung" erreicht.

Definition der Rundung:
Es sei $\{M, \leq\}$ eine (halb-) geordnete Menge und R dazu ein Raster. Die Abbildung $r : M \rightarrow R$ heißt "Rundung von M in R", wenn gilt

$\forall a,b(a,b \in M,\ a \leq b \rightarrow ra \leq rb)$ (monoton).

Dabei sei diese speziell mit

$\forall a(a \in R,\ ra = a)$ (optimal),
$\forall a(a \in M,\ ra \leq a)$ (nach unten gerichtet),
$\forall a(a \in M,\ a \leq ra)$ (nach oben gerichtet).

Mit diesen Hilfsmitteln erkennt man diejenigen Eigenschaften der üblichen Zahlkörper, die beim Übergang zum Raster erhalten bleiben, die also "rundungsinvariant" sind. Axiomatisch werden sie mit der Definition des Ringoids erfaßt.

Definition des Ringoids:
Eine nichtleere Menge R, in der zwei dyadische Operationen + und × erklärt sind, heißt "Ringoid $\{R, +, \times\}$", wenn die Axiome 1 bis 6 gelten

1. $\forall a,b(a,b \in R,\ a+b = b+a)$,
2. $\exists o\, \forall a(o,a \in R,\ a+o = a)$,
3. $\exists e\, \forall a(e,a \in R,\ e \neq o,\ a \times e = e \times a = a)$,
4. $\forall a(a \in R,\ a \times o = o \times a = o)$,
5. $\exists x\, \forall a,b(x,a,b \in R,\ x \neq e,\ x \times x = e,$
   $x \times (a \times b)=(x \times a) \times b = a \times (x \times b)$,
   $x \times (a + b)= x \times a + x \times b)$,
6. Es gibt genau ein solches Element x.

Man erkennt die Forderung der Kommutativität für die Addition, das Nullelement o, das Einselement e und die "negative Eins" x. Lediglich für dieses wird die Assoziativität der Multiplikation und die Distributivität gefordert.
Ein Ringoid ist "geordnet", wenn es in R eine Ordnungsrelation ($\leq$) gab und die Verträglichkeiten 1 bis 3 erfüllt sind

1. $\forall a,b,c(a,b,c \in R,\ a \leq b \longrightarrow a + c \leq b + c)$,
2. $\forall a,b,c(a,b,c \in R,\ o \leq a \leq b \wedge o \leq c$
   $\longrightarrow a \times c \leq b \times c \wedge c \times a \leq c \times b)$,
3. Für x gilt außer (5) auch noch
   $\forall a,b(a,b \in R,\ a \leq b \longrightarrow x \times b \leq x \times a)$.

In dieser Richtung wird von der Algebra in den letzten Jahren zur Unterstützung der Entwicklung einer Strukturtheorie für die Computerarithmetik gearbeitet. Es gelang,damit viele Erscheinungen des diskreten numerischen Rechnens zu erfassen und erklären. Einige bekannte und einige vielleicht auch den Fachmann überraschende Effekte der numerischen diskreten Arithmetik seien hier mitgeteilt.

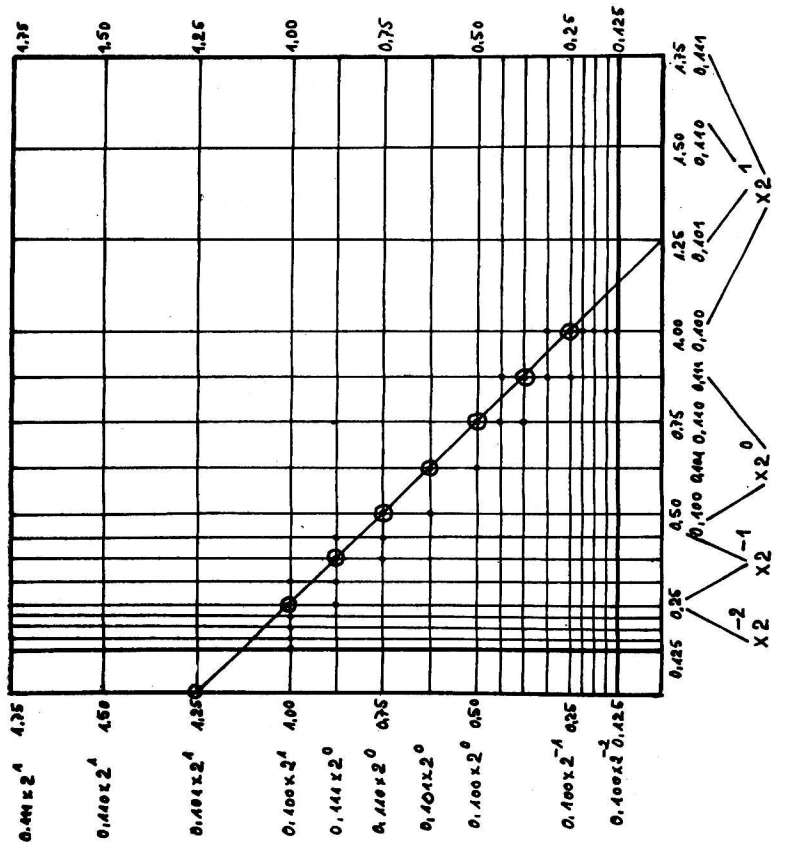


Abb. 1: Diskreter dualer Zahlenbereich

$z_1 \in \{0,1\}$ $n = 3$ $-2 \leq e \leq 1$

a + b = 1,25 im Kontinuum
a + b = 1,25 diskret, exakt
a + b = 1,25 durch Rundung

In der Abb. 1 liegt ein Zahlenraster R für dreistellige Dualzahlen mit dem Exponentenbereich $-2 \leq e \leq 1$ zugrunde, d.h. Gleitkommadarstellung. Das Bild zeigt die Menge $R \times R$ als Operandenmenge für die Addition. In einem solchen Zahlenraster liegen anstelle der überabzählbar vielen Summen vom Wert 1,25 aus dem Kontinuum (alle auf einer Geraden) nur 9 Summen dieses Wertes auf der Geraden und 22 weitere solche Summen durch Rundung entstehen (in diesem Beispiel wird nur viermal abgerundet, trotz symmetrischer Rundungsvorschrift bzgl. des Kontinuums).

Ein konkretes Beispiel für die Verletzung des Assoziativgesetzes der Addition ist gegeben durch

$$(a + b) + c = a + (b + c)$$

mit $a = 0{,}100 \cdot 2^{1} = 1{,}000$

$b = 0{,}100 \cdot 2^{-2} = 0{,}125$

$c = 0{,}100 \cdot 2^{-2} = 0{,}125$, da im Raster der Abbildung (dual)

$0{,}100 \cdot 2^{1} = 1{,}5$ (dezimal) $\neq 0{,}101 \cdot 2^{1} = 1{,}25$ (dezimal)

entsteht.

Auch mathematische Aussagen über numerisches Rechnen, die auf der Grundlage des Kontinuums getroffen wurden, verlieren ihre Gültigkeit beim Rechnen auf einem Raster.

Die Reihenentwicklung für e^{x}

$$1 + \frac{x}{1!} + \frac{x^2}{2!} + \ldots$$

ist bekanntlich beständig konvergent. Für negative x wird daraus eine alternierende Reihe und der Abbruchfehler beim numerischen Auswerten einer konvergenten alternierenden Reihe ist kleiner als der Betrag des ersten vernachlässigten Gliedes (Knopp "Unendliche Reihen"). Die praktische Anwendung dieses bekannten Satzes liefert bei 10-stelligem Rechnen mit der Genauigkeitsschranke 10^{-8} aber

x	alternierende Reihe	e^x exakt	$\lvert\Delta\rvert$	Anzahl der Glieder
- 1	$3{,}6787944 \times 10^{-1}$	$3{,}6787944 \times 10^{-1}$	10^{-8}	12
- 10	$4{,}4802480 \times 10^{-5}$	$4{,}5399929 \times 10^{-5}$	6×10^{-7}	41
- 11	$1{,}7772109 \times 10^{-5}$	$1{,}6701700 \times 10^{-5}$	1×10^{-6}	43
- 12	$3{,}6000740 \times 10^{-6}$	$6{,}1442123 \times 10^{-6}$	3×10^{-6}	46
- 13	$4{,}7107442 \times 10^{-6}$	$2{,}2603294 \times 10^{-6}$	2×10^{-6}	49
- 20	$2{,}1825293 \times 10^{-3}$	$2{,}0611536 \times 10^{-9}$	2×10^{-3}	69
- 50	$1{,}1589190 \times 10^{11}$	$1{,}9287498 \times 10^{-22}$	2×10^{11}	151

Zwischen -4 und -5 wird erstmalig die mathematische Aussage über das Fehlerverhalten nicht mehr bestätigt. Bei -12 ist keine Ziffer des Resultates mehr richtig. Bei -13 steigen die Werte der alternierenden Reihe sogar wieder an und bei -50 wird ein Ergebnis geliefert, das um 33 Größenordnungen falsch ist.

Die algebraisch leicht prüfbaren Identitäten

$$99 - 70\sqrt{2} = \frac{1}{99 + 70\sqrt{2}} = \frac{1}{(1 + \sqrt{2})^6}$$

liefern bei 10-stelliger Rechnung

0,005050660000 0,005050633885 0,005050633890

Wird 5-stellig gerechnet, entsteht

0,0060000 0,0050507 0,0050508

und bei 3-stelliger Rechnung sogar

0,300 0,00506 0,00506

wobei der erste Wert um zwei Größenordnungen falsch wird und gegenüber dem zweiten und dritten den 30 000-fachen Fehler zeigt.

Frage der Akzeptierbarkeit von Ergebnissen in der Numerischen Mathematik:

Gegeben sei ein Gleichungssystem

$$2{,}2969\, x_1 + 0{,}8648\, x_2 = 0{,}8642$$
$$0{,}2161\, x_1 + 0{,}1441\, x_2 = 0{,}1440$$

Seine formale Lösung ist (bei 10-stelliger Rechnung)

$$x_1 = 2 \qquad x_2 = -2$$

und diese ist auch die exakte Lösung im Kontinuum. Wendet man ein iteratives Verfahren an, so erhält man beim Defekt

$$Ax^{(k)} - b = \begin{pmatrix} -10^{-8} \\ 10^{-8} \end{pmatrix}$$

die Näherungslösung

$$x^{(k)} = \begin{pmatrix} 0{,}9911 \\ -0{,}4870 \end{pmatrix} .$$

Es muß angemerkt werden, daß das System mit der Determinante 10^{-8} schlecht konditioniert ist. Dieses Lösungsangebot ist "weit" von der exakten Lösung entfernt.
Die viel "näher" am exakten Wert liegenden

$$\bar{x}_1 = 2{,}01$$

$$\bar{x}_2 = -1{,}99$$

liefern den relativ "großen" Defekt

$$A\bar{x} - b = \begin{pmatrix} 0{,}022 \\ 0{,}004 \end{pmatrix}$$

und sind zu verwerfen. Dies stellt man heute auch ohne Kenntnis des exakten Wertes nach einem Kriterium von Prager-Öttli fest (1964).

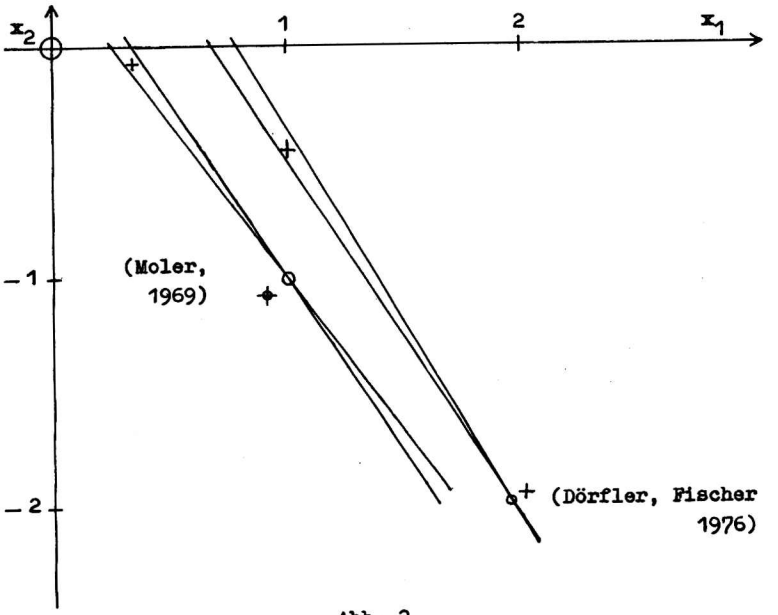


Abb. 2

Die exakte Lösung des Systems ist durch (o) markiert. Eine akzeptierbare Näherungslösung ist (+), da diese im Rundungsbereich liegt.
Der Punkt (⊕) ist als Lösungsangebot zu verwerfen.
Wegen des Rundungsfehlers von 10^{-4} ist der Fehlerstreifen stark vergrößert dargestellt.

Fragt man bei der Auflösung eines linearen Gleichungssystems $Ax = b$ (A nichtsingulär) danach, wie gut eine berechnete Näherungslösung $\bar{x}$ ist, so stützt man sich gewöhnlich auf das Residuum: $r = b - A\bar{x}$, man könnte aber, wenn die exakte Lösung x bekannt ist, auch von dem Fehler $e = x - \bar{x}$ ausgehen.

Daß die beiden Forderungen "kleines Residuum" oder "kleiner Fehler" nicht gleichbedeutend sind, hat C. B. Moler (1969) an einem schönen Beispiel gezeigt:

$$A = \begin{pmatrix} 0{,}780 & 0{,}563 \\ 0{,}913 & 0{,}659 \end{pmatrix}, \quad b = \begin{pmatrix} 0{,}217 \\ 0{,}254 \end{pmatrix}.$$

Exakte Lösung: $x = \begin{pmatrix} 1 \\ -1 \end{pmatrix}$, $r = b - Ax = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$.

Kleines Residuum, aber großer Fehler:

$$\bar{x}_1 = \begin{pmatrix} 0{,}341 \\ -0{,}087 \end{pmatrix}, \quad r = b - A\bar{x}_1 = \begin{pmatrix} 0{,}000001 \\ 0 \end{pmatrix}.$$

Kleiner Fehler, aber relativ großes Residuum:

$$\bar{x}_2 = \begin{pmatrix} 0{,}999 \\ -1{,}001 \end{pmatrix}, \quad r = b - A\bar{x}_2 = \begin{pmatrix} 0{,}001243 \\ 0{,}001572 \end{pmatrix}.$$

Diese Unterschiede liegen natürlich wieder an der schlechten Kondition von A, denn es ist

$$A^{-1} = \begin{pmatrix} 659000 & -563000 \\ -913000 & 780000 \end{pmatrix}, \quad \det(A) = 10^{-6},$$

und solche Beispiele lassen sich leicht in großer Zahl angeben. Soweit also die Verhältnisse im Kontinuum, wie sie die numerische Mathematik mit Hilfe der Funktionalanalysis untersucht.

Die Gitter- oder Rasterpunkte in einem praktisch, d.h. im Computer, verwendeten Raster liegen keineswegs willkürlich, sondern folgen einer Gesetzmäßigkeit. Diese sorgt zwar für einen gewissen Zusammenhang mit Rechenresultaten im Kontinuum aber wiederum auch für Abweichungen. Eine solche auch den Fachmann überraschende Abweichung zeigt das folgende Beispiel (Klatte, Ulrich 1977).

Die Vektoriteration

$$x_{n+1} := \begin{pmatrix} 0{,}11 & 0{,}87 \\ 0{,}96 & 0{,}03 \end{pmatrix} x_n + \begin{pmatrix} 3{,}4 \\ -3{,}6 \end{pmatrix}$$

hat im Kontinuum genau einen Fixpunkt, wie sich mit Hilfsmitteln der Funktionalanalysis nachweisen läßt. Mit 10-stelliger Rechnung erhält man zunächst die Lösung des äquivalenten Gleichungssystems

$$0{,}89\ x^{(1)} - 0{,}87\ x^{(2)} = 3{,}4$$
$$-0{,}96\ x^{(1)} + 0{,}97\ x^{(2)} = -3{,}6$$

mit

$$x_0 = \begin{pmatrix} 5{,}907473311 \\ 2{,}135231317 \end{pmatrix}$$

und noch fünfmaliges Iterieren ist nötig, um den Fixpunkt (bei 10-stelliger Rechnung) zu erhalten.

$$x = \begin{pmatrix} 5{,}907473312 \\ 2{,}135231320 \end{pmatrix}.$$

Rechnet man jedoch in einem groben zweistelligen Raster, so gibt es 11 (!) Fixpunkte und dazu 17 Rechenzyklen aus zwei Werten. In der Abbildung 3 sind die Fixpunkte markiert und keiner stimmt mit dem eben angegebenen überein.
Die diskrete Rasterarithmetik theoretisch, d.h. algebraisch, zu untersuchen, ist eine Aufgabe von möglicherweise größerer praktischer Tragweite. Außerdem sind größere experimentelle arithmetische Arbeiten nötig, um "Material" zu bekommen.

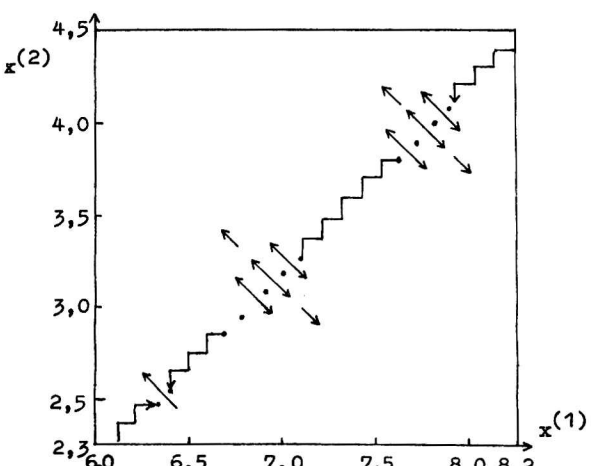


Abb. 3

In einem Raster der Genauigkeit 10^{-2} hat der Iterationsprozeß (siehe Text) 11 Fixpunkte und 17 Zweierzyklen, die aber nicht alle eingezeichnet sind.

Literatur

/1/ Kerner, I. O., Numerische Mathematik in der Lehrerausbildung und im Schulunterricht, Wiss. Zeitschrift der Päd. Hochsch. Dresden, Heft /1978 (im Druck)

/2/ Kerner, I. O., Numerische Mathematik und Rechentechnik Bd. 1, Teubner, Leipzig, 1970.

/3/ Klatte, R., Zyklisches Enden bei Iterationsverfahren Diss. Universität Karlsruhe, 1975.

/4/ Klatte, R., Ullrich, Chr., Zur mathematischen Struktur von Rechnerarithmetiken, MNU 30 (1977) 1, 1-8.

/5/ Knopp, K., Unendliche Reihen, Berlin, 1931.

/6/ Prager, W., Öttli, W., Computability of approximate solution of linear equations with given error bounds for coeffizients and right hand sides, Num. Math. 6, (1964) 405-409.

/7/ Ullrich, Chr., Rundungsinvariante Strukturen mit äußeren Verknüpfungen, Diss. Universität Karlsruhe, 1972.

/8/ Moler, C. B., (1969; Privatmitteilung von G. Zielke/Halle)

eingegangen: 15.9.1978

Anschrift des Verfassers:

Prof.Dr.sc.nat. I. O. Kerner
Pädagogische Hochschule Dresden
Sektion Mathematik/Geographie
Wissenschaftsbereich Numerische Mathematik

Rostock. Math. Kolloq. 10, 83 - 86 (1978)

Werner Nehrlich

Eine Bemerkung zur effektiven Fixpunktberechnung bei vollständigen Numerierungen

In der Theorie der Numerierungen, wie sie etwa in /2/ entwickelt wird, spielen die auf Malzew zurückgehenden vollständigen bzw. fastvollständigen Numerierungen eine zentrale Rolle. Bekannte und wichtige Beispiele solcher Numerierungen stellen die Gödelisierungen als spezielle effektive Numerierungen der Menge der partiell rekursiven Funktionen bzw. die Postschen Numerierungen der Menge aller rekursiv aufzählbaren Mengen dar. Vollständige und fastvollständige Numerierungen besitzen besonders interessante algorithmentheoretische Eigenschaften, wie die Fixpunkteigenschaft und die Ricesche Unentscheidbarkeitseigenschaft und die Inseperabilitätseigenschaft.

In der vorliegenden Note wird die Fixpunkteigenschaft betrachtet. Ersov zeigt in /2/, daß diese Eigenschaft die Menge der fastvollständigen Numerierungen charakterisiert. Dabei wird der Fixpunkt einer vorgegebenen allgemein rekursiven Funktion, ausgehend von einer Gödelnummer dieser Funktion, effektiv berechnet (siehe Definition 4). Wir untersuchen die Frage, wann eine solche Fixpunkt berechnende allgemein rekursive Funktion sogar eineindeutig gewählt werden kann.

Unter einer Numerierung ν einer Menge S verstehen wir eine eindeutige Abbildung der Menge $Nz = \{0,1,2,\ldots\}$ der natürlichen Zahlen auf S.

Im folgenden sei mit $\{\varphi_n\}_{n \in Nz}$ eine fest gewählte Gödelisierung der Menge der partiell rekursiven Funktionen gegeben.

Definition 1: (Malzew, /1/) Eine Numerierung ν einer Menge S heißt vollständig $\leftrightarrow_{Df}$. Es existiert ein Element $a \in S$ (das sogenannte "ausgezeichnete" Element), so daß zu jeder partiell rekursiven Funktion g eine allgemein rekursive Funktion f

existiert, welche die Beziehung

$$\nu(f(n)) = \begin{cases} \nu(g(n)) & \text{, falls } g(n) \text{ definiert} \\ a & \text{, sonst} \end{cases}$$

erfüllt.

Jede Gödelisierung ist vollständig, ausgezeichnetes Element ist die nirgends definierte Funktion /1/.

Definition 2: (siehe /2/) Eine Numerierung ν einer Menge S heißt fastvollständig $\longleftrightarrow_{Df}$. Zu jeder partiell rekursiven Funktion g existiert eine allgemein rekursive Funktion f, so daß $\nu(g(n)) = \nu(f(n))$ immer dann gilt, wenn g definiert ist.

Man kann leicht zeigen:
(1) Jede vollständige Numerierung ist fastvollständig.
Die Umkehrung gilt jedoch nicht (siehe /2/).

Definition 3: Eine Numerierung ν einer Menge S erfüllt die Fixpunkteigenschaft (FPE) $\longleftrightarrow_{Df}$.
Zu jeder allgemein rekursiven Funktion h existiert ein $n \in Nz$, so daß $\nu(h(n)) = \nu(n)$ gilt. Die Zahl n nennen wir ν-Fixpunkt der allgemein rekursiven Funktion h.

Bei den wichtigen Beispielen von Numerierungen, die die FPE erfüllen, kann der Fixpunkt der vorgegebenen allgemein rekursiven Funktion, ausgehend von einer Gödelnummer dieser Funktion, effektiv berechnet werden.

Definition 4: Eine Numerierung ν einer Menge S besitzt die effektive Fixpunkteigenschaft (EFPE) $\longleftrightarrow_{Df}$. Es existiert eine allgemein rekursive Funktion f derart, daß gilt:
Ist n eine φ-Nummer einer allgemein rekursiven Funktion, so gilt

$$\nu(f(n)) = \nu(\varphi_n(f(n))) \ .$$

Die Funktion f berechnet also einen ν-Fixpunkt der allgemein rekursiven Funktion φ_n.
Goetze zeigt in /4/, daß es Numerierungen gibt, die zwar die FPE, aber nicht die EFPE besitzen.
Eine Charakterisierung der Numerierungen mit FPE ist nicht bekannt. Für die Numerierung mit EFPE zeigt Ersov in /2/:

Satz 1: Eine Numerierung ν ist genau dann fastvollständig, wenn sie die EFPE besitzt.

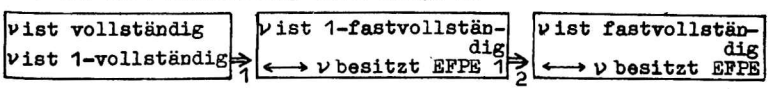
Malzew untersuchte bereits 1961 fastvollständige Numerierungen und zeigte (1).
Die Begriffe der Vollständigkeit, Fastvollständigkeit und der EFPE können nun modifiziert werden, indem wir in den Definitionen 1, 2 und 4 jeweils fordern, daß die dort auftauchende allgemein rekursive Funktion f sogar eineindeutig ist. Wir wollen in diesem Falle von 1- Vollständigkeit, 1- Fastvollständigkeit und EFPE 1 sprechen.
Man kann zeigen, daß jede Gödelisierung die EFPE 1 besitzt (siehe z.B. /3/). Ein tiefliegendes Resultat von Ersov (/2/, S. 331 ff.) besagt, daß die Begriffe Vollständigkeit und 1-Vollständigkeit identisch sind.
Hieraus ergibt sich leicht als Verschärfung von (1), daß jede vollständige Numerierung sogar 1- fastvollständig ist.

Satz 2: Eine Numerierung ν ist genau dann 1- fastvollständig, wenn sie die EFPE 1 besitzt.

Der Beweis von Satz 2 kann hier nicht ausgeführt werden.
Es sei bemerkt, daß alle oben definierten Begriffe für den etwas allgemeineren Fall modifiziert werden können, daß man unter einer Numerierung eine partielle eindeutige Abbildung aus der Menge Nz auf die numerierte Menge versteht (vgl. /1/). Für solche Numerierungen können z.B. die Begriffe Fastvollständigkeit und 1- Fastvollständigkeit sowie Vollständigkeit und 1- Vollständigkeit mit Hilfe spezieller Reduzierbarkeitsbegriffe für (partielle) Numerierungen charakterisiert werden. Der Beweis von Satz 2 verwendet wesentlich diesen Begriffsapparat. Eine ausführliche Darstellung ist in /5/ enthalten.

Folgerung: Jede vollständige Numerierung besitzt die EFPE 1.

Es besteht folgende Implikationskette:
ν sei eine Numerierung einer Menge S.



Wegen (1) ist mindestens eine der Implikationen 1 und 2 nicht umkehrbar. Zur Klärung dieser offenen Probleme wäre es nützlich, genaue Kenntnis darüber zu erlangen, welche der in /2/ aufgeführten "Erhaltungssätze" von den 1- fastvollständigen Numerierungen erfüllt werden. Man kann zeigen, daß die Erhaltungssätze für direktes Produkt und Faktorisierung (welche gleichermaßen für vollständige und fastvollständige Numerierungen gelten /2/) auch von den 1- fastvollständigen Numerierungen erfüllt werden. Aus /2/ ist bekannt, daß n- Teilobjekte fastvollständiger Numerierungen wieder fastvollständig sind, während die vollständigen Numerierungen einen solchen Erhaltungssatz nicht erfüllen. Aus der Beantwortung dieser Fragestellung für 1- fastvollständige Numerierungen erhielte man daher eine genauere Kenntnis der obigen Implikationskette.

Ich danke Herrn Dr. B. Goetze (Jena) herzlich für eine Reihe wertvoller Anregungen.

Literatur

/1/ Малцев, А.И., Полно нумерованные множества, Алгебра и Логика II, вып. 2, 4-29 (1963).

/2/ Ersov, Ju. L., Theorie der Numerierungen I, ZML, 19, Heft 4, (1973).

/3/ Schnorr, C. P., Rekursive Funktionen und ihre Komplexität, Teubner Verlag, Stuttgart (1974).

/4/ Goetze, B. G., Effektive Numerierungen und ihre algorithmischen Eigenschaften, Dissertation, FSU Jena (1976).

/5/ Nehrlich, W., Zur effektiven Berechnung des Fixpunktes bei vollständigen Numerierungen, Manuskript, Jena (1978).

eingegangen: 15. 09. 1978

Anschrift des Verfassers:

Dipl. Math. Werner Nehrlich
Friedrich-Schiller-Universität Jena
Sektion Mathematik
DDR - 69 Jena, Universitätshochhaus

Rostock. Math. Kolloq. 10, 87 - 96 (1978)

Fred Sobik

Kombinatorische Aspekte der Codierungstheorie

Im vorliegenden Beitrag soll auf eine Reihe von Zusammenhängen zwischen der algebraischen Codierungstheorie (der Theorie "fehlerkorrigierender" Codes) und der Kombinatorik hingewiesen werden. Dabei wollen wir auf die folgenden drei Problemkreise näher eingehen:

- Äquivalenz der Existenz von Codes mit bestimmten Parametern und von bestimmten Konfigurationen,
- in Codes enthaltene Konfigurationen und
- aus Konfigurationen abgeleitete Codes.

Da nur eine Übersicht über in der Literatur vorliegende Resultate gegeben werden soll, wird auf die Angabe von Beweisen verzichtet.

Das Übertragungsproblem und e-Fehler-Codes

Bei der Übertragung von Nachrichten über reale Kanäle treten Störungen auf, die ein Wiedererkennen der gesendeten Nachricht zumindest erschweren. Bei diskreten Kanälen können z.B. Asynchronitäten der Übertragung, Auslöschung von übertragenen Zeichen oder Verfälschungen der übertragenen Zeichen auftreten. Wir wollen hier nur Fehler betrachten, die darin bestehen, daß in einer gesendeten Symbolfolge eine Reihe von Symbolen in andere Symbole übergehen, die also die gesendete Symbolfolge verfälschen. Sei nun ein Kanal mit einer Menge X von Ein- und Ausgabesymbolen gegeben. Um eine gesendete Nachricht aus der am Kanalausgang empfangenen Symbolfolge richtig wiedererkennen zu können, werden die Nachrichten in "geeignete" Symbolfolgen über X codiert und diese Codewörter übertragen. Wir werden uns hier auf den Fall von Codewörtern gleicher Länge n, also auf den Fall der Blockcodes beschränken. Damit sind Codewörter hier n-Tupel über X und ein Code ist eine beliebige

Teilmenge von X^n. Werden bei der Übertragung in dem gesendeten Wort u i Stellen verändert, dann hat das empfangene Wort v von u einen um so größeren "Abstand", je größer die Anzahl i der veränderten Stellen ist. Wählt man diese Zahl i als Maßzahl für die Entfernung von u und v, so erhält man die Hamming-Metrik:

Definition: Für alle $u = (u_1,\ldots,u_n)$, $v = (v_1,\ldots,v_n) \in X^n$ sei $d(u,v) = |\{i \mid u_i \neq v_i,\ i = 1,\ldots,n\}|$.

Der Kanal hat also bei der Übertragung genau d(u,v) Fehler gemacht. Obwohl in der Codierungstheorie auch andere Distanzmaße untersucht werden, werden wir uns hier nur mit der Hamming-Metrik befassen. Sei nun $X = \{0, 1,\ldots, q-1\}$. Wir können dann jede Veränderung eines gesendeten Symbols durch die Addition modulo q eines entsprechenden Fehlersymbols beschreiben. Weiterhin können wir das Hamming-Gewicht einführen:

Definition: Für alle $u = (u_1,\ldots,u_n) \in X^n$ sei $w(u) = |\{i \mid u_i \neq 0,\ i = 1,\ldots,n\}|$.

Offensichtlich gilt stets d(u,v) = w(u-v). Wenn wir davon ausgehen, daß der Kanal in n Takten nicht mehr als e Stellen stört, dann ist klar, daß es genügt, die Codewörter so zu wählen, daß es für je zwei Codewörter in X^n kein Wort gibt, das von diesen beiden Codewörtern jeweils einen nicht größeren Abstand als e hat. Alle Codewörter müssen dann mindestens den Abstand 2e +1 voneinander haben. Im weiteren wollen wir solche Codes betrachten.

Definition: Für alle $u \in X^n$ und alle $e \in \{0, 1,\ldots,n\}$ sei $K(u,e) = \{v \mid v \in X^n,\ d(u,v) \leq e\}$.
Ein Code $C \subseteq X^n$ heißt e-Fehler-Code g dw.
$K(u,e) \cap K(v,e) = \emptyset$ für alle $u, v \in C,\ u \neq v$.

Einen e-Fehler-Code $C \subseteq X^n$ mit $|C| = N$ und $d = \min\{d(u,v) \mid u, v \in C,\ u \neq v\}$ bezeichnen wir auch als (n,N,d;q)-e-Fehler-Code.
Eine der grundlegenden Fragestellungen der Codierungstheorie ist die nach der maximal möglichen Zahl N von Codewörtern in einem e-Fehler-Code mit gegebener Codewortlänge n. Wir werden jetzt verschiedene Schranken für Codeparameter betrachten und

die Frage nach der Existenz von Codes stellen, deren Parameter diese Schranken erreichen.

Volumenschranke und perfekte Codes

Da die Anzahl $V_e = |K(u,e)| = \sum_{i=0}^{e} \binom{n}{i}(q-1)^i$ nicht von $u \in X^n$ abhängt und da in einem e-Fehler-Code die Mengen K(u,e) für alle Codewörter u paarweise disjunkt sein müssen, ergibt sich sofort die Volumenschranke (Hamming-Schranke):

Satz: Für jeden (n,N,d;q)-e-Fehler-Code gilt

$$N \leq q^n/V_e = q^n/\left(\sum_{i=0}^{e} \binom{n}{i}(q-1)^i\right).$$

(n,N,d;q)-e-Fehler-Codes, deren Parameter diese Schranke erreichen, werden als perfekte Codes bezeichnet.
Sei S eine Menge mit $|S| = v$ und sei $\mathfrak{B}$ ein System von Teilmengen der Mächtigkeit k von S.

Definition: $(S,\mathfrak{B})$ heißt t-(v,k,λ)-Blockplan (oder kurz t-Blockplan) gdw. jede Teilmenge der Mächtigkeit t von S in genau λ Elementen von $\mathfrak{B}$ enthalten ist.

Einen 2-Blockplan bezeichnen wir auch als Blockplan. Im Fall $\lambda = 1$ sprechen wir von einem Steiner-System und schreiben dafür S(t,k,v).
Wie der folgende Satz zeigt, bestimmen Teilmengen von Codewörtern perfekter Codes Steiner-Systeme.

Satz: Sei C ein perfekter (n,N,d;2)-e-Fehler-Code und sei e ungerade. Sei C' der aus C durch hinzufügen einer Paritätsteststelle erhaltene erweiterte Code. Dann bestimmen die Codewörter $u \in C$ mit dem Gewicht $w(u) = 2e + 1$ ein Steiner-System S(e+1,2e+1,n) und die Codewörter $v \in C'$ mit dem Gewicht $w(v) = 2e + 2$ bestimmen ein Steiner-System S(e+2,2e+2,n+1).

Codes, die die Voraussetzungen dieses Satzes erfüllen sind die $(2^m-1, 2^{2^m-1-m}, 3; 2)$-Hamming-Codes ($m = 2, 3, \ldots$) und der binäre $(23, 2^{12}, 7; 2)$-Golay-Code. Mit Hilfe dieser Codes können

also Steiner-Systeme $S(2,3,2^m-1)$ und $S(3,4,2^m)$ $(m = 2, 3,\ldots)$ bzw. $S(4,7,23)$ und $S(5,8,24)$ bestimmt werden.
Es ist bekannt, daß perfekte 1-Fehler-Codes, perfekte (triviale) $(n,1,d;q)$-n-Fehler- und $(2e+1,2,2e+1;2)$-e-Fehler-Codes sowie der perfekte binäre $(23,N,7;2)$-3-Fehler-Golay-Code und der perfekte ternäre $(11,N,5;3)$-2-Fehler-Golay-Code existieren. Es konnte gezeigt werden (/13/, /14/):

Satz: Im Fall, daß q Primzahlpotenz ist, existieren außer in den oben angegebenen Fällen keine weiteren perfekten $(n,N,d;q)$-e-Fehler-Codes.

Im Fall, daß q keine Primzahlpotenz ist, gilt (/15/):

Satz: Es existieren keine nichttrivialen perfekten $(n,N,d;q)$-e-Fehler-Codes für $e \geq 3$ und $q = p_1^r p_2^s$, wobei p_1 und p_2 verschiedene Primzahlen und r und s positive ganze Zahlen sind.

Für beliebiges q gilt dann noch (/11/ bzw. /1/):

Satz: Es existieren keine nichttrivialen perfekten $(n,N,d;q)$-3-Fehler-Codes für $q > 2$.

Satz: Für jedes $e \geq 3$ und für jedes $q > 2$ gibt es höchstens endlich viele perfekte $(n,N,d;q)$-e-Fehler-Codes.

Damit ist klar, daß für $e \geq 3$ mit Hilfe von perfekten e-Fehler-Codes nicht wie im Fall $e = 1$ unendliche Familien von Steiner-Systemen bestimmt werden können.

Die Johnson-Schranke und fast perfekte Codes

Da sich gezeigt hat, daß kaum perfekte Codes existieren, gewinnen andere Schranken und die durch sie fixierten Codeklassen an Interesse. So konnte Johnson /7/ folgenden Satz beweisen:

Satz: Für jeden $(n,N,d;2)$-e-Fehler-Code gilt

$$N \leq 2^n / \left(\sum_{i=0}^{e} \binom{n}{i} + \frac{1}{\lfloor n/(e+1) \rfloor} \binom{n}{e} \left(\frac{n-e}{e+1} - \left[\frac{n-e}{e+1} \right] \right) \right).$$

Codes, für deren Parameter hier die Gleichheit gilt, werden als fast perfekte Codes bezeichnet.

Definition: Sei $C \subseteq X^n$ ein beliebiger Code und sei $u \in X^n$. Dann sei $\varrho(u,C) = \min \{d(u,v) \mid v \in C\}$.

Auch bei den fast perfekten Codes bestimmen Teilmengen der Codewörter eine Reihe von t-Blockplänen.

Definition: Sei C ein $(n,N,d;q)$-e-Fehler-Code mit $d > 1$. Ein punktierter Code von C besteht aus den Wörtern, die man erhält, wenn man in allen Codewörtern von C eine (festgelegte) Stelle streicht.

Goethals, Snover /5/ zeigten:

Satz: Sei C ein fast perfekter $(n,N,d;2)$-e-Fehler-Code.
1. Die Codewörter $v \in C$, die von einem gegebenen Codewort u den Abstand $d(u,v) = 2e + 1$ haben, bestimmen einen $e-(n,2e+1,\lambda)$-Blockplan mit $\lambda = [(n-e)/(e+1)]$.
2. Die Wörter $v \in X^n$, die von einem gegebenen Codewort u den Abstand $d(u,v) = 2e + 2$ haben, bestimmen einen $e-(n,2e+2,\lambda)$-Blockplan mit $\lambda = [(n-e)/(e+1)]\ [(n-2e-1)/(e+2)]$.
3. Die Wörter $v \in X^n$ mit $\varrho(v,C) = e + 1$, die von einem gegebenen Codewort u den Abstand $d(u,v) = e + 1$ haben, bestimmen einen $e-(n,e+1,\lambda)$-Blockplan mit $\lambda = (n-e) - (e+1)\,[(n-e)/(e+1)]$.

Satz: Wenn der punktierte Code C eines $(n+1,N,d;2)$-Codes C' ein fast perfekter e-Fehler-Code ist, dann bestimmen die Wörter $v \in X^{n+1}$, die von einem gegebenen Codewort $u \in C'$ den Abstand $d(u,v) = 2e + 2$ haben, einen $(e+1)-(n+1,2e+2,\lambda)$-Blockplan mit $\lambda = [(n-e)/(e+1)]$.

$[x]$ bezeichne stets den ganzen Teil der Zahl x.
In /5/ und /12/ konnte gezeigt werden:

Satz: Für $e = 1, 2, 3, 4$ existieren fast perfekte $(n,N,d;2)$-e-Fehler-Codes, die nicht perfekt sind, nur mit den Parametern

$e = 1$, $n = 2^r-2$, $N = 2^{n-r}$ und $r \geq 3$ und
$e = 2$, $n = 4^m-1$, $N = 2^{n-r}$ mit $r = 4m-1$ und $m \geq 2$.

Weiter konnte in /8/ und /17/ bewiesen werden:

Satz: Für $e \geq 5$ existieren keine fast perfekten e-Fehler-Codes.

Damit kann man mit Hilfe fast perfekter e-Fehler-Codes nur folgende t-Blockpläne erhalten: $2-(2^r-1,4,2^{r-1}-2)$-Blockpläne mit $r \geq 3$ ($e = 1$) und $2-(4^m-1,5,(4^m-4)/3)$-, $2-(4^m-1,6,(4^{m-1}-1)(4^m-6)/3)$-, $2-(4^m-1,3,1)$- (Steiner-Tripel-Systeme) und $3-(4^m,6,(4^m-4)/3)$-Blockpläne mit $r = 4m-1$ und $m \geq 2$. Der Fall $t = 1$ blieb hierbei unberücksichtigt.

Hadamard-Matrizen und maximale Codes

Plotkin /10/ konnte die folgende Schranke für Codeparameter bestimmen:

Satz: Für jeden $(n,N,d;2)$-Code gilt

$$N \leq \begin{cases} 2d/(2d-n), & \text{falls } 2d > n, \\ 4t, & \text{falls } n = 4t-2 \text{ und } d = 2t-1 \text{ oder} \\ & \text{falls } n = 4t-1 \text{ und } d = 2t, \\ 8t, & \text{falls } n = 4t \text{ und } d = 2t. \end{cases}$$

Codes, für deren Parameter hier die Gleichheit gilt, werden als **maximale Codes** bezeichnet.

Definition: Eine **Hadamard-Matrix** der Ordnung n ist eine quadratische (n,n)-Matrix H, deren Elemente gleich +1 oder -1 sind und für die $H H^T = nI_n$ gilt, wobei I_n die Einheitsmatrix n-ter Ordnung ist.

Bose, Shrikhande /3/ zeigten nun:

Satz: Es existiert ein maximaler $(4t,N,2t;2)$-Code und ein maximaler $(4t-1,N,2t;2)$-Code gdw. eine Hadamard-Matrix der Ordnung 4t existiert.

Goethals /4/ bewies:

Satz: Für gerades $d = ks$ existiert ein maximaler $((2k-1)s,N,ks;2)$-Code gdw. ein Blockplan mit den Parametern $v = 2k-1$, k und $\lambda = ks/2$ existiert.

Da bekannt ist, daß eine Hadamard-Matrix der Ordnung 264 existiert, ist damit auch die Existenz eines $(264,528,132;2)$-Codes, eines $(263,264,132;2)$-Codes und natürlich auch eines $2-(263,132,66)$-Blockplanes gesichert. Andrerseits ist

nicht bekannt, ob eine Hadamard-Matrix der Ordnung 268 existiert. Damit bleibt auch die Existenz eines (268,536,134;2)-Codes, eines (267,268,134;2)-Codes und eines 2-(267,134,67)-Blockplanes offen.

Optimale Codes und orthogonale lateinische Quadrate

Als weitere Schranke für die Codeparameter wollen wir jetzt die Singleton-Schranke betrachten:

Satz: Für jeden (n,N,d;q)-e-Fehler-Code gilt

$$N \leq q^{n-d+1} .$$

Codes, für deren Parameter hier die Gleichheit gilt, werden als optimale Codes bezeichnet.

Definition: Unter einem lateinischen Quadrat der Ordnung n verstehen wir eine quadratische (n,n)-Matrix, deren Elemente Zahlen aus der Menge $\{1, 2, \ldots, n\}$ sind und in der in jeder Zeile und Spalte jede Zahl i ($i \in \{1, 2, \ldots, n\}$) genau einmal vorkommt.

Definition: Zwei lateinische Quadrate $A = (a_{i,j})$ und $B = (b_{i,j})$ der Ordnung n heißen orthogonal gdw. alle n^2 geordneten Paare $(a_{i,j}, b_{i,j})$ verschieden sind.

In /6/ wurde folgende Existenzbedingung für eine Teilklasse der optimalen Codes gezeigt:

Satz: Es existiert ein (optimaler) $(n,q^2,n-1;q)$-Code gdw. n - 2 paarweise orthogonale lateinische Quadrate der Ordnung q existieren.

Es ist bekannt, daß für q = 49 48 , für q = 50 6 und für q = 51 4 paarweise orthogonale lateinische Quadrate existieren. Also existiert entsprechend jeweils ein $(50,49^2,49;49)$-, ein $(8,50^2,7;50)$- und ein $(6,51^2,5;51)$-Code. Da maximal q - 1 paarweise orthogonale lateinische Quadrate der Ordnung q existieren können, existieren keine $(q+2,q^2,q+1;q)$-Codes und damit z.B. kein $(51,49^2,50;49)$-Code.

Konstruktion von Codes

So wie einerseits kombinatorische Konfigurationen durch Codes bestimmt werden, so kann man andrerseits auch mit Hilfe solcher Konfigurationen Codes konstruieren. So gilt z.B.:

Satz: Wählt man die Zeilen der Inzidenzmatrix eines Steiner-Systems $S(t,k,v)$ als Codewörter, so erhält man einen $(v,\binom{v}{t}/\binom{k}{t},d;2)$-Code mit $d \geq 2(k-t+1)$.

Eine weitere Möglichkeit besteht in der Verwendung normalisierter Hadamard-Matrizen.

Definition: Eine Hadamard-Matrix heißt normalisiert gdw. alle Elemente der ersten Zeile und der ersten Spalte dieser Matrix jeweils gleich +1 sind.

Offensichtlich kann jede Hadamard-Matrix durch Multiplikation von Zeilen und Spalten mit -1 in eine normalisierte Hadamard-Matrix überführt werden.
Wir wollen nun eine Methode zur Konstruktion maximaler Codes auf der Grundlage einer Hadamard-Matrix angeben. Es sei eine normalisierte Hadamard-Matrix H der Ordnung n gegeben. Wir ersetzen in H +1 jeweils durch 0 und -1 jeweils durch 1. Die so erhaltene Matrix wollen wir mit A bezeichnen. Es gibt verschiedene Möglichkeiten, die Matrix A für die Konstruktion von Codes zu benutzen. Streicht man in A die erste Spalte und wählt die Zeilen der verbleibenden Matrix als Codewörter, so erhält man einen $(n-1,n,n/2;2)$-Code C. Nimmt man zu den Codewörtern von C noch deren Komplemente hinzu, so erhält man einen $(n-1,2n,n/2-1;2)$-Code. Wählt man schließlich die Zeilen der Matrix A selbst und deren Komplemente als Codewörter, so ergibt sich ein $(n,2n,n/2;2)$-Code. Wie man leicht sieht, sind alle so erhaltenen Codes maximal.
Auf dem hier zur Verfügung stehenden Raum konnte natürlich nur ein Ausschnitt der vielfältigen Beziehungen zwischen der algebraischen Codierungstheorie dargestellt werden.

Literatur

/1/ Bannai, E., On perfect codes in the Hamming schemes H(n,q) with q arbitrary. J. Combinat. Theory (A) 23 (1977), 52-67.

/2/ Blake, I. F., Mullin, R. C., The Mathematical Theory of Coding. Academic Press, New York (1975).

/3/ Bose, R. C., Shrikhande, S. S., A note on a result in the theory of code construction. Inform. and Control 2 (1959), 183-194.

/4/ Goethals, J. M., Some combinatorial aspects of coding theory. In: Srivastava, J. N. et al. (eds.), A Survey of Combinatorial Theory. North-Holland, Amsterdam (1973), chapter 17, 189-208.

/5/ Goethals, J. M., Snover, S. L., Nearly perfect binary codes. Discrete Math. 3 (1972), 65-88.

/6/ Golomb, S. W., Posner, E. C., Rook domains, Latin squares, affine planes, and error-distributing codes. IEEE Trans. Inform. Theory IT-10 (1964), 196-208.

/7/ Johnson, S. M., A new upper bound for error-correcting codes. IRE Trans. Inform. Theory IT-8 (1962), 203-207.

/8/ Lindström, K., The nonexistence of unknown nearly perfect binary codes. Ann. Univ. Turku. Ser. A. I, 169 (1975).

/9/ MacWilliams, F. J., Sloane, N. J. A., The Theory of Error-Correcting Codes. I + II. North-Holland, Amsterdam (1977).

/10/ Plotkin, M., Binary codes with specified minimum distance. IRE Trans. Inform. Theory IT-6 (1960), 445-450.

/11/ Reuvers, H., A nonexistence proof for 3-error-correcting codes. (mimeographed note) Memorandum No. 1974-13, Eindhoven University of Technology (1974).

/12/ Семаков, Н.В., Зиновьев, В.А., Зайцев, Г.В., Равномерно упакованные коды. Пробл. Перед. Инф. 7 (1971), I, 38-50.

/13/ Зиновьев, В.А., Леонтьев, В.К., Несуществование совершенных кодов над полям Галуа.
Probl. Control and Inform. Theory 2 (1973), 2, 123-132.

/14/ Tietäväinen, A., On the nonexistence of perfect codes over finite fields. SIAM J. Appl. Math. 24 (1973), 88-96.

/15/ Tietäväinen, A., Nonexistence of nontrivial perfect codes in case $q = p_1^r p_2^s$, $e \geq 3$. Discrete Math. 17 (1977), 199-205.

/16/ van Lint, J. H., Coding Theory. Lecture Notes in Mathematics, vol. 201, Springer-Verlag, Berlin (1971).

/17/ van Lint, J. H., Recent results on perfect codes and related topics. In: Hall, M., Jr., van Lint, J. H., (eds.), Combinatorics. Mathematical Centre Tracts 55, Amsterdam (1974), 158-178.

eingegangen: 15.9.1978

Anschrift des Verfassers:

Dipl.-Math. Fred Sobik
Akademie der Wissenschaften der DDR
Zentralinstitut für Kybernetik und
Informationsprozesse
DDR - 1199 Berlin
Rudower Chaussee 5

Rostock. Math. Kolloq. 10, 97 - 102 (1978)

Fred Sobik und Erdmute Sommerfeld

Klassifikation strukturierter Objekte auf der Grundlage der Isomorphie von Untergraphen

Es wird ein Ansatz vorgestellt, mit dem auf Möglichkeiten der Anwendung graphentheoretischer Prinzipien bei der Klassifikation strukturierter Objekte hingewiesen werden soll.
Allgemein besteht ein Klassifizierungsproblem darin, für eine Menge vorliegender Objekte, die Elemente bestimmter Klassen sind, Verfahren anzugeben, die jedes Objekt auf Grund einer bestimmten Beschreibung einer entsprechenden Klasse zuordnen.

Für die Klassifikation von Objekten, die durch Meßwertvektoren charakterisiert sind, liegen solche Verfahren, wie z.B. Trennebenen- und Entscheidungsbaumverfahren vor. Unser Ansatz basiert auf dem Prinzip der Entscheidungsbaumverfahren. Binäre Entscheidungsbäume lassen sich logisch durch folgenden Kalkül darstellen:

Syntax (in Bacchus-Normalform):
Alphabet t | k | * | (|)
⟨Test⟩ ::= t | ⟨Test⟩ *
⟨Klassenname⟩ ::= k | ⟨Klassenname⟩ *
⟨Entscheidungsbaum⟩ ::= ⟨Klassenname⟩ |
⟨Test⟩ (⟨Entscheidungsbaum⟩
⟨Entscheidungsbaum⟩)

$$t_i = t \underbrace{* \ldots * *}_{i\text{-mal}},\ k_j = k \underbrace{* \ldots * *}_{j\text{-mal}}$$

Wertfunktion:
Sei $x = (x_1, \ldots x_n) \in \{0, 1\}^n$
Seien E_1, E_2 Entscheidungsbäume.
Wert $(k_i, x) = k_i$.

$$\text{Wert } (t_i(E_1\ E_2), x) = \begin{cases} \text{Wert } (E_1, x), & \text{wenn } x_i = 1 \\ \text{Wert } (E_2, x), & \text{wenn } x_i = 0 \end{cases}$$

Ist ein Objekt nicht durch einen Meßwertvektor, sondern durch eine Menge von Elementarobjekten und eine Menge von ein- bzw. mehrstelligen Relationen über diesen Elementarobjekten charakterisiert, so bezeichnen wir es als s t r u k t u r i e r t e s O b j e k t . Formal hat man damit die Struktur einer relationalen Algebra. Strukturierte Objekte, wie sie u.a. auch in /1/ und /2/ untersucht werden, und ihre Klassifikation können in verschiedenen Gebieten, wie z.B. in der Bildverarbeitung, der Medizin oder Psychologie eine Rolle spielen. Ein konkretes Beispiel wären soziale Gruppenstrukturen. Zur Zeit liegen nur prinzipielle Ansätze vor, die an relativ einfachen Objektbereichen (geometrische Gebilde, Bausteinwelten) erprobt werden.

Für die Klassifikation strukturierter Objekte besteht ein wesentliches Problem darin, gemeinsame bzw. trennende Eigenschaften von Mengen von Objekten zu finden. Offensichtlich spielt das Vorhandensein bzw. Fehlen bestimmter Relationen oder Relationengruppen zwischen Elementarobjekten oder Gruppen von Elementarobjekten mit bestimmten Eigenschaften eine wichtige Rolle. Das entspricht aber gerade dem Vorhandensein oder Fehlen einer bestimmten T e i l s t r u k t u r .

Beschränkt man sich auf ein- und zweistellige Relationen, so kann man strukturierte Objekte als knoten- und kanteninterpretierte Graphen beschreiben. Wir betrachten hier nur diesen Fall. Dabei sehen wir strukturierte Objekte, deren beschreibende Graphen isomorph sind, als nicht unterscheidbar an.

Ziel des hier beschriebenen Ansatzes ist es, existierende Entscheidungsbaumverfahren (s. z.B. /3/, /4/) für die Klassifikation strukturierter Objekte anwendbar zu machen, d.h. strukturierte Objekte durch eine geeignete vektorielle Darstellung zu beschreiben. Dazu muß ein geeignetes Merkmalsbildungsprinzip gefunden werden, das (möglichst automatisiert) Merkmale liefert, denen Zahlenwerte zugeordnet werden können. Forderungen an ein solches Merkmalsbildungsprinzip sind:

- unterscheidbare Objekte müssen an Hand der Merkmalswerte unterscheidbar sein,
- nicht unterscheidbaren Objekten sollen stets die gleichen

Merkmalswerte zugeordnet werden,
- der Aufwand zur Bestimmung der Merkmalswerte muß in einem vertretbaren Rahmen bleiben.

Für die Klassifikation strukturierter Objekte wurde ein Algorithmus entwickelt, dessen Grundidee darin besteht, die Klassen durch das Vorhandensein bzw. Fehlen von Untergraphen mit einer bestimmten Struktur zu charakterisieren. Diese Eigenschaft des Vorhandenseins bzw. Fehlens eines bestimmten Untergraphen, d.h. eines zu einem bestimmten Graphen i s o m o r p h e n U n t e r g r a p h e n , wird dabei direkt als binäres Merkmal verwendet. Es wird folgendes Merkmalsbildungsprinzip realisiert:

Es sei eine Trainingsmenge von (endlichen, gerichteten, interpretierten) Graphen $G_1, G_2, \ldots, G_n$ gegeben. Für jeden Graphen G sei U(G) die Menge aller (nicht notwendig echten) induzierten Untergraphen von G. Sei

$$U^* = \bigcup_{i=1}^{n} U(G_i) .$$

Sei nun $U_0 \subseteq U^*$ eine Menge von Graphen mit der Eigenschaft, daß für alle $G_1', G_2' \in U_0$ gilt: $G_1' \not\cong G_2'$ und für alle $G \in U^*$ ein $G' \in U_0$ existiert mit $G \cong G'$. Sei $U_0 = \{G_1', \ldots, G_m'\}$. Dann können wir jedem Graphen G die Merkmale

$$t_i(G) = \begin{cases} 1, & \text{wenn ein } G_0 \in U(G) \text{ existiert mit } G_0 \cong G_i' . \\ 0 & \text{sonst} \end{cases}$$

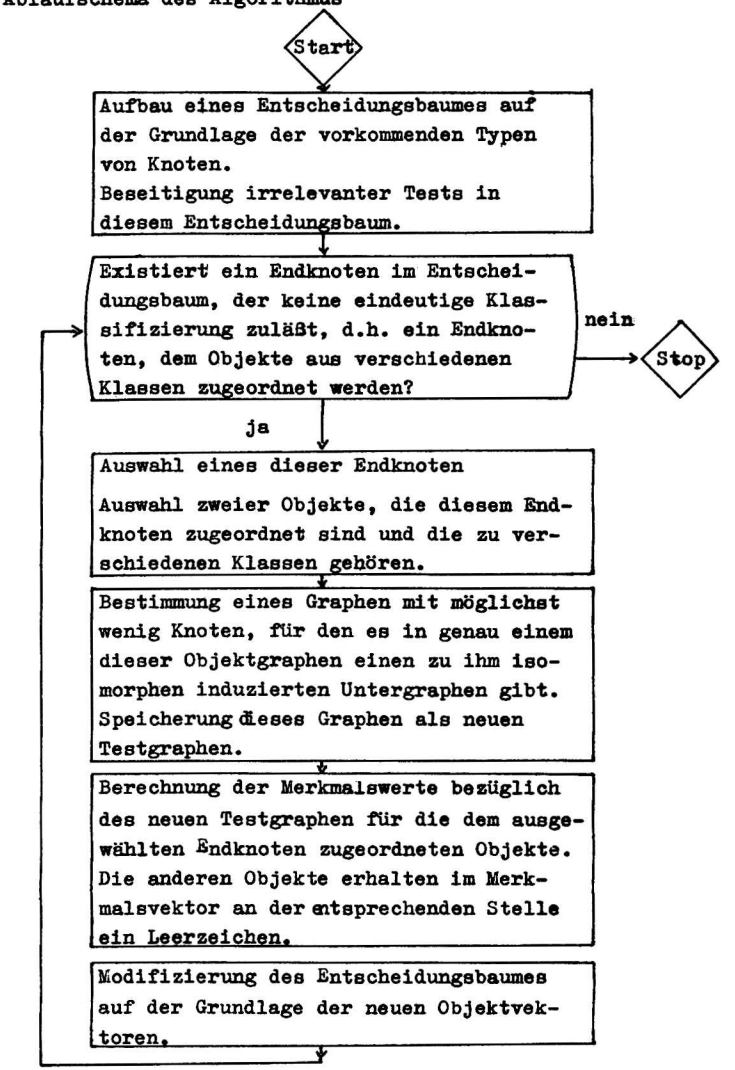
$(i = 1, \ldots, m)$ zuordnen.

Damit kann jedem Graphen G eindeutig der binäre Vektor $t(G) = (t_1(G), \ldots, t_m(G))$ zugeordnet werden. Geht man von der Suffizienz der Trainingsmenge aus, so können nun Entscheidungsbaumverfahren sinnvoll auf diese Vektoren entsprechender Graphen angewendet werden.

Da die Entscheidungsbaumverfahren sequentiell auf die Merkmalswerte zurückgreifen, werden im allgemeinen nicht alle Elemente von U_0 als Testgraphen benötigt. Außerdem kann man davon ausgehen, daß zwischen Objekten ein- und derselben Klasse gewisse Gemeinsamkeiten existieren. Dies wird im Algorithmus neben weiteren Prinzipien zur Senkung des Aufwands

berücksichtigt, so daß damit die sonst zu umfangreiche Menge U_0 von Untergraphen handhabbar wird. Eine ausführliche Beschreibung dieses Algorithmus ist in /5/ enthalten.
Im Rahmen der Klassifikation interpretierter Graphen auf der Grundlage der Auswahl von Untergraphen bestehen die wesentlichen algorithmischen Probleme in der Entscheidung der Isomorphie von Graphen und in der Auswahl geeigneter Untergraphen. Die Entscheidung der Isomorphie zweier Graphen ist im allgemeinen Fall relativ aufwendig. Beim Aufbau des Entscheidungsbaumes und bei der Klassifikation mit Hilfe dieses Entscheidungsbaumes muß in der Regel mehrmals die Prüfung der Isomorphie von Untergraphen und Testgraphen vorgenommen werden. Es wäre daher sinnvoll, als Testgraphen möglichst Graphen zu wählen, für die die Entscheidung der Isomorphie mit geringem Aufwand verbunden ist (z.B. Graphen niedriger Ordnung, Bäume, planare Graphen).
Neben diesen algorithmischen Problemen sind die Problematik der Repräsentation von Graphen durch Untergraphen und die Problematik der Strukturtransformation zur Extraktion relevanter Teilstrukturen von Bedeutung. Bei der Problematik der Repräsentation von Graphen durch Untergraphen haben wir es mit der Frage zu tun, ob Graphen durch Untergraphen (bis auf Isomorphie) eindeutig charakterisiert werden können und wenn ja, durch welche Untergraphen dies geschehen kann. Bei der Strukturtransformation zur Extraktion relevanter Teilstrukturen geht es um die Frage, ob eine Vereinfachung der Struktur (z.B. eine Vergröberung) der zu klassifizierenden Graphen vorgenommen werden kann, ohne für die Klassifikation relevante Information zu verlieren. Dabei können sowohl Eigenschaften des Objektbereichs (z.B. funktionelle Zusammenhänge) als auch strukturelle Eigenschaften des Graphen eine Rolle spielen. Sowohl zur Repräsentation von Graphen als auch zum Problem der Strukturtransformation sind noch Grundlagenuntersuchungen erforderlich.

Ablaufschema des Algorithmus



Literatur

/1/ Hayes-Roth, F., Uniform representations of structured patterns and an algorithm for the induction of contingency response rules, Inform. and Control 33 (1977), s, 87-116.

/2/ Winston, P. H., Learning structural descriptions from examples. MIT-Report MAC TR-76, Cambridge, Mass. (Sept. 1970).

/3/ Weiterentwicklung von Metaalgorithmen zur Erzeugung von Klassifizierungsalgorithmen (Entscheidungsbäumen). AdW d. DDR, ZKI, Forschungsbericht 132 (1972).

/4/ Algorithmische Entwicklung von Entscheidungsbäumen für statistische Klassenbildungen. AdW d. DDR, ZKI, Forschungsbericht 196 (1973).

/5/ Sobik, F., Sommerfeld, E., Ein Ansatz zur Klassifikation strukturierter Objekte. AdW d. DDR, ZKI, Teilbericht des Forschungsberichts KK (1978).

eingegangen: 15.9.1978

Anschrift der Verfasser:
Dipl.-Math. Fred Sobik, Dipl.-Ing. Erdmute Sommerfeld
Akademie der Wissenschaften der DDR
Zentralinstitut für Kybernetik und
Informationsprozesse
DDR - 1199 Berlin
Rudower Chaussee 5

Rostock. Math. Kolloq. 10, 103 - 107 (1978)

Dietmar Uhlig

Boolean Functions with Linear Combinational Complexity

In this paper we consider the combinational complexity of Boolean functions satisfying a certain property, $p^n_{k,m}$. This property has been investigated by W. N. Hsieh, L. H. Harper and J. E. Savage /1/.

We consider binary combinatorial networks, or simply, networks, and functions they compute. A (binary) network is a directed nodelabelled acyclic graph such that the graph consists of n input nodes, of output nodes and of 2-input gates. (Precise definition see, for example, /1/, /2/ or /3/). The combinational complexity, $L(\mathcal{A})$, of a network $\mathcal{A}$, is the total number of gates in $\mathcal{A}$. The combinational complexity, L(f), of a Boolean function f, is the minimum of $L(\mathcal{A})$, where $\mathcal{A}$ ranges over all networks with a unique output node computing f. The Shannon function, $L(\mathcal{F})$, of a family $\mathcal{F}$ of Boolean functions is the maximum of L(f), where f ranges over all functions in $\mathcal{F}$. Further, by $c(\mathcal{F})$ is denoted the minimum of L(f), where f ranges over all functions in $\mathcal{F}$. (The definition of $c(\mathcal{F})$ is given in /1/.) Thus, $L(\mathcal{F})$ (or $c(\mathcal{F})$) is the number of two input gates sufficient to construct networks for all (or for one) of the functions in $\mathcal{F}$.

Let G_n be the set of all Boolean functions of the variables $x_1,\ldots,x_n$. A well-known result due to O. B. Lupanow /2, 3/ establishes that

$$L(G_n) \sim \frac{2^n}{n} \quad .^1 \qquad (1)$$

1 $a(n) \sim b(n)$ denotes $\lim \frac{a(n)}{b(n)} = 1$.

Let $f \in G_n$, let $Y = \{y_1,\ldots,y_k\}$ and $Y \subseteq \{x_1,\ldots,x_n\}$.
A Boolean function g is said to be a subfunction of the function f of the variables $y_1,\ldots,y_k$, if g is obtained from f by setting each x_j, $x_j \in \{x_1,\ldots,x_n\} \setminus Y$, for 0 or 1.
Let $S_Y(f)$ be the number of distinct subfunctions of the function f of the variables $y_1,\ldots,y_k$.
Let $S(f) = \sum_{Y \subseteq \{x_1,\ldots,x_n\}} S_y(f)$. This integer S(f) is called the number of subfunctions of f.
S. W. Jablonski /7/ has shown that for every positive number ε "almost" all Boolean functions of n variables have at least $3^n(1-\varepsilon)$ and at most 3^n subfunctions. The author of this paper shows /4, 5/ that L(f) is dependent on S(f). For instance, if $S(f) \leq \text{const.} \cdot 2^n$, then f has a linear (according to the number of variables) combinational complexity /4/. In /5/ is given a Theorem, which can be interpreted as follows: There is a function B(x) (given constructively) such that $B(x) < x$ for $0 < x < 1$ and such that if the number n of variables of a Boolean function f is sufficiently large and if S(f) is nearly $\gamma \cdot 3^n$, then the probability of the event "L(f) is nearly $B(\gamma)\frac{2^n}{n}$" is nearly 1. This also shows the dependence of L(f) from S(f). But on the other hand there is no or only a little dependence of S(f) from L(f). For example, there are Boolean functions of n variables having linear combinational complexity and "almost" 3^n subfunctions /6/.
Similar statements are obtained by using the $p^n_{k,m}$ property. A Boolean function depending on $x_1,\ldots,x_n$ has the $p^n_{k,m}$ property, is for each subset $\{y_1,\ldots,y_{n-k}\}$ of $\{x_1,\ldots,x_n\}$ there are at least m subfunctions of f of the variables $y_1,\ldots,y_{n-k}$. The set of functions having the $p^n_{k,m}$ property is denoted by $P^n_{k,m}$. In /1/ are given $c(P^n_{1,2})$, $c(P^n_{2,3})$ and $c(P^n_{3,5})$. There is also shown that for $k > 0$ there are infinitely many n with $c(P^n_{k,2^k}) \leq 13(n+1)$. In this paper we give the idea of the proof of the following.

Theorem: For each $k > 0$,

$$c\,(P^n_{k,2^k}) \lesssim 3n\,.$$ [1]

Corollary: For each $k > 0$, every positive number ε and sufficiently large n

$$c\,(P^n_{k,2^k}) \leq (3+\varepsilon)\,n.$$

Idea of proof: A function h_n with the properties $P^n_{k,2^k}$ and $c(h_n) \lesssim 3n$ is the function

$$h_n = \sum_{0 \leq |\tilde{\sigma}| \leq n-r\cdot t-1} P_{\sigma_1}(g_1) \cdot \ldots \cdot P_{\sigma_t}(g_t) \cdot x_{|\tilde{\sigma}|} \qquad (2)$$

where

1) $\sum$ denotes sum modulo 2,

2) n is a sufficiently large number,

3) $\tilde{\sigma}$ denotes $(\sigma_1,\ldots,\sigma_t)$,

4) $|\tilde{\sigma}|$ denotes $\sum_{i=1}^{t} \sigma_i\, 2^{i-1}$

5) $P_0(0) = P_1(1) = 1$ and $P_0(1) = P_1(0) = 0$,

6) $r = \left[\frac{1}{10}\ \log_2 n\right]$, $t = \left[\log_2 n + 1\right]$,

7) g_i is a Boolean function of the variables $z^i_1,\ldots,z^i_r$, $(i = 1,\ldots,t)$ and has the $P^r_{k+1,2^{k+1}}$ property,

8) $\{z^i_1,\ldots,z^i_r\} \cap \{z^j_1,\ldots,z^j_r\} = \Lambda$ for $i \neq j$,

9) $\{z^i_1,\ldots,z^i_r\} \subset \{x_{n-rt},\ldots,x_n\}$.

It is not difficult to show that the number of functions of r variables which have not the $P^r_{k,2^k}$ property, is $o(2^{2^r})$, and therefore "almost" all functions of r variables have the $P^r_{k,2^k}$

---

1 $a(n) \lesssim b(n)$ denotes $\overline{\lim}\ \frac{a(n)}{b(n)} \leq 1$.

property. Thus, there exists a function g_i satisfying 7). But for certain n it is also possible to apply the function used in /1/ as a function with this property. It follows that each subfunction obtainable from function g_i by setting at most k variables to 0 or 1 is non-constant, and from this follows that h_n has the $p^n_{k,2^k}$ property. Now we show that

$$L(h_n) \lesssim 3n \ . \tag{3}$$

First note that there exists a network $\mathcal{A}$ realizing all of the functions

$$P_{\sigma_1}(y_1) \cdot \ldots \cdot P_{\sigma_t}(y_t) \qquad (0 \le |\tilde{\sigma}| \le n-rt-1)$$

and satisfying the inequality

$$L(\mathcal{A}) \lesssim n-rt \qquad \text{(see, for example, /6/).}$$

Further we have by 6), 7) and by (1)

$$\sum_{i=1}^{r} L(g_i) \lesssim t\,\frac{2^r}{r} \lesssim 10\ \sqrt[10]{n} = o(n) \ .$$

Now we take n-rt 2-input ANDs for realizing all

$$P_{\sigma_1}(y_1) \cdot \ldots \cdot P_{\sigma_t}(y_t) \cdot x_{|\tilde{\sigma}|}$$

(by the assumption that the functions $P_{\sigma_1}(y_1) \cdot \ldots \cdot P_{\sigma_t}(y_t)$ have already been realized).

Then we take n-rt-1 gates for realizing

$$\bigvee_{0 \le |\tilde{\sigma}| \le n-r\cdot t-1} P_{\sigma_1}(y_1) \cdot \ldots \cdot P_{\sigma_t}(y_t) \cdot x_{|\sigma|} \ . \tag{4}$$

Connecting the outputs of networks realizing $g_1,\ldots,g_t$ with the inputs $y_1,\ldots,y_t$ of network realizing (4) we get a network realizing function (2) and satisfying (3).

Literature

/1/ **Hsieh, W. N.; Harper, L. H.; Savage, J. E., A class of Boolean functions with linear combinational complexity, Proj. MAC Techn. Memo, 5, 1974.**

/2/ Лупанов, О.Б., Об одном методе синтеза схем, Радиофизика I, I, 1958, 120-140.

/3/ Лупанов, О.Б., О синтезе некоторых классов управляющих систем, Сб. "Проблемы кибернетики", вып. 10, Москва, Физматгиз, 1963, 63-97.

/4/ Улиг, Д., О связи между сложностью схемной реализации фуикций алгебры логики и числом их подфункций, Сб. "Проблемы кибернетики", вып. 26, Москва, Наука, 1972, 183-201.

/5/ Улиг, Д., Об одном семействе классов просто реализуемых функций алгебры логики, Сб. "Проблемы кибернетики", вып. 28, Москва, Наука, 1974, 25-42.

/6/ Улиг, Д., Об одной функции алгебры логики, имеющей много подфункций и небольшую сложность реализации, Сб. "Проблемы кибернетики", 1978.

/7/ Ядонский, С.В., Об алгоритмических трудностях синтеза минимальных контактных схем, Сб. "Проблемы кибернетики", вып. 2, Москва, Физматгиз, 1959, 75-121.

eingegangen: 15.9.1978

Anschrift des Verfassers:

Dr.sc.nat. Dietmar Uhlig
IH Mittweida
WB Mathematik/Informationsverarbeitung
DDR - 925 Mittweida
Platz der DSF 17

Rostock. Math. Kolloq. 10, 109 - 114 (1978)

Lutz Voelkel

Fehlerdiagnose digitaler Schaltungen - Probleme und Methoden

1. Einführung

Bei der Herstellung digitaler elektronischer Schaltungen, die mit integrierten Bausteinen bestückt sind, ist die Fehlerquote relativ hoch (nach /7/ ca. 50 %). Dies führt dazu, daß die Zeit für die Fehlerdiagnose einen bedeutenden Anteil der gesamten Produktionszeit ausmacht. Eine wesentliche Zeiteinsparung gegenüber der "traditionellen" Diagnose von Hand kann durch die Einführung von rechnergesteuerten Diagnoseverfahren erzielt werden.
Der Begriff "Fehlerdiagnose" wird im folgenden Sinne verwendet: Zunächst wird eine Fehlererkennung (Funktionsprüfung) durchgeführt. Ist der Prüfung fehlerhaft, so erfolgt die Fehlerlokalisierung, an die sich dann die Reparatur der defekten Bauteile anschließt
Im 2. Abschnitt werden Grundbegriffe aus der Fehlerdiagnose-Theorie eingeführt. Den Inhalt des 3. Abschnittes bilden Überlegungen zur prinzipiellen Schwierigkeit von Fehlererkennungsproblemen. In Nr. 1 wird kurz auf einige Verfahren zur Fehlererkennung und -lokalisierung bei kombinatorischen und sequentiellen Schaltungen eingegangen.

2. Grundbegriffe

S sei eine Schaltung mit n Eingängen und k Ausgängen. Ein Eingabevektor $t \in \{0,1\}^n$ heißt ein Erkennungstest (oder kurz Test) für den Fehler F von S, wenn es einen Ausgang i (und, falls S r Speicher hat, einen von jedem Zustand erreichbaren Speicherzustand $q \in \{0,1\}^r$) gibt, so daß der Wert der Ausgangsvariablen y_i^F der mit dem Fehler F behafteten Schaltung bei Eingabe von t (im Zustand q) vom entsprechenden Ausgangswert y_i der fehlerfreien Schaltung verschieden ist. Als Unter-

scheidungstest für das Fehlerpaar (F,F') bezeichnet man eine Eingangsbelegung , für die analog zum obigen Fall $y_i^F \neq y_i^{F'}$ gilt.

Um verschiedene Fehler erkennen, bzw. voneinander unterscheiden zu können, werden i.a. mehrere Teste benötigt. Bei einer kombinatorischen Schaltung faßt man dies meist einfach zu einer Testmenge zusammen, deren Elemente in beliebiger Reihenfolge abgearbeitet werden können. Bei sequentiellen Schaltungen, für die neben den "reinen" Tests noch Zustandsüberführungen zu realisieren sind, verwendet man Testfolgen. Für beide Fälle wird im folgenden die Bezeichnung "Testsatz" gewählt. S sei Schaltung, F eine Klasse von Fehlern. Ein Testsatz T, der zu jedem Fehler $F \in F$ einen Erkennungstest enthält, heißt vollständiger Erkennungstestsatz für S bezüglich F. Gibt es sogar zu jedem Paar (F,F') einen Unterscheidungstest $t \in T$, so wird T ein vollständiger Lokalisierungstestsatz für S bezüglich F genannt. Die Aufstellung von vollständigen Erkennungs- bzw. Lokalisierungssätzen für digitale Schaltungen bezüglich möglichst umfangreicher Fehlerklassen ist eine mögliche (und die meistangewandte) Diagnosestrategie, zumindest für die Fehlererkennung (zur Lokalisierung vergleiche Nr. 4). In der Mehrzahl der Arbeiten zur Fehlerdiagnose beschränkt man sich auf die Betrachtung der sogenannten "stuck-at" oder Festfehler. Dabei liegt der Fehler s-a-i für eine (Eingangs-, Ausgangs- oder innere) Variable genau dann vor, wenn die dieser Variablen zugeordnete Leitung ständig das Signal i führt, i = 0,1. Eine Motivierung dieser Fehlertypen durch verschiedene Defekte an Transistorenschaltungen findet man in /5/. Im folgenden beziehen sich Bemerkungen über die Vollständigkeit eines Testsatzes ohne Relativierung auf eine Fehlerklasse stets auf die Klasse aller Einzelfehler vom Typ s-a-0 bzw. s-a-1. Es sei angemerkt, daß in einigen Artikeln umfangreichere Fehlerklassen betrachtet werden, zu denen z.B. noch Kurzschluß-, Zwischenwert- und Mehrfachfehler der genannten Arten gehören.

Eine Schaltung S wird als irredundant bezeichnet, wenn jeder Einzelfehler vom Typ s-a-0 bzw. s-a-1 erkennbar ist, d.h. wenn für S ein vollständiger Erkennungstestsatz existiert.

3. NP-vollständige Probleme bei der Erkennung von Fehlern in kombinatorischen Schaltungen

NP sei die Klasse aller Probleme, die mit nichtdeterministisch arbeitenden Turingmaschinen in polynomialer Schrittzahl (abhängig von der Länge der Eingabewörter) lösbar sind; P bezeichne die Klasse der Probleme, für die dies mit determinierter Arbeitsweise möglich ist. Die Frage, ob P = NP ist, gehört zu den klassischen ungelösten Problemen der Komplexitätstheorie, die Vermutung $P \neq NP$ wird auch als Cooksche Hypothese bezeichnet. Seit 1971/72 sind viele Probleme aus den verschiedensten Bereichen als so schwer nachgewiesen worden, daß ihre Zugehörigkeit zu P das Zusammenfallen von ganz NP mit P zur Folge hätte; solche Probleme werden als NP-vollständig bezeichnet. Ibarra und Sahni zeigten in /8/, daß (neben anderen) die beiden folgenden Probleme NP-vollständig sind:

1) Ist eine kombinatorische Schaltung irredundant?
2) Realisiert eine kombinatorische Schaltung eine vorgegebene Boolesche Funktion?

Der Beweis wird durch Polynomialzeit-Reduktion des als NP-vollständig bekannten Tautologie-Problems für alternative Normalformen auf die genannten Probleme geführt.

4. Zu Methoden der Fehlererkennung und -lokalisierung

Den meisten praktisch verwendeten automatischen Prüfverfahren liegt folgendes Prinzip zugrunde: Die Eingänge des Prüflings werden der Reihe nach mit den Tests eines als vollständig angenommenen Erkennungstestsatzes belegt. In jedem Schritt erfolgt dabei ein Vergleich der Ausgangssignale mit denen der fehlerfreien Schaltung, letztere können durch Schaltungssimulation (z.B. auf einem Großrechner) erzeugt oder an einem fehlerfreien Musterexemplar gemessen werden. Bei dieser Vorgehensweise ist das Problem der Fehlererkennung somit im wesentlichen dem der Testsatzgenerierung gleichwertig. Für diese gibt es eine Reihe von systematischen Verfahren, die mehr

(wie z.B. der D-Algorithmus, vergl. /12/, /6/, /1/) oder weniger (wie z.B. die Fehlermatrizenmethode, vergl. /5/) gut für eine Abarbeitung auf großen Rechenanlagen geeignet sind und zumindest für irredundante kombinatorische Schaltungen einen vollständigen Erkennungstestsatz garantieren. Für eine nähere Beschreibung solcher Verfahren, ihre Anwendbarkeit auf sequentielle Schaltungen usw. sei auf die angegebene Literatur verwiesen.

Als eine Vorstufe für Testsatzgenerierungsprogramme können solche zur Testsatzanalyse angesehen werden. Mit solchen kann man heuristische Methoden erzeugte Teilsätze auf Vollständigkeit überprüfen und Hinweise zur Vervollständigung geben (vergl. /4/, /6/, /7/).

Heuristische Verfahren zur Testsatzgenerierung werden noch vielfach verwendet, vor allem für kleinere Schaltungen. Sie sind i.a. sehr eng an die spezielle Schaltungsstruktur angelehnt und können gewisse für den Prüfablauf günstige Nebenbedingungen meist gut erfüllen. Obwohl das Anliegen der Fehlererkennung eigentlich nur die Binärentscheidung "fehlerfrei - fehlerhaft" ist, liefern fast alle praktizierten Erkennungsverfahren doch weit mehr Informationen, meist werden Fehlerprotokolle mit Angaben der Form "im Testschritt x Ausgang y fehlerhaft" ausgegeben. Diese Informationen stellen Lokalisierungshilfen dar, ihre Verbesserung ist schon Aufgabe der Fehlerlokalisierung. Die Aufstellung eines vollständigen Lokalisierungstestsatzes, mit dem die Lokalisierung nach dem oben für die Erkennung beschriebenen Prinzip ablaufen könnte, ist in vielen Fällen nachweislich unmöglich oder nur mit unvertretbar hohem Aufwand realisierbar, andererseits eine Unterscheidung von Fehlern innerhalb einer bestimmten Baugruppe unnötig. Aus diesen Gründen arbeitet man bei der Lokalisierung meist mit unvollständigen Testsätzen, zur Erhöhung der Lokalisierungsgenauigkeit werden dann andere Hilfsmittel herangezogen. Ein solches, daß die sogenannten Abtastverfahren (vergl. /7/, /11/) charakterisiert, besteht darin, daß neben den Ausgangswerten auch die an inneren Punkten der Schaltung anliegenden Signale ausgewertet werden (technisch zu realisieren

durch Nadeln oder Chipabtaster). Im einfachsten Fall geht man mit einem solchen Verfahren von einem Erkennungstestsatz aus (Fehlerpfad-Rückverfolgung) und kann zumindest für irredundante kombinatorische Schaltungen alle Einzel-Festfehler lokalisieren. Für sequentielle Schaltungen sind i.a. weitere Überlegungen erforderlich, außerdem ist ein solches Abtastverfahren für Schaltungen hinreichend großer Tiefe zu zeitaufwendig. Daher wird meistens ein zwar unvollständiger, doch über den zur Prüfung benötigten weit hinausgehender Lokalisierungstestsatz zugrunde gelegt, und auf ein Abtastverfahren erst zur Trennung von mit diesem Testsatz nicht mehr unterscheidbaren Fehlern zurückgegriffen. Für eine nähere Beschreibung von Verfahren zur Aufstellung solcher Testsätze sowie weiterer Lokalisierungsmethoden sei auf die angegebene Literatur verwiesen.

Literatur

/1/ Bobey, K. und Heinz, E., Dissertation A, Humboldt-Univ. 1975.

/2/ Chang, H. Y., Manning, E. G., Metze, G.; Fault Diagnosis of Digital Systems. New York 1970.

/3/ Ebel, B., Dissertation, RWTH Aachen 1976.

/4/ Fallberg, G., Kubisch, J., IfR-Mitteilung 1975 (mar 18H.6)

/5/ Görke, W., Fehlerdiagnose digitaler Schaltungen. Stuttgart 1973.

/6/ Habiger, E., Küchler, D., Lange, S., Walter, U., AGENTEFO..., Informationsartikel in mar 19H.11, 1976.

/7/ Institut für Nachrichtentechnik Berlin, interner Studienbericht.

/8/ Ibarra, O. H., Sahni, S. K., IEEE Trans. Comp. C-24/3, 1975.

/9/ Küchler, D., Dissertation A, TU Dresden 1974.

/10/ Küchler, D., Plietzsch, Thiems, mar 19H.11 u. H.12, 1976.
/11/ Matthies, Merz, messen und prüfen / automatik, Mai 1976.
/12/ Roth, J. P., Bouricius, W. G., Schneider, P. R., IEEE Tr. EC - 16, H. 5, 1967.
/13/ Schenderlein, G., Krämer, D., IMEKO-Reprints VI, Section 2, Juni 1973.

Literaturzusammenstellung zur Fehlerdiagnose

1. Bannetts, R. G., Lewin, D. W., The Computer Journal 14, 1971.
2. Habinger, E., Lange, S., Oelschlegel, G., mar 2oH.5 u. H.6, 1977.
3. INT/OT-interne Literaturzusammenstellung (Auszüge in /7/).

eingegangen: 15.9.1978

Anschrift des Verfassers:

Dr. Lutz Voelkel
Ernst-Moritz-Arndt-Universität
Sektion Mathematik
DDR - 22 Greifswald
Jahnstraße 15a

Rostock. Math. Kolloq. 10, 115 - 121 (1978)

Stefan von Weber

Zur Definition eines speziellen algorithmischen Fehlers mittels Turingmaschinen

Zu jeder Aktion eines künstlichen oder menschlichen Automatons erwarten wir bestimmte Resultate, d.h. Informationen. Als fehlerhaft bezeichnen wir solche Informationen, die bestimmte von uns geforderte Bedingungen nicht erfüllen. Die Suche von Fehlern und die Korrektur fehlerhafter Daten oder Programme ist eine Haupttätigkeit des EDV-Anwenders. In fast allen Programmiersprachen wird deshalb dieser Problematik Aufmerksamkeit gewidmet. Von theoretischer wie auch praktischer Seite finden wir Beiträge zu diesem Thema z.B. bei Bachmann /1/, Burgess /2/, Grishman /3/ und Randell /4/.

Der zeitliche und logische Ablauf einer Problemlösung mittels eines der bekannten und üblichen Programmiersysteme besteht im allgemeinen aus den separierbaren Schritten Kodierung, lexikalische und syntaktische Analyse, semantische Analyse und Synthese, Systemgenerierung, Interpretation des Algorithmus. Ein solches System bezeichnen wir als endlichen Stufenprozeß und definieren es als Turingmaschine mit speziellen Eigenschaften.

Definition 1: Als endlichen Stufenprozeß bezeichnen wir einen Algorithmus, darstellbar durch eine Turingmaschine T mit den folgenden Eigenschaften:

a) Die Zustandsmenge besteht aus n disjunkten Teilmengen $Q_1, Q_2, \ldots, Q_n$. Der Startzustand befindet sich in der Teilmenge Q_1, der Stopzustand in der Teilmenge Q_n.
b) Das Band besteht aus n+1 endlichen Zonen $Z_0, Z_1, \ldots, Z_n$.
c) Die Zone Z_0 enthält das Eingabewort für den Stufenprozeß. Die Zonen $Z_1, \ldots, Z_n$ sind den Teilmengen $Q_1, \ldots, Q_n$ zugeordnet.
d) Befindet sich T für ein beliebiges Eingabewort Z_0 zum er-

sten Mal in einem Zustand $q \in Q_i$, dann hat der Kopf M bis zu diesem Zeitpunkt nur die Zonen $Z_0, Z_1, \ldots, Z_{i-1}$ berührt.

e) Es existiert kein $q_{ij} = g(q_i, s_j) \cap q_{ij} \in Q_r \cap q_i \in Q_s \cap r < s$ mit $g(q,s)$ als Übergangsfunktion von T.

Für die Stufen des endlichen Stufenprozesses gilt der Satz: Ein Fehler in der Stufe i kann nur mit Sicherheit erkannt werden, wenn kein Fehler in den Stufen 1,2,...,i-1 auftritt. Der Beweis dieses Satzes beruht auf der Tatsache, daß jede Stufe i des endlichen Stufenprozeßes als homomorphe Abbildung der alten Zoneninhalte $Z_0, Z_1, \ldots, Z_i$ auf die neuen Zoneninhalte $Z'_0, Z'_1, \ldots, Z'_i$ zu sehen ist. Auf Grund des Homomorphismus der Abbildung ist diese nicht umkehrbar. Der Gegenstand der vorliegenden Arbeit ist nun die algorithmische Darstellung des Fehlererkennungsprozesses der Algorithmusinterpretation mit dem Fernziel, eine geschlossene Definition der Fehlererkennung in einem Programmiersystem zu gewinnen - unter der Voraussetzung, daß dieses sich durch einen endlichen Stufenprozeß darstellen läßt.

Die Theorien der Algorithmenverifikation /5/ beruhen auf der Tatsache, daß sich Algorithmen auf verschiedene Weise darstellen lassen oder aber, daß sich bestimmte Eigenschaften eines Algorithmus unabhängig von der Darstellung des Algorithmus angeben lassen. Unter einem algorithmischen Fehler verstehen wir die Inäquivalenz zweier Darstellungen eines Algorithmus bzw. das Fehlen bestimmter Eigenschaften, die explizit vom Algorithmus gefordert werden. In dieser Arbeit wird nun ein spezieller algorithmischer Fehler der zweiten Kategorie betrachtet, d.h. daß von einem fehlerlosen Algorithmus im Sinne unserer Definition eine ganz bestimmte Eigenschaft gefordert wird. Zum Begriff des Algorithmus benutzen wir ausschließlich die Definition mittels Turingmaschinen, wie sie bei Trachtenbrot /6/ oder Minsky /7/ zu finden ist.

Ein Algorithmus A ist die durch eine Turingmaschine T^A definierte Transformationsvorschrift für eine auf dem Band der Maschine T^A befindliche Anfangsbelegung.

Als Situation der Turingmaschine T bezeichnen wir das Paar

(q_i, s_j). Dabei ist q_i der Zustand der Maschine T und s_j das Zeichen unter dem Kopf M_t der Maschine.

Definition 2: Als algorithmischen Fehler 1. Art f_1^n bezeichnen wir das (n+1)-malige unmittelbar nacheinander erfolgende Auftreten identischer Situationen während der Arbeit des Algorithmus A, dargestellt durch die Turingmaschine T^A, mit der beliebigen Anfangsbelegung W des Bandes von T^A.

Diese Definition des algorithmischen Fehlers 1. Art deckt sich mit in der Programmierung digitaler Rechenanlagen auftretenden Situationen. Als Beispiele können Fehlerarten wie Überlauf Speicherende, Überlauf Bandende oder Dateiende angeführt werden.

Satz: Zu jeder ganzen Zahl $n \geq 1$ existiert ein Algorithmus U, der für jedes beliebige Tupel, bestehend aus einem Algorithmus A und einer Anfangsbelegung W, mit einem algorithmischen Fehler 1. Art diesen Fehler erkennt und durch einen speziellen Endzustand anzeigt.

Beweis: Wir beweisen den Satz durch die Konstruktion einer Turingmaschine U mit den angegebenen Eigenschaften. Ohne Einschränkung der Allgemeinheit setzen wir voraus, daß T^A ein Band besitzt, das nach links unendlich ist und mit dem Alphabet $\{0,1\}$ arbeitet. Wir ordnen auf dem Band von U folgende Informationen von links nach rechts an:

- Die Anfangsbelegung der Maschine T^A, wobei das Zeichen unter dem Kopf M_t der Maschine T^A durch das Hilfssymbol M ersetzt ist.
- Ein Trennzeichen Y.
- Den geeignet durch k Zeichen aus dem Alphabet $\{0,1\}$ verschlüsselten aktuellen Zustand q_{ot} von T^A.
- Das Zeichen s_{ot} unter dem Kopf M_t.
- Eine Menge P von Quintupeln mit der Beschreibung der Maschine T^A. Ein Quintupel p hat die Form $p=(X,q_i,s_j,q_{ij}, s_{ij},d_{ij})$. Dabei ist X ein Trennzeichen, q_i der i-te Zustand von T^A und s_j das j-te Zeichen aus dem Alphabet von T^A, das im Zustand q_i erwartet wird. q_{ij} ist der Folgezustand

der Situation (q_i, s_j), s_{ij} das ausgegebene Zeichen und d_{ij} die kodierte Bewegungsrichtung des Kopfes M_t. q_i und q_{ij} sind im gleichen k-stelligen Kode, wie q_{ot}, so kodiert, daß alle Kodierungen der Zustände von T^A paarweise verschieden sind. d_{ij} ist durch das Zeichen 0 für die Linksverschiebung von M_t und durch das Zeichen 1 für die Rechtsverschiebung von M_t kodiert.

- Ein Trennzeichen Y.
- (n-1) mal das Zeichen 0 (Strichliste).
- Ein Trennzeichen Y.

Das äußere Alphabet der sogenannten universellen Turingmaschine U besteht aus dem Alphabet {0,1} von T^A, den Trennzeichen X,Y sowie den Hilfssymbolen A,B,C,D,G,H,M. Die Arbeit von U beschreiben wir durch einen gerichteten Graphen /8/ G(U) mit den folgenden Eigenschaften: Jeder Bogen von G(U) beschreibt einen Takt oder Arbeitszyklus von U, wobei der Bogen durch das geordnete Paar (s_i, s_{ij}) gekennzeichnet wird. Die Bewegungsrichtung des Kopfes M_u von U wird im Zielknoten des Bogens angezeigt. Bögen, für die $s_i = s_{ij}$ gilt und für die Start- und Zielknoten identisch sind, werden nicht aufgeführt. Die Knoten des Graphen G(U) kennzeichnen dabei die Zustände der Turingmaschine U.

Abb. 1 zeigt den Graphen G(U). Darin beschreibt der Teilgraph mit den Knoten a,b,c,d,e,f das Aufsuchen des Quintupels auf dem Band von U, das zu q_{ot} und s_{ot} gehört. Durch den Teilgraph g,h,i,j,k,l,m wird die Prüfung auf Identität von q_i und q_{ij} beschrieben. Der Teilgraph n,o,L1,L2,L3,L4,p,q,r,s,t,u,v, w,x,y,L,I beschreibt die Prüfung auf Identität von s_j und s'_j, wobei s'_j das Zeichen neben dem Kopf M_t ist, und zwar auf der Seite, die durch d_{ij} angegeben ist. Der Teilgraph M1,M,F beschreibt die Verlängerung der Strichliste im Falle identischer Situationen von T^A und den Übergang zum Fehlerzustand F bei Erkennen von f_1^n. Der Teilgraph I2,K beschreibt das Löschen der Strichliste für den Fall nichtidentischer aufeinanderfolgender Situationen. Der Teilgraph N,O beschreibt das Entfernen der Hilfssymbole C,D,G,H vom Band der Maschine U. Der Teil-

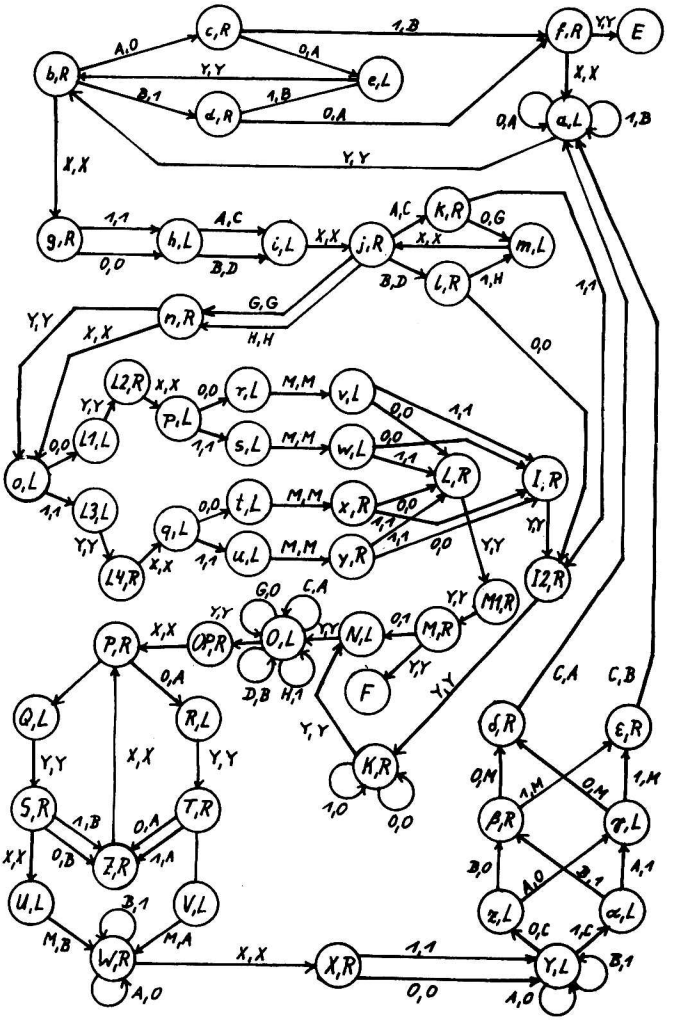


Abb.1 Graph G(U)

graph OP,P,Q,R,S,T,U,V,W,X,Y,z,$\alpha,\beta,\gamma,\sigma,\delta$ beschreibt die Ausführung eines Arbeitszyklus von T^A durch die bekannte universelle Turingmaschine von Minsky.
Die Maschine U interpretiert demnach die Arbeit der Maschine T^A und geht für jedes Algorithmus-Anfangsbelegung-Paar mit einem Fehler f_1^n in den Endzustand F über. Damit ist der Satz bewiesen.
Mit der dargestellten Methode lassen sich weitere Fehlerkategorien definieren. Allen Fehlerdefinitionen ist jedoch gemeinsam, daß sie nur für Algorithmen und Anfangsbelegungen gelten, die eine endliche Zahl von Zeichen zu ihrer Darstellung in einer universellen Turingmaschine benötigen.

Literatur

/1/ Bachmann, P., Grundlagen der Compilertechnik, Leipzig, 1975.

/2/ Burgess, C. J., Compile-time error diagnostics in syntax-directed compilers, Computer J., 15, 302-307, (1972).

/3/ Grishman, R., Criteria for a debugging language, Express-inf. Rechentechn., 15, 1-5, (1974).

/4/ Randell, B., Operating systems - the problems of performance and reliability, Proceedings IFIP-congr. 1971.

/5/ Stoyan, H., Der gegenwärtige Stand bei der Implementierung von Systemen zur Programmverfikation, Schriftenreihe WBZ MKR TU Dresden H. 22, (1977).

/6/ Trachtenbrot, B. A., Wieso können Automaten rechnen?, Berlin 1968.

/7/ Minsky, M. L., Berechnung und Automaten, Moskau 1971 (russ.)

/8/ Sachs, H., Einführung in die Theorie der endlichen Graphen, Leipzig 1971.

eingegangen: 15.9.1978

Anschrift des Verfassers:

Dr.rer.nat. Stefan von Weber
Wilhelm-Pieck-Universität Rostock
Rechenzentrum
DDR - 25 Rostock
Albert-Einstein-Straße 21

Rostock. Math. Kolloq. 10, 123 - 125 (1978)

Martin Weese

Mad families and its classification

Let ω denote the set of natural numbers. $P(\omega)$ denotes the power set of ω. Let $a,b \in P(\omega)$. We write $a \subseteq^* b$ ($a =^* b$) if $b \setminus a$ is finite ($b \setminus a$ and $a \setminus b$ are finite). a and b are almost disjoint if $a \cap b$ is finite. $X \subseteq P(\omega)$ is an almost disjoint family if each $a \in X$ is infinite, the elements of X are pairwise almost disjoint and for each finite subset S of X, $\omega \setminus \bigcup S$ is infinite. X is a mad family if it is a maximal almost disjoint family.

Let $\langle P, \leq \rangle$ be a partially ordered set. Two elements a,b of P are compatible if there is $c \in P$ with $a \leq c$ and $b \leq c$. A subset D of P is dense if for each $a \in P$ there is $b \in D$ with $a \leq b$. If D is a dense subset of P then a subset G of P is a D-generic filter if the following conditions are satisfied:

(i) if $a \in G$ and $b \leq a$ then $b \in G$;
(ii) if $a,b \in G$ then a and b are compatible;
(iii) $D \cap G \neq \emptyset$.

If $\mathcal{D}$ is a family of dense subsets of P and G is D-generic for each $D \in \mathcal{D}$ then G is called $\mathcal{D}$-generic. If $\mathcal{D}$ is a countable set then there exists always a $\mathcal{D}$-generic filter.

The partial order $\langle P, \leq \rangle$ has the countable chain condition (ccc) if any set $A \subseteq P$ of pairwise incompatible elements is countable.

Martin's axiom (MA) is the following statement:

If $\langle P, \leq \rangle$ is a partally ordered set with the ccc and $\mathcal{D}$ is a family of dense subsets with card $\mathcal{D} < 2^{\omega}$ then there is a $\mathcal{D}$-generic filter.

It is well known that Martin's axiom is consistent with the negation of the continuum hypothesis (CH). Martin's axiom implies that each mad family is of cardinality 2^{ω}. Hechler /3/ showed that it is consistent with ZFC that there exists

a mad family of cardinality less than 2^{ω}.

Let X be an almost disjoint family, $a \subseteq \omega$. a is a partitioner of X if for each $b \in X$ either $a \cap b =^* \emptyset$ or $b \subseteq^* a$. We set $X^P = \{a \subseteq \omega : a \text{ is a partitioner of } X\}$. Then X^P is a Boolean algebra under the usual set-theoretic operations. Let $J(X)$ be the ideal of X^P generated by $X \cup \omega$. With $B\{X\}$ we denote the Boolean algebra $X^P/_{J(X)}$.

In /4/ it is shown:

Theorem 1: (MA) Let $\mathfrak{B}$ be a countable Boolean algebra. Then there exists a mad family X with $B\{X\} \cong \mathfrak{B}$.

Let X be an almost disjoint family. We set

$$F(X) = \{a \subseteq \omega : \text{card } \{s \in X : s \subseteq^* a\} = 2^{\omega}\}.$$

In /5/ it is shown:

Theorem 2: (MA) Let U be a nonprincipal ultrafilter on ω. Then there exists a mad family X with $F(X) = U$.

Let X and Y be almost disjoint families. We set

$X \approx Y$ if

(i) for each $a \in X$ there is $b \in Y$ with $a \subseteq^* b$ or $b \subseteq^* a$ and vica versa;

(ii) for each $a \in X$ the set $\{b \in Y : b \subseteq^* a\}$ is finite and vica versa.

We set $X \lhd Y$ if there are an injective function $f : \omega \to \omega$ and $Y' \subseteq Y$ such that $\{f[a] : a \in X\} \approx Y'$. Let $\langle \beta\omega \setminus \omega, \leq \rangle$ be the set of the nonprincipal ultrafilters with the Rudin-Keisler ordering. Using the fact that the continuum hypothesis implies that there are $2^{2^{\omega}}$ minimal elements in $\beta\omega \setminus \omega$ (see Theorem 9.13 in /1/) we get:

Theorem 3: (CH) There are $2^{2^{\omega}}$ mad families incomparable with respect to $\lhd$.

As an application of theorem 3 we get:

Theorem 4: (CH) There are $2^{2^{\omega}}$ superatomic Boolean algebras on ω such that none of them can be embedded into another.

Proof: Choose a set $\mathcal{U}$ of $2^{2^{\omega}}$ ultrafilters corresponding to different minimal elements in $\beta\omega \setminus \omega$. For each $U \in \mathcal{U}$ choose a mad family X_U with $F(X_U) = U$. For each $U \in \mathcal{U}$ let $\mathcal{A}_U$ be the superatomic Boolean algebra on ω generated by $\omega \cup X_U$. Then it is not hard to see that $\{\mathcal{A}_U : U \in \mathcal{U}\}$ is the desired family.

Literature

/1/ Comfort, W. W., Negrepontis, S., The Theory of Ultra filters. Berlin - Heidelberg - New York, Springer 1974.

/2/ Hechler, S., Classifying almost-disjoint families with applications to $\beta N \setminus N$. Israel J. Math. 10 (1971), 413-432.

/3/ Hechler, S., Short complete nested sequences in $\beta N \setminus N$ and small almost disjoint families. General Topology and App. 2 (1972), 139-149.

/4/ Weese, M., Mad families and superatomic Boolean algebras. to appear.

/5/ Weese, M., Mad families and ultrafilter. to appear.

eingegangen: 15.9.1978

Anschrift des Verfassers:

Martin Weese
Sektion Mathematik
Humboldt-Universität
DDR - 108 Berlin
Unter den Linden 6