ISSN 0138 - 3248

Rostocker Mathematisches Kolloquium

Heft 20

WILHELM-PIECK-UNIVERSITÄT ROSTOCK

ROSTOCKER MATHEMATISCHES KOLLOQUIUM

Heft 20

Vorträge der Tagung

Diskrete Mathematik und ihre Anwendungen in der Mathematischen Kybernetik

veranstaltet von der
Mathematischen Gesellschaft der DDR
und der
Sektion Mathematik der
Wilhelm-Pieck-Universität Rostock

vom 30. 03. bis 03. 04. 1981

Teil 2

1982

Wilhelm-Pieck-Universität Rostock
Sektion Mathematik

Herausgeber: Der Rektor der Wilhelm-Pieck-Universität Rostock

Schriftleitung: Prof. Dr. Wolfgang Engel, Direktor der Sektion Mathematik
Prof. Dr. Gerhard Maeß, Schriftleiter
Dr. Klaus-Dieter Drews, Lektor
Dorothea Meyer, Herstellung der Druckvorlage

Sektion Mathematik der
Wilhelm-Pieck-Universität Rostock,
DDR - 2500 Rostock, Universitätsplatz 1

Das Rostocker Mathematische Kolloquium erscheint in der Regel dreimal im Jahr und ist im Rahmen des Schriftentausches über die Universitätsbibliothek, Tauschstelle, DDR - 2500 Rostock, Universitätsplatz 5, zu beziehen.

Veröffentlicht durch die Abt. Wissenschaftspublizistik der
Wilhelm-Pieck-Universität Rostock, DDR - 2500 Rostock
Vogelsang 13/14
Fernruf: 369 577
Leiter: Dipl.-Ges.-Wiss. Bruno Schrage
Genehmigungs-Nr.: C 55 / 82
Druck: Ostsee-Druck Rostock, Werk II Ribnitz

https://doi.org/10.18453/rosdok_id00003826

Inhalt

		Seite
Gube, Wolfgang	Ein lokaler Algorithmus zum Aufbau minimaler Oberdeckungen	5
Vogel, Hans-Jürgen	Eine Beschreibung von Verknüpfungen für partielle Funktionen	21
Demetrovics, J.; Gyepesi, Gy.	Some logical Dependencies in Relational Data Base	33
Wedler, Volker	Invariant Properties and Normalizations of Pictures	53
Wolf, Gottfried	Zur Charakterisierung stabiler Konfigurationen auf binären zellularen Automaten	61
Mikołajczak, Bolesław	On some new NP-complete problems in the theory of finite automata	73
Burkhard, Hans-Dieter	On Fairness in Petri Nets	85
Klette, Reinhard	Time Complexity of Digital Image Processing Problems	97

Rostock. Math. Kolloq. 20, 5 - 19 (1982) 04A20
05A05

Wolfgang Guba

Ein lokaler Algorithmus zum Aufbau minimaler Oberdeckungen

Die vorliegende Arbeit beschäftigt sich mit minimalen Oberdekkungen von endlichen Mengen M. Auf die Betrachtung solcher Oberdeckungen wird man bei einer Reihe bekannter mathematischer Aufgaben geführt. Es können zum Beispiel die Minimisierung Boolescher Funktionen und die Oberdeckung von Graphen durch Kanten genannt werden (vgl. /3/).

Ein möglicher Zugang zu derartigen Problemen wird durch die Theorie der lokalen Algorithmen gegeben. Dabei geht es um die Gewinnung und Abspeicherung von Informationen über Mengen N_j, $N_j \subseteq M$, bezüglich der folgenden Hauptprädikate:

- N_j geht als Element in alle minimalen Oberdeckungen von M ein, und
- N_j ist in keiner minimalen Oberdeckung von M enthalten.

Daneben können weitere Prädikate zur Entscheidung der Hauptprädikate herangezogen werden. Diese dienen dann als Hilfsprädikate. Entsprechend dem lokalen Charakter der Algorithmen werden neue Informationen nur aus einer gewissen Umgebung von N_j abgeleitet.

Besondere Bedeutung hat die Angabe maximaler lokaler Algorithmen. Die Maximalität ist dabei wie folgt charakterisiert: Wenn mit einem lokalen Algorithmus, der die gleichen Parameter besitzt (Prädikatenauswahl, Umgebungsbegriff), eine Information bezüglich eines der Hauptprädikate gewonnen werden kann, so ist dieses auch mit Hilfe des maximalen Algorithmus möglich. In /5/ wurde von Ju. I. Žuravlev die Konstruktion eines maximalen lokalen Algorithmus mit den angegebenen Hauptprädikaten und ohne Hilfsprädikate angegeben. Ziel unserer Betrachtungen ist es, diesen Algorithmus durch Hinzunahme von Hilfsprädikaten zu erweitern. Dadurch können weitere Informationen über die Hauptprädikate gewonnen werden.

Zunächst sollen aber noch einige Grundbegriffe aus der Theorie der lokalen Algorithmen in enger Anlehnung an die Oberdeckungsproblematik geklärt werden. Es sei jedoch betont, daß die Anwendung lokaler Algorithmen nicht nur bei Oberdeckungsaufgaben möglich ist (vgl. /4/).

1. Grundlegende Begriffe und Bezeichnungen

Es sei M eine endliche Menge. Mit $\mathfrak{M} = \{N_1, \ldots, N_n\}$, $N_j \neq \emptyset$, bezeichnen wir ein System, für das $\bigcup_{j=1}^{n} N_j = M$ vorausgesetzt wird. Jede Teilmenge $O(M, \mathfrak{M}) = \{N_{i_1}, \ldots, N_{i_t}\}$ von $\mathfrak{M}$, für die $\bigcup_{j=1}^{t} N_{i_j} = M$ gilt, nennen wir **Oberdeckung** von M mit Elementen aus $\mathfrak{M}$. $O(M, \mathfrak{M})$ heißt unverkürzbar, wenn keine echte Teilmenge von $O(M, \mathfrak{M})$ eine Oberdeckung von M darstellt. Unter einer Wichtung der Elemente aus $\mathfrak{M}$ verstehen wir eine eindeutige Abbildung μ von $\mathfrak{M}$ in die Menge der positiven reellen Zahlen. Diese Abbildung ordnet jedem N_j, $N_j \in \mathfrak{M}$, ein Gewicht $\mu(N_j)$ zu. Auf der Grundlage einer Wichtung μ nennen wir $O(M, \mathfrak{M}) = \{N_{i_1}, \ldots, N_{i_t}\}$ minimale Oberdeckung, wenn $\sum_{j=1}^{t} \mu(N_{i_j})$ minimal bez. aller Oberdeckungen von M ist.

Wie angekündigt, besteht unser Ziel darin, Informationen über folgende zwei Eigenschaften (Hauptprädikate) zu erhalten:

$P_1^0(N_j, \mathfrak{M}, M)$: N_j ist als Element in allen minimalen Oberdeckungen der Menge M mit Elementen aus $\mathfrak{M}$ enthalten, und

$P_2^0(N_j, \mathfrak{M}, M)$: N_j geht in keine minimale Oberdeckung von M mit Elementen aus $\mathfrak{M}$ ein.

Daneben betrachten wir die Eigenschaften $P_1^i(N_j, \mathfrak{M}, M)$, $P_2^i(N_j, \mathfrak{M}, M)$, $i = 1, \ldots, l$, die sich auf weitere Oberdeckungsklassen beziehen. Für $i = 1, \ldots, s$ sei $Q^i(M, \mathfrak{M}) = \{O(M, \mathfrak{M}) \mid O(M, \mathfrak{M})$ ist minimale Oberdeckung und $\mathfrak{M}^i \subseteq O(M, \mathfrak{M})\}$, für $i = s+1, \ldots, l$ sei $Q^i(M, \mathfrak{M}) = \{O(M, \mathfrak{M}) \mid O(M, \mathfrak{M})$ ist unverkürzbare Oberdeckung und $\mathfrak{M}^i \equiv O(M, \mathfrak{M})\}$ ($\mathfrak{M}^i$ ($i=1,\dots,l$) sei jeweils als Teilmenge einer minimalen Oberdeckung vorgegeben) und damit

$P_1^i(N_j, \mathfrak{M}, M)$: N_j geht in alle Oberdeckungen aus $Q^i(M, \mathfrak{M})$ ein,

$P_2^i(N_j, \mathfrak{M}, M)$: N_j geht in keine Oberdeckung aus $Q^i(M, \mathfrak{M})$ ein.

Bezeichnen wir mit $Q^O(M, \mathfrak{M})$ die Klasse aller minimalen Oberdeckungen von M mit Elementen aus $\mathfrak{M}$, so gilt offenbar $Q^O(M, \mathfrak{M}) \supseteq Q^i(M, \mathfrak{M})$ ($i=1,\dots,s$) und $Q^O(M,\mathfrak{M}) \cap Q^i(M,\mathfrak{M}) \neq \emptyset$ ($i = s+1,\dots,l$). Hieraus ergibt sich die Möglichkeit, P_1^i und P_2^i zur Entscheidung von P_1^O und P_2^O heranzuziehen (P_1^i, P_2^i fungieren als Hilfsprädikate).

Jedem N_j aus $\mathfrak{M}$ ordnen wir einen (2l+2)-stelligen Informationsvektor $\tilde{\alpha}_j = (a^O_{1;j}, a^O_{2;j}, a^1_{1;j}, \dots, a^l_{2;j})$, $a^i_{h;j} \in \{0,1,\Delta\}$, zu. Wenn $a^i_{h;j} = 1$ (=0) ist, so folgt daraus "$P^i_h(N_j,\mathfrak{M},M)$ wahr" (bzw. "falsch"). Im Fall $a^i_{h;j} = \Delta$ liegt keine Information über $P^i_h(N_j, \mathfrak{M}, M)$ vor. Ausgehend von einer Menge $\mathfrak{M}^* = \{N_1^{\tilde{\alpha}_1}, \dots, N_n^{\tilde{\alpha}_n}\}$ (die Elemente aus $\mathfrak{M}$ werden mit Informationsvektoren markiert) bestimmen die lokalen Algorithmen neue derartige Mengen, die mehr Informationen enthalten, z. B. $\mathfrak{M}'^* = \{N_1^{\tilde{\beta}_1}, \dots, N_n^{\tilde{\beta}_n}\}$ mit

$\tilde{\beta}_j = (b^O_{1;j}, \dots, b^l_{2;j}) = \varphi^i_h(N_j^{\tilde{\alpha}_j}, S_k(N_j^{\tilde{\alpha}_j}, \mathfrak{M}^*))$ und $\tilde{\alpha}_j \leq \tilde{\beta}_j$, d. h. $b^i_{h;j} = a^i_{h;j}$, falls $a^i_{h;j} \in \{0,1\}$ gilt. Hinter φ^i_h verbergen sich hinreichende Kriterien, die Informationen über N_j bezüglich des Prädikates P^i_h liefern. Neue Informationen über N_j werden nur aus einer Umgebung k-ter Ordnung des Elementes $N_j^{\tilde{\alpha}_j}$ in der Menge $\mathfrak{M}^*$ gewonnen. Es soll dabei folgender Umgebungsbegriff verwendet werden:

$S_1(N_j^{\tilde{\alpha}_j}, \mathfrak{M}^*)$ ist die Menge aller $N_p^{\tilde{\alpha}_p}$ aus $\mathfrak{M}^*$, für die gilt:

a) $N_p \cap N_j \neq \emptyset$ oder b) $N_p \subseteq \bigcup_{q=1}^{d} N_{a_q}$, wobei $N_{a_1}, \ldots, N_{a_d}$ aus $\mathfrak{M}$ sind und der Bedingung a) genügen.

Für $k \geq 2$ ist

$S_k(N_j^{\tilde{\alpha}_j}, \mathfrak{M}^*)$ die Menge aller $N_b^{\tilde{\alpha}_b}$ aus $\mathfrak{M}^*$, für die gilt:

a) Es existiert ein Element $N_t^{\tilde{\alpha}_t}$ aus $S_{k-1}(N_j^{\tilde{\alpha}_j}, \mathfrak{M}^*)$ mit $N_b \cap N_t \neq \emptyset$, oder b) es ist $N_b \subseteq \bigcup_{h=1}^{g} N_{f_h}$, wobei $N_{f_1}, \ldots, N_{f_g}$ aus $\mathfrak{M}$ sind und der Bedingung a) genügen.

Oftmals werden wir auch $S_k(N_j, \mathfrak{M}) = \{N_u | N_u^{\tilde{\alpha}_u} \in S_k(N_j^{\tilde{\alpha}_j}, \mathfrak{M}^*)\}$ verwenden.

2. Ein lokaler Algorithmus für die Prädikate P_1^i, P_2^i $(i = 0,1,2,\ldots,1)$

Es erfolgt nun die Angabe eines lokalen Algorithmus für die angegebene Prädikatenauswahl. Dabei steht als Frage im Mittelpunkt der Betrachtungen, wie man aus der Kenntnis der Umgebung $S_k(N_j^{\tilde{\alpha}_j}, \mathfrak{M}^*)$ neue Informationen über N_j ableiten kann.

2.1. Bilden eines Systems von Teilmengen der Menge M:

Wir setzen $M_{(k-1,j)} = \bigcup_{N_p \in S_{k-1}(N_j, \mathfrak{M})} N_p$, $M_{(k,j)} = \bigcup_{N_q \in S_k(N_j, \mathfrak{M})} N_q$ und $M_{k,k-1,j} = M_{(k,j)} \setminus M_{(k-1,j)}$. Mit Hilfe der Äquivalenzrelation – a ist äquivalent b, wenn $\{N_h | a \in N_h \text{ und } N_h \in S_k(N_j; \mathfrak{M})\} = \{N_g | b \in N_g \text{ und } N_g \in S_k(N_j, \mathfrak{M})\}$ – wird $M_{k,k-1,j}$ in Klassen $K_1, \ldots, K_{q(j,k)}$ zerlegt. Der nächste Schritt besteht dann im Aussondern der Mengen M_y:

$M_y = M_{(k-1,j)} \cup A_y$ (A_y ist Vereinigung gewisser Klassen K_1, $y = 1, \ldots, 2^{q(j,k)}$).

Wir erhalten dadurch schließlich ein System von Teilmengen der Menge M, das wir mit $\mathfrak{M}_j$ bezeichnen.

<u>Beispiel:</u>

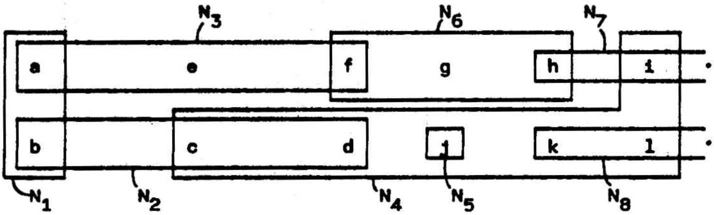


Abb. 1

Mit $\mathfrak{M}^* = \{N_1^{(\Delta,\dots,\Delta)},\dots,N_8^{(\Delta,\dots,\Delta)},\dots\}$ ergeben sich folgende Umgebungen:

$$S_1(N_1^{(\Delta,\dots,\Delta)},\mathfrak{M}^*) = \{N_1^{(\Delta,\dots,\Delta)}, N_2^{(\Delta,\dots,\Delta)}, N_3^{(\Delta,\dots,\Delta)}\},$$

$$S_2(N_1^{(\Delta,\dots,\Delta)},\mathfrak{M}^*) = \{N_1^{(\Delta,\dots,\Delta)}, \quad \dots \quad , N_6^{(\Delta,\dots,\Delta)}\}.$$

Weiterhin erhält man $M_{(1,1)} = \{a,b,c,d,e,f\}$,
$M_{(2,1)} = \{a,b,c,d,e,f,g,\dots,l\}$ und $M_{2,1,1} = \{g,h,i,j,k,l\}$ sowie

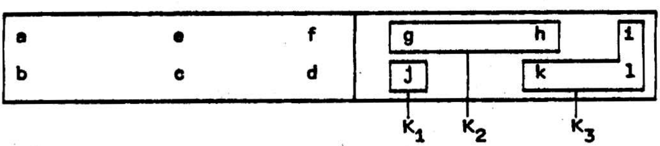


Abb. 2

$\mathfrak{M}_1$: $M_1 = \{a,b,c,d,e,f\}$, $\quad M_2 = \{a,b,c,d,e,f,j\}$,
$M_3 = \{a,b,c,d,e,f,j,g,h\}$, $\quad M_4 = \{a,b,c,d,e,f,k,l,i,j\}$,
$M_5 = \{a,b,c,d,e,f,g,h\}$, $\quad M_6 = \{a,b,c,d,e,f,g,h,k,i,l\}$,
$M_7 = \{a,b,c,d,e,f,k,l,i\}$, $\quad M_8 = \{a,b,c,d,e,f,g,h,i,j,k,l\}$.

Wir wollen nun untersuchen, welche Oberdeckungen von Mengen M_y, $M_y \in \mathfrak{M}_j$, mit Elementen aus $S_k(N_j, \mathfrak{M})$ als Teil einer Oberdeckung aus $Q^i(M, \mathfrak{M})$ in Frage kommen. Im Fall $i \in \{0,1,\ldots,s\}$ können dieses nur minimale Oberdeckungen sein, die (falls $i \neq 0$) $\mathfrak{M}^i \cap S_k(N_j, \mathfrak{M})$ als Teilmenge enthalten. Weiterhin dürfen keine Widersprüche zu bereits vorhandenen Informationen bestehen. Im Fall $i \in \{s+1,\ldots,l\}$ kommen unverkürzbare Oberdeckungen in Frage, die $\mathfrak{M}^i \cap S_k(N_j, \mathfrak{M})$ als Teilmenge enthalten und in keinem Widerspruch zu bereits vorhandenen Informationen stehen. Jede zu einem Widerspruch führende unverkürzbare Oberdeckung enthält $\mathfrak{M}^i \cap S_k(N_j, \mathfrak{M})$ nicht als Teilmenge oder überdeckt eine Menge N_f^i mit (siehe Abb. 3).

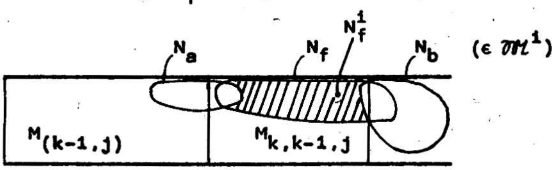


Abb. 3

Da solche Mengen vom Typ N_f und N_b aber nicht in $S_k(N_j, \mathfrak{M})$ enthalten sind, müssen Teilmengen von $M_{k,k-1,j}$ überprüft werden, ob sie für eine Menge vom Typ N_f^i in Frage kommen (und somit aussondernd wirken können). Diese Gedanken werden durch folgende Definitionen präzisiert und führen zum Begriff der i-zulässigen Oberdeckung.

2.2. Bilden von i-zulässigen Oberdeckungen

a) Eine Oberdeckung $O(\tilde{M}_y, \tilde{\mathfrak{M}})$ einer Menge $\tilde{M}_y$ mit Elementen aus $\tilde{\mathfrak{M}}$ nennen wir widersprüchlich mit den Informationen über (P_1^i, P_2^i), wenn folgendes gilt:

(1) Es existiert in $S_k(N_j^{\tilde{\alpha}_j}, \mathfrak{M}^*)$ ein Element $N_f^{(\ldots,1,a^i_{2;f},\ldots)}$ (bzw. $N_h^{(\ldots,a^i_{1;h},1,\ldots)}$) und N_f geht in die Oberdeckung

$O(\tilde{M}_y, \widetilde{\mathfrak{M}})$ nicht ein (bzw. N_h geht in die Oberdeckung ein), und

(2) falls $i \neq 0$ ist, so enthält die Oberdeckung $\mathfrak{M}^i \cap S_k(N_j, \mathfrak{M})$ als Teilmenge.

b) Es seien $A_{s+1} = \{A_1^{s+1}, \ldots, A_{n_{s+1}}^{s+1}\}, \ldots\ldots, A_1 = \{A_1^1, \ldots, A_{n_1}^1\}$ Systeme von Teilmengen der Menge $M_{k,k-1,j}$.

$A = \bigcup_{i=s+1}^{1} A_i$ bezeichnen wir als System von entscheidenden Mengen, wenn bei geeigneter Wichtung der Elemente aus A nachfolgende Bedingungen erfüllt sind.

Für alle $i = 0, 1, \ldots, s$ gilt:

(1) Wenn in $S_k(N_j^{\tilde{\alpha}_j}, \mathfrak{M}^*)$ ein Element $N_f^{(\ldots, 0, a^i_{2;f} \ldots)}$ (bzw. $N_f^{(\ldots, a^i_{1;f}, 0 \ldots)}$) vorkommt, so gibt es in $\mathfrak{M}_j$ eine Menge M_y mit den Eigenschaften a), b) und c):

a) M_y besitzt eine minimale Oberdeckung mit Elementen aus $S_k(N_j, \mathfrak{M}) \cup A$, die (falls $i \neq 0$) $\mathfrak{M}^i \cap S_k(N_j, \mathfrak{M})$ als Teilmenge enthält.

b) N_f ist kein Element dieser Oberdeckung (bzw. N_f ist Element dieser Oberdeckung).

c) Alle minimalen Oberdeckungen von M_y mit Elementen aus $S_k(N_j, \mathfrak{M}) \cup A$ sind nicht widersprüchlich mit den Informationen über (P_1^0, P_2^0) und (P_1^i, P_2^i).

(2) Es gibt mindestens eine Menge M_y aus $\mathfrak{M}_j$ mit den Eigenschaften (1) a) und c).

Für alle $i = s+1, \ldots, l$ gilt:

(3) Wenn in $S_k(N_j^{\tilde{\alpha}_j}, \mathfrak{M}^*)$ ein Element $N_g^{(\ldots, 0, a^i_{2;g}, \ldots)}$ (bzw. $N_g^{(\ldots, a^i_{1;g}, 0 \ldots)}$) vorkommt, so gibt es eine Menge

M_z^i, $M_z^i = M_z \cup \bigcup_{j=1}^{n_i} A_j^i$, $M_z \in \mathfrak{M}_j$, mit den Eigenschaften a), b) und c):

a) M_z^i besitzt eine unverkürzbare Oberdeckung mit Elementen aus $S_k(N_j, \mathfrak{M}) \cup A$, die $(\mathfrak{M}^i \cap S_k(N_j, \mathfrak{M})) \cup A_i$ als Teilmenge enthält.

b) N_g ist kein Element dieser Oberdeckung (bzw. N_g ist Element dieser Oberdeckung).

c) Alle unverkürzbaren Oberdeckungen von M_z^i mit Elementen aus $S_k(N_j, \mathfrak{M}) \cup A$, die $(\mathfrak{M}^i \cap S_k(N_j, \mathfrak{M})) \cup A_i$ als Teilmenge enthalten, sind nicht widersprüchlich mit Informationen über (P_1^i, P_2^i).

(4) Es gibt eine Menge M_x^i, $M_x^i = M_x \cup \bigcup_{j=1}^{n_i} A_j^i$, $M_x \in \mathfrak{M}_j$, mit den Eigenschaften a) und b):

a) M_x^i besitzt eine minimale Oberdeckung mit Elementen aus $S_k(N_j, \mathfrak{M}) \cup A$, die $(\mathfrak{M}^i \cap S_k(N_j, \mathfrak{M})) \cup A_i$ als Teilmenge enthält.

b) Alle minimalen Oberdeckungen von M_x^i mit Elementen aus $S_k(N_j, \mathfrak{M}) \cup A$ sind nicht widersprüchlich mit Informationen über (P_1^0, P_2^0), und alle unverkürzbaren Oberdeckungen, die $(\mathfrak{M}^i \cap S_k(N_j, \mathfrak{M})) \cup A_i$ als Teilmenge enthalten, sind nicht widersprüchlich mit Informationen über (P_1^i, P_2^i).

<u>Definition:</u> Eine Oberdeckung $O(M_y, S_k(N_j, \mathfrak{M}))$ einer Menge M_y aus $\mathfrak{M}_j$ mit Elementen aus $S_k(N_j, \mathfrak{M})$ heißt i - z u l ä s s i g bez. $A = \{A_{s+1}, \ldots, A_l\}$, wenn gilt:

Für $i = 0$:

a) $O(M_y, S_k(N_j, \mathfrak{M}))$ ist minimal, und

b) alle minimalen Oberdeckungen von M_y mit Elementen aus $S_k(N_j, \mathfrak{M}) \cup A$ sind nicht widersprüchlich mit Informationen über (P_1^0, P_2^0).

Für $i = 1,\ldots,s$:

a) $O(M_y, S_k(N_j, \mathfrak{M}))$ ist minimal und enthält $\mathfrak{M}^i \cap S_k(N_j, \mathfrak{M})$ als Teilmenge.

b) Alle minimalen Überdeckungen von M_y mit Elementen aus $S_k(N_j, \mathfrak{M}) \cup A$ sind nicht widersprüchlich mit Informationen über (P_1^0, P_2^0) und, wenn sie $\mathfrak{M}^i \cap S_k(N_j, \mathfrak{M})$ als Teilmenge enthalten, nicht widersprüchlich mit Informationen über (P_1^i, P_2^i).

Für $i = s+1,\ldots,l$:

a) $O(M_y, S_k(N_j, \mathfrak{M}))$ ist unverkürzbar und enthält $\mathfrak{M}^i \cap S_k(N_j, \mathfrak{M})$ als Teilmenge.

b) $O(M_y, S_k(N_j, \mathfrak{M})) \cup A_i$ ist unverkürzbare Überdeckung einer Menge M_y^i. Jede unverkürzbare und mit Informationen über (P_1^i, P_2^i) widersprüchliche Überdeckung von M_y^i mit Elementen aus $S_k(N_j, \mathfrak{M}) \cup A$ enthält A_i nicht als Teilmenge.

Wir sprechen von einer i-zulässigen Überdeckung, wenn die Überdeckung i-zulässig bezüglich eines Systems A ist. Damit ist uns eine Charakterisierung von Überdeckungen gelungen, die als Teil einer Überdeckung aus $Q^i(M, \mathfrak{M})$ auftreten können.

<u>Hilfssatz:</u> Wenn $O(M, \mathfrak{M}) \in Q^i(M, \mathfrak{M})$ gilt, so ist $O(M, \mathfrak{M}) \cap S_k(N_j, \mathfrak{M})$ eine i-zulässige Überdeckung.

<u>Beweis:</u> Wir wollen hier nur auf den Fall $i = 0$ eingehen. Es ist offensichtlich, daß $O(M, \mathfrak{M}) \cap S_k(N_j, \mathfrak{M})$ minimale Überdeckung einer Menge M_y, $M_y \in \mathfrak{M}_j$, ist. Ferner ist $A = \bigcup_{i=s+1}^{l} A_i$,

$$A_i = \left\{ N_h^i \mid N_h^i = N_h \setminus \bigcup_{\substack{N_f \in \mathfrak{M}^i \\ N_f \neq N_h}} N_f,\ N_h \in \mathfrak{M}^i \setminus S_k(N_j, \mathfrak{M}) \right\},$$

immer ein System von entscheidenden Mengen. Wenn M_y nun eine minimale Überdeckung $\tilde{O}(M_y, S_k(N_j, \mathfrak{M}) \cup A)$ besäße, die widersprüchlich zu Informationen über (P_1^0, P_2^0) ist, so könnte man eine minimale Überdeckung von M konstruieren, die ebenfalls im Widerspruch zu bereits vorhandenen Informationen steht:

$(\ddot{U}(M, \mathfrak{M}) \setminus (\ddot{U}(M, \mathfrak{M}) \cap S_k(N_j, \mathfrak{M}))) \cup \tilde{\ddot{U}}(M_y, S_k(N_j, \mathfrak{M}) \cup \tilde{\ddot{A}})$.

$(\tilde{\ddot{U}}(M_y, S_k(N_j, \mathfrak{M}) \cup \tilde{\ddot{A}})$ entsteht aus $\tilde{\ddot{U}}(M_y, S_k(N_j, \mathfrak{M}) \cup A)$, indem eventuell auftretende Elemente N_h^i aus A wieder durch N_h ersetzt werden.) $\ddot{U}(M, \mathfrak{M}) \cap S_k(N_j, \mathfrak{M})$ ist deshalb eine 0-zulässige Überdeckung. Der Hilfssatz ist damit bewiesen.

Folgerung: Wenn N_j in allen i-zulässigen Überdeckungen enthalten ist, so geht N_j auch in alle Überdeckungen aus $Q^i(M, \mathfrak{M})$ ein.

Damit haben wir ein Kriterium erhalten, das 1-Informationen liefert (d. h., in den Informationsvektoren kann gegebenenfalls ein $\triangle$ durch eine 1 ersetzt werden). Bei der Gewinnung von 0-Informationen wollen wir hier nur auf die folgenden Möglichkeiten zurückgreifen:

Aus	P_1^0,	P_2^0,	P_1^a,	P_2^a,	P_1^b,	P_2^b	folgt	P_1^0,	P_2^0,	P_1^a,	P_2^a,	P_1^b,	P_2^b
	w								f		f		f
		w						f		f		f	
			w						f		f		
				w				f		f			
			f					f					
				f					f	f			
					w				f				f
						w		f				f	

$a \in \{1,2,\ldots,s\}$, $b \in \{s+1,\ldots,l\}$, w: wahr, f: falsch.

Wir fassen nun die Ergebnisse zusammen:

2.3. Die Funktionen φ_1^i und φ_2^i $(i=0,1,\ldots,l)$:

$\varphi_1^i(N_j^{\tilde{\alpha}_j}, S_k(N_j^{\tilde{\alpha}_j}, \mathfrak{M}^*))$

$= (\ldots,1,a_{2;j}^i,\ldots)$, wenn $a_{1;j}^i = 1$ oder $a_{1;j}^0 = 1$ ist, oder N_j in alle i-zulässigen Überdeckungen eingeht,

$= (\ldots,0,a^i_{2;j},\ldots)$, wenn $a^i_{1;j} = 0$ oder $a^i_{2;j} = 1$ ist, oder im Falle $i = 0$:

$a^f_{1;j} = 0$ mit $f \in \{1,\ldots,s\}$ oder

$a^f_{2;j} = 1$ mit $f \in \{s+1,\ldots,l\}$,

$= (\ldots,\triangle,a^i_{2;j},\ldots)$ in allen anderen Fällen.

$\varphi^i_2(N_j^{\tilde{\alpha}_j}, S_k(N_j^{\tilde{\alpha}_j}, \mathfrak{M}^*))$

$= (\ldots,a^i_{1;j},1,\ldots)$, wenn $a^i_{2;j} = 1$ oder $a^0_{2;j} = 1$ oder N_j in keiner i-zulässigen Oberdeckung enthalten ist,

$= (\ldots,a^i_{1;j},0,\ldots)$, wenn $a^i_{2;j} = 0$ oder $a^i_{1;j} = 1$ ist, oder im Falle $i = 0$:

$a^f_{2;j} = 0$ mit $f \in \{1,2,\ldots,s\}$ oder

$a^f_{1;j} = 1$ mit $f \in \{s+1,\ldots,l\}$,

$= (\ldots,a^i_{1;j},\triangle,\ldots)$ in allen anderen Fällen.

<u>Satz 1:</u> Die Funktionen φ^i_1 und φ^i_2 $(i=0,\ldots,l)$ liefern keine falschen Informationen.

Der Beweis dieses Satzes ergibt sich im wesentlichen durch Anwenden des Hilfssatzes und der Folgerung.

<u>Satz 2:</u> Wenn $S_1 = S_k(N_j^{\tilde{\alpha}_j}, \mathfrak{M}^*) \leq S_k(N_j^{\tilde{\beta}_j}, \tilde{\mathfrak{M}}^*) = S_2$ gilt (d. h. $S_k(N_j, \mathfrak{M}) = S_k(N_j, \tilde{\mathfrak{M}})$ und jede in den Informationsvektoren S_1 auftretende 1 oder 0 steht auch in den Informationsvektoren von S_2), so folgt aus

$\varphi^i_a(N_j^{\tilde{\alpha}_j}, S_k(N_j^{\tilde{\alpha}_j}, \mathfrak{M}^*)) = (\ldots,c^i_{a;j},\ldots)$; $c^i_{a;j} \in \{0,1\}$,

$\varphi^i_a(N_j^{\tilde{\beta}_j}, S_k(N_j^{\tilde{\beta}_j}, \tilde{\mathfrak{M}}^*)) = (\ldots,c^i_{a;j},\ldots)$ für $i = 0,\ldots,l$ und $a = 1,2$.

Beweis: Wir bezeichnen mit I_1 bzw. I_2 die Menge aller i-zulässigen Oberdeckungen, die mit Hilfe von S_1 bzw. S_2 gebildet werden können. Aus $S_1 \leq S_2$ läßt sich die Beziehung $I_2 \subseteq I_1$ ableiten. Wenn nun N_j in jeder Oberdeckung aus I_1 enthalten ist (bzw. in keiner enthalten ist), so ist N_j auch in jeder Oberdeckung aus I_2 enthalten (bzw. in keiner enthalten). Werden Informationen aus in S_1 bereits vorhandenen Aussagen gewonnen, so lassen sich diese natürlich auch aus S_2 ableiten, womit Satz 2 bewiesen ist.

3. Die Maximalität des angegebenen Algorithmus

Der folgende Satz rechtfertigt die Aussage, daß der angegebene lokale Algorithmus hinsichtlich der 1-Informationen bestmöglich ist. Es sei hier nur bemerkt, daß man durch ähnliche Betrachtungen, wie sie in /5/ gemacht wurden, auch bezüglich der O-Informationen zu bestmöglichen Ergebnissen kommen kann.

Satz 3: Wenn ein lokaler Algorithmus, der auch für die angegebene Prädikatauswahl und mit dem gleichen Umgebungsbegriff arbeitet, eine 1-Information liefert (d. h., in einem Informationsvektor wird $\triangle$ durch eine 1 ersetzt), so erhalten wir diese Information auch mit Hilfe des angegebenen Algorithmus.

Beweis: Angenommen unser Algorithmus hat nicht die angeführte Eigenschaft. In diesem Fall gibt es eine Funktion $\tilde{\varphi}_1^0$ (eines anderen lokalen Algorithmus) und eine Umgebung $S_k(N_j^{\tilde{\alpha}_j}, \mathfrak{M}^*)$, so daß gilt:

$$\varphi_1^0(N_j^{\tilde{\alpha}_j}, S_k(N_j^{\tilde{\alpha}_j}, \mathfrak{M}^*)) = (\triangle, \ldots\ldots) \text{ und} \qquad (a)$$

$$\tilde{\varphi}_1^0(N_j^{\tilde{\alpha}_j}, S_k(N_j^{\tilde{\alpha}_j}, \mathfrak{M}^*)) = (1, \ldots\ldots). \qquad (b)$$

(Wir gehen nur auf diese eine Möglichkeit ein. Die anderen Fälle werden analog ausgeschlossen.)

Aus (a) leiten wir ab, daß N_j nicht in alle O-zulässigen Oberdeckungen eingeht. Eine solche Oberdeckung sei $O(M_0, S_k(N_j, \mathfrak{M}))$.

$A = \bigcup_{i=s+1}^{l} A_i$ $(A_i = \{A_1^i, \ldots, A_{n_i}^i\})$ sei ein System von entscheidenden Mengen, das die O-zulässigkeit dieser Überdeckung bewirkt. Ferner seien

$M_{i;1}, \ldots, M_{i;m_i}$ $(M_{i;t} \in \mathfrak{M}_j,\ i = 0,1,\ldots,s)$, (1), (2),

$M_{i;1}^i, \ldots, M_{i;o_i}^i$ $(M_{i;t}^i = M_{i;t} \cup \bigcup_{a=1}^{n_i} A_a^i,\ M_{i;t} \in \mathfrak{M}_j,$
$i = s+1, \ldots, l)$, (3),

$M_x^{s+1}, \ldots, M_x^l$ $(M_x^t = M_x \cup \bigcup_{a=1}^{n_i} A_a^i,\ M_x \in \mathfrak{M}_j,\ t = s+1, \ldots, l)$, (4)

solche Mengen, die garantieren, daß A ein System von entscheidenden Mengen ist. (Man vergleiche hierzu die Punkte (1), (2), (3) und (4) bei der Festlegung von A.)

Darauf aufbauend konstruieren wir folgende Überdeckungsaufgabe:

$B_i = \{i\}$ $(i = 0,1,\ldots,s)$,

$B_i = \{a_{i;1}^i, \ldots, a_{i;o_i}^i, a_x^i\}$ $(i = s+1, \ldots, l)$,

$B = \bigcup_{i=0}^{l} B_i$,

$\underline{\tilde{M} = M_{(k,j)} \cup \{a,b\} \cup B},$

$X_0 = (M_{(k,j)} \setminus M_0) \cup \{a,b\} \cup (B \setminus \{0\})$,

$i=0,\ldots,s,\ t=1,\ldots,m_i$: $X_{i;t} = (M_{(k,j)} \setminus M_{i;t}) \cup \{a,b\} \cup (B \setminus \{0,i\})$,

$i=s+1,\ldots,l,\ t=1,\ldots,o_i$: $X_{i;t}^i = (M_{(k,j)} \setminus M_{i;t}^i) \cup \{a,b\} \cup (B \setminus \{a_{i;t}^i\})$,

$i=s+1,\ldots,l$: $X_x^i = (M_{(k,j)} \setminus M_x^i) \cup \{a,b\} \cup (B \setminus \{0, a_x^i\})$,

$X = \{X_0, \ldots, X_x^l\}$,

$i=s+1,\ldots,l,\ t=1,\ldots,n_i$: $Y_t^i = A_t^i \cup \{b\}$,

$i=s+1,\ldots,l$: $Y_i = \{Y_1^i, \ldots, Y_{n_i}^i\}$,

$Y = \{Y_{s+1}, \ldots, Y_l\}$,

$\underline{\tilde{\mathfrak{M}} = S_k(N_j, \mathfrak{M}) \cup X \cup Y \cup \{B_0, \ldots, B_l\}}.$

Bei entsprechender Wichtung der Elemente aus X, Y und $\{B_0,\ldots,B_l\}$ (die Wichtung der Elemente aus $S_k(N_j,\mathfrak{M})$ wird beibehalten) erhalten wir $\tilde{\mathfrak{M}}^i = (S_k(N_j,\mathfrak{M}) \cap \mathfrak{M}^i) \cup \{B_i\}$ $(i = 1,\ldots,s)$ und $\tilde{\mathfrak{M}}^i = (S_k(N_j,\mathfrak{M}) \cap \mathfrak{M}^i) \cup \{B_i\} \cup Y_i$ $(i = s+1,\ldots,l)$ als Teile von minimalen Oberdeckungen der Menge $\tilde{M}$ mit Elementen aus $\tilde{\mathfrak{M}}$. Damit ergibt sich aber auch $Q^0(\tilde{M},\tilde{\mathfrak{M}}) \supseteq Q^i(\tilde{M},\tilde{\mathfrak{M}})$ $(i = 1,\ldots,s)$ und $Q^0(\tilde{M},\tilde{\mathfrak{M}}) \cap Q^i(\tilde{M},\tilde{\mathfrak{M}}) \neq \emptyset$ $(i = s+1,\ldots,l)$. Weiterhin wurde die neue Oberdeckungsaufgabe so konstruiert, daß

$$\mathfrak{M}^* = S_k(N_j^{\tilde{\alpha}_j},\mathfrak{M}^*) \cup \{D^{(\Delta,\ldots,\Delta)} \mid D \in (X \cup Y \cup \{B_0,\ldots,B_l\})\}$$

keine falschen Informationen enthält. Da nun

$S_k(N_j^{\tilde{\alpha}_j},\tilde{\mathfrak{M}}^*) = S_k(N_j^{\tilde{\alpha}_j},\mathfrak{M}^*)$ ist, gilt

$$\tilde{\varphi}_1^0(N_j^{\tilde{\alpha}_j},S_k(N_j^{\tilde{\alpha}_j},\tilde{\mathfrak{M}}^*)) = \tilde{\varphi}_1^0(N_j^{\tilde{\alpha}_j},S_k(N_j^{\tilde{\alpha}_j},\mathfrak{M}^*)) = (1,\ldots\ldots).$$

Das ist aber eine falsche Information, denn $O(M_0,S_k(N_j,\mathfrak{M}) \cup \{X_0\} \cup \{\{0\}\}$ ist eine minimale Oberdeckung von $\tilde{M}$ mit Elementen aus $\tilde{\mathfrak{M}}$, in der N_j nicht enthalten ist.

Literatur

/1/ Guba, W.: Ein maximaler lokaler Algorithmus für Klassen unverkürzbarer Oberdeckungen. Rostock. Math. Kolloq. 3, 57 - 68 (1977)

/2/ Guba, W.: Maximale lokale Algorithmen zur Konstruktion minimaler Oberdeckungen. Dissertation A, Wilhelm-Pieck-Universität Rostock 1977

/3/ Jablonskij, S. V.: Funkcional'nye postroenija v k-značnoj logike. Trudy Mat. Inst. Steklov. 51, 5 - 142 (1958)

/4/ Žuravlev, Ju. I.: Lokal'nye algoritmy vyčislenija informacii I. Kibernetika (Kiev) 1965, No. 1, 12 - 19

/5/ Žuravlev, Ju. I.: Lokal'nye algoritmy vyčislenija informacii II. Kibernetika (Kiev) 1966, No. 2, 1 - 11

eingegangen: 30. 03. 1981

Anschrift des Verfassers:

Dr. rer. nat. W. Guba
Wilhelm-Pieck-Universität Rostock
Sektion Mathematik
Universitätsplatz 1
DDR-2500 Rostock

Rostock. Math. Kolloq. 20, 21 - 32 (1982) 08A05 08A25

Hans-Jürgen Vogel

Eine Beschreibung von Verknüpfungen für partielle Funktionen

1. Vorbemerkung

In dem vorliegenden Artikel werden beliebige partielle Funktionen zwischen Mengen betrachtet, wobei der Fall endlicher Mengen für konkrete Anwendungen von besonderem Interesse ist. Die zwischen den Funktionen erklärten Verknüpfungen werden als Operationen für partielle Funktionen angesehen und mit Hilfe der Verkettung, des kartesischen Produktes und gewisser Standardfunktionen (konstante Operationen) ausgedrückt. Der axiomatische Aufbau dieses Anliegens führt auf spezielle angereicherte monoidale Kategorien (Eilenberg-Kelly /1/), sogenannte diagonal-halbterminal-symmetrische Kategorien (kurz: dht-symmetrische Kategorien).

Für den Definitionsbereich einer Funktion f wird kurz D(f) und für den Wertebereich W(f) geschrieben. Die Verkettung fg zweier Funktionen ist durch fg(a) = g(f(a)) gegeben und das kartesische Produkt $f \otimes g$ durch $(f \otimes g)(a,b) = (f(a), g(b))$. Weiterhin sei für ein Mengensystem M mit M^* die Menge bezeichnet, die aus allen kanonisch beklammerten kartesischen Produkten $(\ldots(A_1 \otimes A_2) \otimes \ldots) \otimes A_n$ von endlich vielen Mengen aus M, den Mengen von M selbst und der Menge $I = \{\emptyset\}$ besteht. Schließlich sei $1_A: A \to A$ die identische Funktion von A auf A.

2. Die Kategorie Par

Die Objektklasse der Kategorie Par besteht aus allen Mengen, die Morphismenklasse aus allen partiellen Funktionen zwischen Mengen. Dabei bedeutet die Schreibweise $f: A \to B$ keineswegs D(f) = A bzw. W(f) = B. Die Menge $I = \{\emptyset\}$ sei eine ausgezeichnete einelementige Menge und O die leere Menge $\emptyset$. In der Kategorie Par läßt sich neben der Verkettung von Funktionen (Morphismenkomposition) auch das kartesische Produkt für Mengen

und Funktionen in der üblichen Weise einführen. Das kartesische Produkt ist nicht strikt assoziativ, jedoch sind die Mengen $A \otimes (B \otimes C)$ und $(A \otimes B) \otimes C$ zueinander isomorph vermöge der Zuordnung $(a,(b,c)) \mapsto ((a,b),c)$.
Bezeichnet man diese Bijektion mit

$$a_{A,B,C}: A \otimes (B \otimes C) \longrightarrow (A \otimes B) \otimes C,$$

dann überzeugt man sich leicht für Funktionen $f: A \to A'$, $g: B \to B'$ und $h: C \to C'$ von der Richtigkeit der Gleichung

(a1) $\quad a_{A,B,C}((f \otimes g) \otimes h) = (f \otimes (g \otimes h))a_{A',B',C'}$,

wobei für Funktionen $u: X \to Y$, $v: X' \to Y'$ die Gleichung $u = v$ genau dann zutrifft, wenn $X = X'$, $Y = Y'$, $D(u) = D(v)$ und $u(z) = v(z)$ für alle $z \in D(u)$ gilt.
Zwischen den Mengen $A \otimes B$ und $B \otimes A$ vermittelt die Zuordnung $(a,b) \longmapsto (b,a)$ stets eine Bijektion $c_{A,B}: A \otimes B \to B \otimes A$, die mit beliebigen Funktionen $f: A \to A'$, $g: B \to B'$ und der entsprechenden Bijektion $c_{A',B'}$ das folgende Diagramm kommutativ schließt:

(c1)
$$\begin{array}{ccc} A \otimes B & \xrightarrow{\quad c_{A,B} \quad} & B \otimes A \\ {\scriptstyle f \otimes g}\downarrow & (=) & \downarrow{\scriptstyle g \otimes f} \\ A' \otimes B' & \xrightarrow[\quad c_{A',B'} \quad]{} & B' \otimes A' \end{array}.$$

Weiterhin existieren für jede Menge A die Bijektionen

$$r_A: A \otimes I \longmapsto A \quad ((a,\emptyset) \longmapsto a),$$
$$l_A: I \otimes A \longrightarrow A \quad ((\emptyset,a) \longmapsto a),$$

die mit jeder Funktion $f: A \to A'$ folgende Diagramme kommutativ schließen:

(rl1)
$$\begin{array}{ccc} A \otimes I & \xrightarrow{\quad r_A \quad} & A \\ {\scriptstyle f \otimes 1_I}\downarrow & (=) & \downarrow{\scriptstyle f} \\ A' \otimes I & \xrightarrow[\quad r_{A'} \quad]{} & A' \end{array}, \qquad \begin{array}{ccc} I \otimes A & \xrightarrow{\quad l_A \quad} & A \\ {\scriptstyle 1_I \otimes f}\downarrow & (=) & \downarrow{\scriptstyle f} \\ I \otimes A' & \xrightarrow[\quad l_{A'} \quad]{} & A' \end{array}.$$

Damit erhalten wir

<u>Satz 1:</u> In der Kategorie <u>Par</u> mit dem üblichen kartesischen Produkt für Mengen und Funktionen bilden die Familien

$a = (a_{A,B,C})_{A,B,C \in |\underline{Par}|}$, $c = (c_{A,B})_{A,B \in |\underline{Par}|}$, $r = (r_A)_{A \in |\underline{Par}|}$

und $l = (l_A)_{A \in |\underline{Par}|}$ natürliche Isomorphismen für die Funktoren

$Z: \underline{Par}^3 \to \underline{Par}$ $((A,(B,C)) \mapsto A \otimes (B \otimes C),\ (f,(g,h)) \mapsto f \otimes (g \otimes h))$

und

$\bar{Z}: \underline{Par}^3 \to \underline{Par}$ $((A,(B,C)) \mapsto (A \otimes B) \otimes C,\ (f,(g,h)) \mapsto (f \otimes g) \otimes h)$

bzw.

$P: \underline{Par}^2 \to \underline{Par}$ $((A,B) \mapsto A \otimes B,\ (f,g) \mapsto f \otimes g)$

und

$\bar{P}: \underline{Par}^2 \to \underline{Par}$ $((A,B) \mapsto B \otimes A,\ (f,g) \mapsto g \otimes f)$

bzw.

$R: \underline{Par} \to \underline{Par}$ $(A \mapsto A \otimes I,\ f \mapsto f \otimes 1_I)$ und $Id_{\underline{Par}}$

bzw.

$L: \underline{Par} \to \underline{Par}$ $(A \mapsto I \otimes A,\ f \mapsto 1_I \otimes f)$ und $Id_{\underline{Par}}$. ■

Zu jeder Menge A existieren schließlich stets die beiden totalen Funktionen

$d_A: A \to A \otimes A$ $(a \mapsto (a,a))$

und

$t_A: A \to I$ $(a \mapsto \emptyset)$.

Für jede Funktion $f: A \to A'$ gilt $fd_{A'} = d_A(f \otimes f)$, d. h., das folgende Diagramm ist kommutativ:

(d)
$$\begin{array}{ccc} A & \xrightarrow{\quad d_A \quad} & A \otimes A \\ f \downarrow & (=) & \downarrow f \otimes f \\ A' & \xrightarrow[\quad d_{A'} \quad]{} & A' \otimes A' \end{array}$$

Das liefert in Ergänzung zu den obigen Feststellungen den

<u>Satz 2:</u> Die Familie $d = (d_a)_{A \in |\underline{Par}|}$ ist eine natürliche Transformation zwischen den Funktoren $Id_{\underline{Par}}$ und

$D: \underline{Par} \to \underline{Par}$ $(A \to A \otimes A,\ f \to f \otimes f)$. ■

Die Familie $t = (t_A)_{A \in |\underline{Par}|}$ ist dagegen keine natürliche Transformation für geeignete Funktoren, denn es gilt

<u>Satz 3:</u> Eine Funktion $f\colon A \to A'$ ist genau dann total, wenn $ft_{A'} = t_A$ zutrifft.

<u>Beweis:</u> Ist $f\colon A \to A'$ eine totale Funktion, dann gilt $D(f) = A$. Folglich ergibt sich für alle Elemente a aus A die Beziehung $ft_{A'}(a) = t_{A'}(f(a)) = \emptyset = t_A(a)$. Ist umgekehrt $ft_{A'} = t_A$, dann erhält man $D(t_A) = A = D(ft_{A'})$, woraus unmittelbar $D(f) = A$ folgt. ■

In der Kategorie <u>Par</u> lassen sich weiterhin für alle Mengen A,B, C,D und alle Funktionen f,f',g,g' folgende Eigenschaften leicht nachweisen:

(F0) $f\colon A \to B,\ f'\colon A' \to B' \Longrightarrow f \otimes f'\colon A \otimes A' \to B \otimes B'$,

(F1) $1_A \otimes 1_B = 1_{A \otimes B}$,

(F2) $(f \otimes f')(g \otimes g') = fg \otimes f'g'$
$(f\colon A \to B,\ g\colon B \to C,\ f'\colon A' \to B',\ g'\colon B' \to C')$,

(a2) $a_{A,B,C \otimes D}\, a_{A \otimes B,C,D}$
$= (a_{A,B,C} \otimes 1_D)(1_A \otimes a_{B,C,D})a_{A,B \otimes C,D}$,

(a3) $d_A(d_A \otimes 1_A) = d_A(1_A \otimes d_A)a_{A,A,A}$,

(rl2) $a_{A,I,B}(r_A \otimes 1_B) = 1_A \otimes l_B$,

(rl3) $a_{I,A,B}(l_A \otimes 1_B) = l_{A \otimes B}$,

(rl4) $a_{A,B,I}\, r_{A \otimes B} = 1_A \otimes r_B$,

(rl5) $r_I = l_I$,

(rl6) $r_A = c_{A,I}\, l_A$,

(c2) $c_{A,B}\, c_{B,A} = 1_{A \otimes B}$,

(c3) $a_{A,B,C}\, c_{A \otimes B,C}\, a_{C,A,B} = (1_A \otimes c_{B,C})a_{A,C,B}(c_{A,C} \otimes 1_B)$,

(c4) $d_A = d_A c_{A,A}$,

(ht1) $ft_{A'} = t_A \quad (f\colon A \to A',\ D(f) = A)$, (ht2) $a_{A,B,C}t_{(A\otimes B)\otimes C} = t_{A\otimes (B\otimes C)}$,

$c_{A,B}t_{B\otimes A} = t_{A\otimes B}$,

$r_A t_A = t_{A\otimes I}$,

$l_A t_A = t_{I\otimes A}$,

$d_A t_{A\otimes A} = t_A$,

(ht3) $t_I = 1_I$,

(ht4) $d_A(1_A \otimes t_A)r_A = d_A(t_A \otimes 1_A)l_A = 1_A$,

(ht5) $d_{A\otimes B}((1_A \otimes t_B)r_A \otimes (t_A \otimes 1_B)l_B) = 1_{A\otimes B}$,

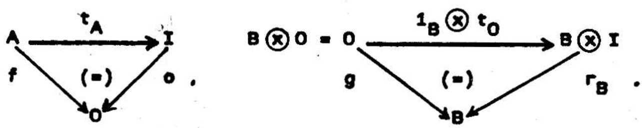
(ht6) $r_A d_A(1_A \otimes t_A) = 1_{A\otimes I}$, $l_A d_A(t_A \otimes 1_A) = 1_{I\otimes A}$,

(ht7) $(t_A \otimes t_B)t_{I\otimes I} = t_{A\otimes B}$.

Da es aus einer beliebigen Menge A nur eine einzige Funktion in die leere Menge O und umgekehrt gibt, ist O Nullobjekt in der Kategorie <u>Par</u>, und es gilt

(O1) $O\otimes A = A\otimes O = O$.

Mit der leeren Funktion $o\colon I\to O$ kommutieren für $f\colon A\to O$ und $g\colon O\to B$ folgende Diagramme:

(O2)


Die bisher zusammengestellten Eigenschaften der Kategorie <u>Par</u> bilden die Grundlage für den axiomatischen Aufbau der Theorie der dht-symmetrischen Kategorien (Hoehnke /4/, Schreckenberger /6/).

<u>Definition:</u> Eine Folge ($\underline{C}$, $\otimes$, I, a, r, l, c, d, t, o) heißt genau dann eine <u>dht-symmetrische Kategorie</u>, wenn $\underline{C}$ eine Kategorie, $\otimes$ ein Bifunktor ((FO) - (F2)) und I ein ausgezeichnetes Objekt in $\underline{C}$ ist, $a = (a_{A,B,C})_{A,B,C\in|\underline{C}|}$, $r = (r_A)_{A\in|\underline{C}|}$, $l = (l_A)_{A\in|\underline{C}|}$, $c = (c_{A,B})_{A,B\in|\underline{C}|}$ Familien von Isomorphismen in $\underline{C}$,

$d = (d_A)_{A\in|\underline{C}|}$, $t = (t_A)_{A\in|\underline{C}|}$ Familien von Morphismen in $\underline{C}$

sind und $o: I \to O$ ein ausgezeichneter Morphismus in $\underline{C}$ ist, so daß die Eigenschaften (a1), (a2), (rl1),...,(rl5), (c1),....,(c3), (ht1),....,(ht7), (O1) und (O2) erfüllt sind.

Dabei ist zu beachten, daß die Bedingung (ht1) in einer beliebigen Kategorie $\underline{C}$ bedeutet, daß für jede Coretraktion $f: A \to A'$ die angegebene Gleichung erfüllt ist.
Die angegebenen Axiome sind nicht unabhängig voneinander, lassen sich aber in dieser Form bequem handhaben.

3. Diagonalen und Projektionen

Wir vereinbaren folgende Symbolik:

$$\underset{i=1}{\overset{0}{\times}} A_i = I, \quad \underset{i=1}{\overset{1}{\times}} A_i = A_1, \quad \underset{i=1}{\overset{n+1}{\times}} A_i = \underset{i=1}{\overset{n}{\times}} A_i \otimes A_{n+1};$$

$$\underset{i=1}{\overset{0}{\times}} f_i = 1_I, \quad \underset{i=1}{\overset{1}{\times}} f_i = f_1, \quad \underset{i=1}{\overset{n+1}{\times}} f_i = \underset{i=1}{\overset{n}{\times}} f_i \otimes f_{n+1}.$$

Insbesondere schreiben wir auch $\underset{i=1}{\overset{n}{\times}} A = A^n$ bzw. $\underset{i=1}{\overset{n}{\times}} f = f^n$.

Die dadurch bestimmte Beklammerung mehrfacher kartesischer Produkte wird <u>kanonisch</u> genannt:

$$\underset{i=1}{\overset{n}{\times}} A_i = (\ldots(A_1 \otimes A_2) \otimes \ldots) \otimes A_n.$$

In der Kategorie $\underline{Par}$ definieren wir für jedes Objekt A die <u>n-fache</u> Diagonale ($n \in N$) gemäß

$$d_A^0 = t_A, \quad d_A^1 = 1_A, \quad d_A^{n+1} = d_A(d_A^n \otimes 1_A): A \to A^{n+1}$$

und für $A = \underset{i=1}{\overset{n}{\times}} A_i$ die Projektionen $p_j^A: A \to A_j$ durch

$$p_1^A = 1_A \text{ für } A = A_1, \quad p_j^{A \otimes A_{n+1}} = (p_j^A \otimes t_{A_{n+1}}) r_{A_j} \text{ für } j \leq n$$

und $p_{n+1}^{A \otimes A_{n+1}} = (p_n^A t_{A_n} \otimes 1_{A_{n+1}}) 1_{A_{n+1}}$.

Die Funktionen p_j^A sind ebenso wie die Diagonalen d_A^{n+1} totale Funktionen und stellen die Projektionen im üblichen Sinn dar. Die Definitionen für die Diagonalen und Projektionen gestatten es, die im folgenden Lemma formulierten Regeln induktiv zu beweisen.

<u>Lemma:</u> a) Für jede Funktion $f\colon A \rightarrow B$ und alle natürlichen Zahlen n gilt $fd_B^n = d_A^n f^n$.

b) Für alle natürlichen Zahlen $n \geq 1$ und j mit $1 \leq j \leq n$ gilt

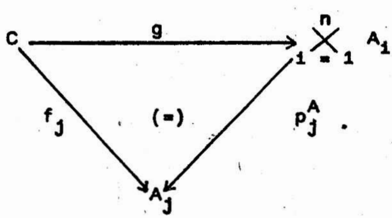
$d_A^n p_j^{A^n} = 1_A$. ■

Die für eine Funktion p_j^A benutzte Bezeichnung Projektion von A auf die Komponente A_j wird auch dadurch gerechtfertigt, daß die Folge $(\bigtimes_{i=1}^{n} A_i,\ (p_j^A\colon A \rightarrow A_j \mid 1 \leq j \leq n))$ ein kategorientheoretisches Produkt der Objekte A_j $(1 \leq j \leq n)$ in der Teilkategorie der totalen Funktionen von <u>Par</u> ist. Dabei ist die zu einer Familie $(f_i\colon C \rightarrow A_i \mid 1 \leq i \leq n)$ von Funktionen gehörende Funktion

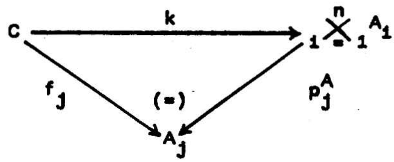
$$g\colon C \rightarrow \bigtimes_{i=1}^{n} A_i \text{ durch } g = d_C^n(\bigtimes_{i=1}^{n} f_i)$$
$$= d_C^n((\ldots(f_1 \otimes f_2) \otimes \ldots) \otimes f_n)$$

gegeben, d. h., das folgende Diagramm ist für jede natürliche Zahl j mit $1 \leq j \leq n$ kommutativ:

(✱)



Die Einzigkeitsaussage für die Funktion g ist nicht auf die Teilkategorie der totalen Funktionen von <u>Par</u> beschränkt, denn wenn eine Funktion $k\colon C \rightarrow A$ aus <u>Par</u> das Diagramm



mit jeder partiellen Funktion einer Familie $(f_j)_{1 \leq j \leq n}$ von Funktionen aus <u>Par</u> kommutativ schließt, dann folgt

$k = d_C^n(\mathop{\times}_{i=1}^{n} f_i)$ (vgl. /8/).

Da die Bedingung $f_j = gp_j^A$ für partielle Funktionen die Übereinstimmung der entsprechenden Definitionsbereiche bedeutet und die Projektionen totale Funktionen sind, erhält man $D(g) = D(f_j)$ mit $1 \leq j \leq n$. Für die Existenz einer Funktion g mit der obigen Eigenschaft ist demzufolge die Übereinstimmung der Definitionsbereiche aller Funktionen f_j $(1 \leq j \leq n)$ notwendig. Diese Bedingung ist offenbar auch hinreichend dafür, daß $g = d_C^n(\mathop{\times}_{i=1}^{n} f_i)$ die Diagramme (*) mit $1 \leq j \leq n$ kommutativ schließt.

4. Superposition, Komposition und Stellentransformation

Der Begriff Superposition wird im Sinne von Hoehnke /3/ benutzt, während Komposition und Stellentransformation in der Begriffsbildung von Wille /7/ verwendet werden.

<u>Satz 4:</u> Es seien $f: \mathop{\times}_{i=1}^{n} A_i \to C$ und $g_i: \mathop{\times}_{j=1}^{m} B_j \to A_i$ $(1 \leq i \leq n)$ partielle Funktionen. Dann ist die partielle Funktion

$$d_B^n(\mathop{\times}_{i=1}^{n} g_i)f: B \to C \quad (B = \mathop{\times}_{j=1}^{m} B_j)$$

genau die durch Superposition aus $f, g_1, \ldots, g_n$ gebildete Funktion $S_m^{n+1}(f, g_1, g_2, \ldots, g_n)$.

Die partielle Funktion $d_D^n(\mathop{\times}_{k=1}^{n} q_k)f: {}^1D \to C$.

mit ${}^{i}D = \bigtimes_{h=1}^{n+m-1} D_h$ und $D_h = A_h$ für $1 \leq h \leq i-1$, $D_h = B_{h-i+1}$ für $i \leq h \leq i+m-1$, $D_h = A_{h-m+1}$ für $i+m \leq h \leq n+m-1$,

$q_k = p_k^D$ für $1 \leq k \leq i-1$, $q_i = d_D^m(\bigtimes_{l=1}^{m} p_{i+l-1}^D)g_i$,

$q_k = p_{k+m-1}^D$ für $i+1 \leq k \leq n$

ist für jedes i ($1 \leq i \leq n$) die durch Einsetzen der Funktion g_i in f an der i-ten Stelle gewonnene partielle Funktion $f(g_i)$.

Schließlich erhalten wir für eine Funktion v von der Menge $\{1,2,\ldots,n\}$ auf die Menge $\{1,2,\ldots,m\}$ und eine Auswahl W von m verschiedenen Mengen $A_1', \ldots, A_m'$ aus dem Mengensystem $\{A_1,\ldots,A_n\}$ durch

$$d_{A'}^n(\bigtimes_{i=1}^{n} p_{v(i)}^{A'})f\colon A' \rightarrow C \quad (A' = \bigtimes_{j=1}^{m} A_j')$$

die durch Stellentransformation aus der Funktion f mit der Funktion v und der Auswahl W gebildete partielle Funktion f_v^W.

<u>Beweis:</u> Es sei $(b_1,b_2,\ldots,b_m)$ (Kurzschreibweise für $((\ldots(b_1,b_2),\ldots),b_m))$ im Vorbereich jeder Funktion g_i und $(g_1(b_1,\ldots,b_m),\ldots,g_n(b_1,\ldots,b_m))$ in $D(f)$ enthalten. Dann gilt

$$\begin{aligned}(d_B^n(\bigtimes_{i=1}^{n} g_i)f)(b_1,\ldots,b_m) &= ((\bigtimes_{i=1}^{n} g_i)f)((b_1,\ldots,b_m),\ldots,(b_1,\ldots,b_m)) \\ &= f(g_1(b_1,\ldots,b_m),\ldots,g_n(b_1,\ldots,b_m)) \\ &= (S_m^{n+1}(f,g_1,\ldots,g_n))(b_1,\ldots,b_m).\end{aligned}$$

Entsprechend erhält man für $(b_1,\ldots,b_m)$ aus dem Vorbereich der Funktion g_i mit $(a_1,\ldots,a_{i-1},g_i(b_1,\ldots,b_m),a_{i+1},\ldots,a_n) \in D(f)$

$$\begin{aligned}(d_D^n(\bigtimes_{k=1}^{n} q_k)f)(a_1,\ldots,a_{i-1},g_i(b_1,\ldots,b_m),a_{i+1},\ldots,a_n) \\ = f(a_1,\ldots,a_{i-1},g_i(b_1,\ldots,b_m),a_{i+1},\ldots,a_n).\end{aligned}$$

Für ein Element $(a_1',\ldots,a_m') \in \bigtimes_{j=1}^{m} A_j'$

mit $(a'_{v(1)},\dots,a'_{v(n)}) \in D(f)$ gilt offenbar

$$(d^n_{A'}.(\overset{n}{\underset{i=1}{\times}} p^{A'}_{v(i)})f)(a'_1,\dots,a'_m) = f^W_v(a'_1,\dots,a'_m).$$

Aus dem Satz 4 ergeben sich die anschließenden Folgerungen.
Im Falle $n = 0$, $m \neq 0$ erhält man durch Superposition die konstante m-stellige Funktion

$$d^0_B(\overset{0}{\underset{i=1}{\times}} g_i)f = t_B 1_I f = t_B f: \overset{m}{\underset{j=1}{\times}} B_j \to C\ ((b_1,\dots,b_m) \mapsto f(\emptyset)),$$

wenn f die totale nullstellige Funktion ist. Sonst entsteht die leere m-stellige Funktion aus B in C.
Bei $m = 0$, $n \neq 0$ erhält man die nullstellige Funktion

$d^n_I(\overset{n}{\underset{i=1}{\times}} g_i)f: I \to C$, die entweder ein Element in C auswählt, nämlich dann, wenn alle $g_i: I \to A_i$ totale Funktionen sind und $(g_1(\emptyset),\dots,g_n(\emptyset)) \in D(f)$ gilt, oder sonst leer ist.
Schließlich ergibt sich bei $m = n = 0$

$$d^0_I(\overset{0}{\underset{i=1}{\times}} g_i)f = 1_I 1_I f = f.$$

Die in Satz 4 angegebenen Bildungsvorschriften zeigen, daß man Komposition und Stellentransformation durch die Superposition ausdrücken kann. Umgekehrt erhält man durch schrittweise Anwendung der Komposition von f mit den Funktionen $g_1, \dots, g_n$ und anschließende geeignete Stellentransformation die Superposition von f mit $g_1, \dots, g_n$.

Da sich die Superposition von Funktionen mit Hilfe der Elementarfunktionen a,r,l,c,g,t, der Verkettung und der Kreuzproduktbildung beschreiben läßt und sich die Projektionen nur aus gewissen Elementarfunktionen durch Verkettung und kartesische Produktbildung zusammensetzen, kann man den Klon einer gewissen Funktionsmenge wie folgt charakterisieren. Dazu vereinbaren wir, daß eine Funktion f über einem Mengensystem M genau eine solche Funktion ist, deren Quelle ein endliches kartesisches Produkt von Mengen aus M und deren Ziel eine Menge aus M ist ($\mathrm{dom} f \in M^*$).

Satz 5: Es sei F eine Menge von Funktionen über einem endlichen Mengensystem $M = \{B_1,\dots,B_z\}$. Dann ist der Klon von F genau die

Menge von Funktionen über M, die in der von F erzeugten dht-symmetrischen Unterkategorie <u>K</u> von <u>Par</u> enthalten ist.

5. Inklusion

Für totale Funktionen $f_1: A \to B$, $f_2: A \to B$ ist für $f_1 = f_2$ notwendig und hinreichend, daß für jedes Element $a \in A$ die Gleichung $f_1(a) = f_2(a)$ zutrifft. Bei partiellen Funktionen ist dagegen noch die Bedingung $D(f_1) = D(f_2)$ zu beachten. So ist z. B. die Bedingung für einen Homomorphismus h von einer totalen Algebra (A,f) in eine totale Algebra (B,g), die durch $h^n g = fh$ für die n-stelligen Operationen f und g gegeben ist, nicht ohne weiteres auf partielle Algebren zu übertragen. Im Sinne von Hoehnke /3/ ist ein partieller Homomorphismus h zwischen partiellen Algebren (A,f) und (B,g) durch $D(fh) \subseteq D(h^n g)$ und $(h^n g)(a_1,\ldots,a_n) = (fh)(a_1,\ldots,a_n)$ für $(a_1,\ldots,a_n) \in D(fh)$ gekennzeichnet. Dieser Sachverhalt läßt sich in der vorgestellten Sprache durch das folgende kommutative Diagramm beschreiben:

$$\begin{array}{ccccc} A^n \otimes A^n & \xleftarrow{\;d_{A^n}\;} & A^n & \xrightarrow{\;f\;} & A \\ h^n g \otimes fh \downarrow & & (=) & & \downarrow h \, . \\ B \otimes B & & \xrightarrow{\;p_1^{B \otimes B}\;} & & B \end{array}$$

<u>Satz 6</u> (Schreckenberger /6/): Für partielle Funktionen $f: A \to B$ und $g: A \to B$ gilt $f \subseteq g$ (Inklusion für die entsprechenden Paarmengen) genau dann, wenn

$$d_A(g \otimes f) p_1^{B \otimes B} = f$$

erfüllt ist. Gleichwertig damit ist $d_A(g \otimes f) = d_A(f \otimes f) = f d_B$. ∎

Unabhängig von der mengentheoretischen Inklusion kann auf Grund der Axiome einer dht-symmetrischen Kategorie gezeigt werden, daß durch $d_A(g \otimes f) = f d_B$ für Morphismen $f,g: A \to B$ eine Halbordnung in der Morphismenklasse definiert ist, bezüglich der die Morphismenkomposition und die $\otimes$-Verknüpfung monoton sind /6/.

Literatur

/1/ Eilenberg, S., and Kelly, G. M.: Closed Categories. In: Proc. Conf. Categorical Algebra, La Jolla 1965, pp. 421 - 562, New York 1966

/2/ Budach, L., und Hoehnke, H. J.: Automaten und Funktoren. Berlin 1975

/3/ Hoehnke, H. J.: Superposition partieller Funktionen. In: Hoehnke, H. J. (Ed.): Studien zur Algebra und ihre Anwendungen. Schr. Zentralinst. Math. Mech. Akad. Wissensch. DDR 16, 7 - 26, Berlin 1972

/4/ Hoehnke, H. J.: On partial algebras. Preprint, Berlin 1976

/5/ Lugowski, H.: Grundzüge der Universalen Algebra. Leipzig 1976

/6/ Schreckenberger, J.: Über die Einbettung von dht-symmetrischen Kategorien in die Kategorie der partiellen Abbildungen zwischen Mengen. Zentralinst. Math. Mech. Akad. Wissensch. DDR, Preprint, Berlin 1980

/7/ Wille, R.: Allgemeine Algebra - zwischen Grundlagenforschung und Anwendbarkeit. Preprint, Darmstadt 1973

/8/ Vogel, H. J.: Operationen-Klone in speziellen dht-symmetrischen Kategorien. Wiss. Z. Pädagog. Hochsch. "Karl Liebknecht" Potsdam 24, Heft 1, 101 - 106 (1980)

eingegangen: 30. 03. 1981

Anschrift des Verfassers:

Dr. H.-J. Vogel
Mangerstraße 20
DDR-1500 Potsdam

Rostock. Math. Kolloq. 20, 33 - 52 (1982)
68B15
08A02

J. Demetrovics
Gy. Gyepesi

Some Logical Dependencies in Relational Data Base

0. Introduction

According to E.F.Codd [6] a relation is a matrix without two identical rows. Rows correspond to data records and columns to the attributes that are to be stored of a data item. He also introduced [7] the concept of functional dependency: a set of columns depends on another if fixing the values in a row taken on the first determine those on the second.

Other concepts of his are the key (a set of attributes on which all depend) and the candidate key (a minimal key).

Candidate keys clearly do not contain each other [12].

The possible mathematical structure of functional dependencies was first investigated by W.W.Armstrong [1]. Among others he found that this structure is determined by the maximal dependencies (those which have maximal attribute subsets depending on minimal ones) and even by the dependent sides of the maximal dependencies. We also heavily use these "maximal dependent subsets of attributes" as technical tools.

Different kinds of functional dependency have also been introduced [3], [9], [13], [15] and axiomatized, usually in similar systems to Armstrong's [8]. [10] discusses an interesting connection between the decomposition of relational data bases and the boolean switching functions.

The harder problems of the topic are usually of combinatorial nature (see [4], [5], [11], [16]).

In this paper in 1. we give the formal definition of the functional, dual, strong and weak dependencies and give new axioms for full f-d- and s-families.

In 2. develop the analogy and differences among the dependencies of different types and give an axiom for full w-families.

In 3. we deal with a question stated in [11].

Certain dependencies of a relational data base are known by its designer. We call these <u>initial dependencies</u>. In general initial dependencies imply new dependencies. W.W.Armstrong [2] has developed a method to find the dependencies implied by a given set of initial functional dependencies. He also gave a characterization of the sets of initial dependencies that imply all the dependencies of a given full f-family and are of minimal cardinality. This characterization has a logical nature; we give a combinatorial equivalent of it.

We use the following notational conventions:

Ω denotes the set of attributes, $P(\Omega)$ denotes his power set. If g is a function with X as its domain and $Z \subseteq X$ then $g\restriction_Z$ denotes the function which has domain Z and for any $z \in Z$ $g(z) = g\restriction_Z(z)$. $\subset$ means strict inclusion.

1. Old and New Axioms

We start with the definitions of functional, dual, strong and weak dependencies based on [1] and [8].

<u>Definition 1.1:</u> Let A,B be subsets of Ω and let R be a relation over Ω. Then we say that <u>B</u>

(i) <u>functionally</u>;
(ii) <u>dually</u>;
(iii) <u>strongly</u>;
(iv) <u>weakly</u>
<u>depends on A in R</u> if

(i) $(\forall g,h\in R)\ (g\restriction_A = h\restriction_A \rightarrow g\restriction_B = h\restriction_B)$;

(ii) $(\forall g,h\in R)\ ((\exists a\in A)(g(a)=h(a))\rightarrow(\exists b\in B)(g(b)=h(b)))$;

(iii) $(\forall g,h\in R)\ ((\exists a\in A)(g(a)=h(a))\rightarrow g\restriction_B = h\restriction_B)$;

(iv) $(\forall g,h\in R)\ (g\restriction_A = h\restriction_A \rightarrow (\exists b\in B)(g(b)=h(b)))$

holds respectively and denote these by

$A \xrightarrow[R]{f} B$, $A \xrightarrow[R]{d} B$, $A \xrightarrow[R]{s} B$, $A \xrightarrow[R]{w} B$ corresponding to the type of the denoted dependency.

The following example [8] illustrates the effect of the dual dependency.

<u>Example:</u> Let Ω = {author, title, hall, shelf}. Let we have a library with eighteen books, three halls and three shelves in every hall; one shelf holds two books. Let the relation R containing the datas of the library given by the following table:

author	title	hall	shelf
1	1	1	2
2	2	1	3
3	3	1	1
4	4	1	2
5	5	2	3
6	6	2	1
7	7	2	2
8	8	2	3
9	9	3	1
10	10	3	2
11	11	3	3
12	12	3	1
1	4	1	1
5	8	3	3
4	1	1	3
7	10	3	2
6	10	2	2
6	9	2	1

Thus {author, title} $\xrightarrow[R]{d}$ {hall, shelf} holds, and for i=1,...12 the book by author i and entitled i is on the $(1+3\cdot\{\frac{i}{3}\})$ -th shelf of the $[\frac{i+3}{4}]$-th hall ($[x]$ denotes the whole part and $\{x\}$ the fraction part of x). The reader, knowing the author or the title of the required book, may find it without examining the whole library: for example if $\underline{l}$ is the author of the book, then it is enough to look the $[\frac{l+3}{4}]$-th hall, and the $(l+3\cdot\{\frac{l}{3}\})$-th shelves of the other two halls.

In R {author, title} $\xrightarrow[R]{f}$ {hall, shelf} holds too, but to store this functional dependency is equivalent to store the table of R; the {author, title} $\xrightarrow[R]{d}$ {hall, shelf} dependency is more effective.

For proving the effectiveness of these dependencies we elaborated in the Automation Institute of the Hungarian Academy of Sciences a large-sized practical application of the relational data model.

We have planned an inventory-recording system for an agricultural corporation. The task of the system is to organize the component-traffic of about 350 agricultural estates. More exactly the task is: to record the inventory-stores, the orders of customers, to help the decisions making in this field and to help services.

First we used a traditional system concept for this purpose. Later this concept was transformed into the relational data model based on recent investigations. We saved about 40 percent of the memory capacity in this way. With using the results of Aho, Sagiv and Ullmann about relational expressions, we proved that the response time remained in the same order.

If R is a relation over Ω, and $Y\in\{F,D,S,W\}$ and $y\in\{f,d,s,w\}$ corresponds to Y then we use the notation

$$Y_R = \{(A,B): \ A \xrightarrow[R]{y} B\}.$$

We call <u>full y-families</u> the sets having this form.

In order to investigate the various dependencies the first step is the axiomatization of full y-families for $y\in\{f,d,s,w\}$. In [1] there is a system of axioms for full f-family and in [8] there are for full d- and s-families. For the sake of completeness we reproduce them here.

Let $Y \subseteq P(\Omega) \times P(\Omega)$. Then we say that Y satisfies the f-axioms, if for all $A,B,C,D \subseteq \Omega$

(F1) $(A,A)\in Y$;

(F2) $(A,B)\in Y,\ (B,C)\in Y \to (A,C)\in Y$;

(F3) $(A,B)\in Y,\ A \subseteq C,\ D \subseteq B \to (C,D)\in Y$;

(F4) $(A,B)\in Y,\ (C,D)\in Y \to (A\cup C,\ B\cup D)\in Y$.

Y satisfies the ϑ-axioms if for all $A,B,C,D \subseteq \Omega$

(D1) $(A,A)\in Y$;

(D2) $(A,B)\in Y,\ (B,C)\in Y \to (A,C)\in Y$;

(D3) $(A,B)\in Y,\ C \subseteq A,\ B \subseteq D \to (C,D)\in Y$;

(D4) $(A,B)\in Y,\ (C,D)\in Y \to (A\cup C,\ B\cup D)\in Y$;

(D5) $(A,\emptyset)\in Y \to A = \emptyset$.

Y satisfies the γ-axioms if for all $A,B,C,D \subseteq \Omega$

(S1) $(\{a\},\{a\})\in Y$;

(S2) $(A,B)\in Y,\ (B,C)\in Y,\ B \neq \emptyset \to (A,C)\in Y$;

(S3) $(A,B)\in Y,\ C \subseteq A,\ D \subseteq B \to (C,D)\in Y$;

(S4) $(A,B)\in Y,\ (C,D)\in Y \to (A\cap C,\ B\cup D)\in Y$;

(S5) $(A,B)\in Y,\ (C,D)\in Y \to (A\cup C,\ B\cap D)\in Y$.

We need the following technical lemma.

Lemma 1.1: Let $F \subseteq P(\Omega) \times P(\Omega)$ be such that $(X,Y)\in F$ and $Y \neq \emptyset$ imply $X \neq \emptyset$. Then F satisfies the f-axioms iff $D = \{(A,B):(B,A)\in F\}$ satisfies the ϑ-axioms.

Proof: Trivial by the f-and ϑ-axioms. (D5) makes necessary the assumption that $(X,Y)\in F$ and $Y \neq \emptyset$ imply $X \neq \emptyset$. □

Remark: The assumption ($(X,Y)\in F$ and $Y \neq \emptyset$ imply $X \neq \emptyset$) in lemma 1.1 is not an important restriction: if F satisfies the f-axioms let $F' = F\setminus\{(\emptyset,X):X \neq \emptyset\}$. Then F' obviously satisfies the f-axioms and the critical assumption as well and we have: $X \neq \emptyset$ implies that $(X,Y)\in F \leftrightarrow (X,Y)\in F'$.

In the following we give new axioms instead of the f-, ϑ- and γ-axioms and give an axiom that characterizes the weak full w-families which is such a full w-family that whenever (X,Y) is an element of the family then X is not void.

F-axiom: Let $F \subseteq P(\Omega) \times P(\Omega)$. Then we say that F satisfies the F-axiom if for any $(X,Y) \in P(\Omega) \times P(\Omega) \setminus F$ there is an $E \subseteq \Omega$ such that

(i) $X \subseteq E$ and $Y \not\subseteq E$;

(ii) if $(X',Y') \in F$ and $X' \subseteq E$ then $Y' \subseteq E$.

D-axiom: Let $D \subseteq P(\Omega) \times P(\Omega)$. Then we say that D satisfies the D-axiom if for any $(X,Y) \in P(\Omega) \times P(\Omega) \setminus D$ there is an $E \subseteq \Omega$ such that

(i) $X \cap E \neq \emptyset$ and $Y \cap E = \emptyset$;

(ii) if $(X',Y') \in D$ and $X' \cap E \neq \emptyset$ then $Y' \cap E \neq \emptyset$.

S-axiom: Let $S \subseteq P(\Omega) \times P(\Omega)$. Then we say that S satisfies the S-axiom if for any $(X,Y) \in P(\Omega) \times P(\Omega) \setminus S$ there is an $E \subseteq \Omega$ such that

(i) $X \cap E \neq \emptyset$ and $Y \not\subseteq E$;

(ii) if $(X',Y') \in S$ and $X' \cap E \neq \emptyset$ then $Y' \subseteq E$.

W-axiom: Let $W \subseteq P(\Omega) \times P(\Omega)$. Then we say that W satisfies the W-axiom if for any $(X,Y) \in P(\Omega) \times P(\Omega) \setminus W$ there is an $E \subseteq \Omega$ such that

(i) $X \subseteq E$ and $Y \cap E = \emptyset$,

(ii) if $(X',Y') \in W$ and $X' \subseteq E$ then $Y' \cap E \neq \emptyset$.

Theorem 1.1: (i) Let $F \subseteq P(\Omega) \times P(\Omega)$. Then F satisfies the f-axioms iff F satisfies the F-axiom.

(ii) Let $D \subseteq P(\Omega) \times P(\Omega)$. Then D satisfies the ϑ-axioms iff D satisfies the D-axiom.

(iii) Let $S \subseteq P(\Omega) \times P(\Omega)$. Then S satisfies the γ-axioms iff S satisfies the S-axiom.

Proof: (i) Suppose that F satisfies the F-axiom. Then:

(F1) If $(A,A) \notin F$ then there is an $E \subseteq \Omega$ such that $A \subseteq E$ and $A \not\subseteq E$ which is a contradiction.

(F2) If $(A,B)\in F$, $(B,C)\in F$ and $(A,C)\notin F$ then there is an $E\subseteq\Omega$ such that $A\subseteq E$ and $C\not\subseteq E$. Furthermore $(A,B)\in F$, $A\subseteq E$ imply $B\subseteq E$ and using $(B,C)\in F$, $C\subseteq E$ which is a contradiction.

The proof of (F3) and (F4) is analogous.

Suppose now that F satisfies the f-axioms. Let $(A,B)\in P(\Omega)\times P(\Omega)\setminus F$.

<u>Claim:</u> There is an $E\supseteq A$ such that $(E,B)\in P(\Omega)\times P(\Omega)\setminus F$ and $E'\supset E$ implies $(E',B)\in F$.

$(\Omega,\Omega)\in F$ by (F1). Thus, by (F3) $(\Omega,B)\in F$ holds. $A\subseteq\Omega$ and $(A,B)\in P(\Omega)\times P(\Omega)\setminus F$, consequently there is an $E\subseteq\Omega$ which is maximal w.r. to the properties $(E,B)\in F$ and $E\supseteq A$.

This E clearly satisfies the restrictions of the Claim. Let $E\supseteq A$ which is guaranteed by the Claim. We state that E satisfies (i) and (ii) of the F-axiom. Namely by the choice of E, $A\subseteq E$ holds. By (F1) and (F3) $B\subseteq E$ implies $(E,B)\in F$. Thus we have $B\not\subseteq E$.

Let $(C,D)\in F$ and $C\subseteq E$. $D\not\subseteq E$ implies $E'=D\cup E\supset E$ and by the maximality of E $(E',B)\in F$ holds.

$(E,E)\in F$ by (F1), hence (F4) implies that $(E,E')\in F$. Now $(E,E')\in F$ and $(E',B)\in F$ and (F2) imply that $(E,B)\in F$ which is a contradiction.

(ii) Let $F=\{(A,B):(B,A)\in D\}$. Then by Lemma 1.1 F satisfies the f-axioms iff D satisfies the ϑ-axioms. Hence, by (i), it is enough to show that F satisfies the F-axiom iff D satisfies the D-axiom.

Suppose that F satisfies the F-axiom. For $(A,B)\in P(\Omega)\times P(\Omega)\setminus F$ let $E(A,B)$ be such a subset of Ω that $A\subseteq E(A,B)$, $B\not\subseteq E(A,B)$ and if both $(A',B')\in F$ and $A'\subseteq E(A,B)$, then $B'\subseteq E(A,B)$. By the F-axiom such an $E(A,B)$ exists. By the definition of F whenever $(A,B)\in P(\Omega)\times P(\Omega)$ then $(A,B)\in P(\Omega)\times P(\Omega)\setminus F$ iff $(B,A)\in P(\Omega)\times P(\Omega)\setminus D$.

Now it is easy to chechk that for $(B,A)\in P(\Omega)\times P(\Omega)\setminus D$ $\Omega\setminus E(A,B)$ satisfies the D-axiom.

If D satisfies the D-axiom, then F satisfies the F-axiom; this can be shown by the same argument.

(iii) Suppose that S satisfies the S-axiom. Then the proof of the fact that S satisfies the γ-axioms is an easy modification of the proof of (i).

Suppose now that S satisfies the γ-axioms.
Let $(A,B)\in P(\Omega)\times P(\Omega)\setminus S$

Claim: There is an $a\in A$ and an $E\subseteq\Omega$
such that (a) $a\in E$;
(b) $(\{a\},E)\in S$ and
(c) $E'\supset E$ implies that $(\{a\},E')\notin S$.

If for any $a\in A$ we have $(\{a\},B)\in S$ then $(A,B)\in S$ by the repeated application of (S5). Hence there is an $a\in A$ such that $(\{a\},B)\notin S$. Now if for every $b\in B$ $(\{a\},\{b\})\in S$ holds then by the repeated application of (S4) we have $(\{a\},B)\in S$. Thus there is a $b\in B$ such that $(\{a\},\{b\})\notin S$.

By (S1) and (S3) there is an $E\subseteq\Omega$ such that $a\in E$, $(\{a\},E)\in S$ and E is maximal w.r. to this property. This E is appropriate for the Claim.

Let $E\subseteq\Omega$ and $a\in A$ guaranteed by the Claim. Then by (S3) we have $b\notin E$. Hence $A\cap E\neq\emptyset$ and $B\cap(\Omega\setminus E)\neq\emptyset$. Now let $(C,D)\in S$ such that $C\cap E\neq\emptyset$; let $c\in C\cap E$. Suppose that $D\cap(\Omega\setminus E)\neq\emptyset$; let $d\in D\cap(\Omega\setminus E)$. By (S3) we have $(\{c\},\{d\})\in S$ and by (S1) we have $(\{c\},\{c\})\in S$ $\cdot(\{a\},E)\in S$ implies that $(\{a,c\},\{c\})\in S$, by (S5). Hence (S3) implies that $(\{a\},\{c\})\in S$. Now $(\{a\},\{c\})\in S$, $(\{c\},\{d\})\in S$ and (S2) imply that $(\{a\},\{d\})\in S$. Thus by (S4) we have $(\{a\},E\cup\{d\})\in S$ which is a contradiction as $E' = E\cup\{d\}\supset E$.

Consequently the E guaranteed by the Claim demonstrates that S satisfies the S-axiom. □

It is worth to remark how can be find the full y-family - for $y\in\{f,d,s\}$ - generated by a given subset of $P(\Omega)\times P(\Omega)$ based on the Y-axiom. Let e.g. $y=f$ and let be given an $F'\subseteq P(\Omega)\times P(\Omega)$. Then the least full f-family containing F' is the following:

$$F = \{(A,B): A,B\subseteq\Omega \& (\forall E\subseteq\Omega)((A\subseteq E \& B\not\subseteq E) \rightarrow$$
$$\rightarrow (\exists(A',B')\in F')(A'\subseteq E \& B'\not\subseteq E))\}.$$

2. The Equality-Set

Definition 2.1: Let R be a relation over Ω. We define the equality set of R, $\&_R$ as follows:
For $h,g\in R$ let $E(h,g) = \{a\in\Omega: h(a) = g(a)\}$ and let $\&_R = \{E(h,g): h,g\in R \text{ and } h \neq g\}$.

Definition 2.2: Let **A** be a set system. Then A is a Δ-system if for any $A,B,C,D\in \mathbf{A}$ $A\neq B$ and $C\neq D$ implies that $A\cap B = C\cap D$.

Remark: It is easy to see that **A** is a Δ-system iff for any $A,B\in \mathbf{A}$ $A\neq B$ implies that $A\cap B = \cap\mathbf{A}$.

Theorem 2.1: (i) Let R be a relation over Ω and let h,f,g be different elements of R. Then $E(h,g)$, $E(h,f)$, $E(g,f)$ form a Δ-system.

(ii) Let $\& = \{E_{i,j}: 1\leq i<j\leq k\}$ such that for each $1\leq i<j<\ell\leq k$ $\{E_{i,j}, E_{i,\ell}, E_{j,\ell}\}$ is a Δ-system. Then there is a relation R over Ω with $\&_R = \&$.

Proof: (i) by symmetry it is enough to prove that $a\in E(h,g)\cap E(h,f)$ implies $a\in E(g,f)$. But this is trivial as $a\in E(h,g)\cap E(h,f)$ means both $h(a)=g(a)$ and $h(a)=f(a)$. Hence $g(a)= f(a)$ i.e. $a\in E(g,f)$.

(ii) We construct the rows of R by induction. Suppose that $n<k$, and the rows $h_1,\ldots h_n$ have been constructed so that for each $1\leq i<j\leq n$ $E(h_i,h_j) = E_{i,j}$ holds. We construct h_{n+1} as follows:

$$h_{n+1}(a) = \begin{cases} h_i(a), \text{ if } a\in E_{i,n+1} \text{ for some } 1\leq i\leq n; \\ \max\ (h_i(b): b\in\Omega \& 1\leq i\leq n)+1 \text{ else.} \end{cases}$$

Then

(a) h_{n+1} is well-defined.

To prove this we have to show that $a\in E_{i,n+1}\cap E_{j,n+1}$ implies $h_i(a) = h_j(a)$. But this is obvious because $E_{i,j}$, $E_{i,n+1}$, $E_{j,n+1}$ from a Δ-system and the induction hypothesis hold for $i,j\leq n$.

(b) if $1\leq i\leq n$ and $a\notin E_{i,n+1}$

then $h_i(a) \neq h_{n+1}(a)$.

Suppose first that $a\in E_{j,n+1}$ for some $1\leq j<n+1$. Then, by (a) and by the definition of h_{n+1}, $h_{n+1}(a) = h_j(a)$ holds. Furthermore $a\notin E_{i,j}$ because $\{E_{i,j}, E_{j,n+1}, E_{i,n+1}\}$ is a Δ-system. Thus the induction hypothesis implies $h_i(a) \neq h_j(a)$, that is $h_i(a) \neq h_{n+1}(a)$.

If $a\notin\bigcup_{1\leq j\leq n} E_{j,n+1}$ then we have $h_{n+1}(a)\neq h_i(a)$ by the definition of h_{n+1}. This completes the proof of (b).

Now by (a) and (b) it is clear that for $1\leq i\leq n$ $E(h_i, h_{n+1}) = E_{i,n+1}$ and hence the induction step works. Let $R = \{h_1,\ldots h_k\}$. Then $\mathcal{E}_R = \mathcal{E}$ obviously holds. □

After **Theorem** 2.1 there is a natural way to axiomatize full families of dependencies of any type. This follows next:

<u>F'-axiom:</u> Let $F\subseteq P(\Omega)\times P(\Omega)$. Then <u>F satisfies the F'-axiom</u> if there is a natural number k and an indexed set of subsets of Ω, $\{E_{i,j}: 1\leq i<j\leq k\}$ such that

(i) If $(X,Y)\in P(\Omega)\times P(\Omega)\setminus F$ then there are $1\leq i<j\leq k$ such that $X\subseteq E_{i,j}$ and $Y\not\subseteq E_{i,j}$.

(ii) If $(X,Y)\in F$, $1\leq i<j\leq k$ and $X\subseteq E_{i,j}$ then $Y\subseteq E_{i,j}$.

(iii) For any $1\leq i<j<\ell\leq k$ $\{E_{i,j}, E_{i,\ell}, E_{j,\ell}\}$ is a Δ-system.

D'-axiom: Let $D\subseteq P(\Omega)\times P(\Omega)$. Then D satisfies the D'-axiom if there is a natural number k and an indexed set of subsets of Ω, $\{E_{i,j}: 1\leq i<j\leq k\}$ such that

(i) If $(X,Y)\in P(\Omega)\times P(\Omega)\setminus D$ then there are $1\leq i<j\leq k$ such that $X\cap E_{i,j}\neq\emptyset$ and $Y\cap E_{i,j}\neq\emptyset$.

(ii) If $(X,Y)\in D$, $1\leq i<j\leq k$ and $X\cap E_{i,j}\neq\emptyset$ then $Y\cap E_{i,j}\neq\emptyset$.

(iii) The same as (iii) of the F'-axiom.

S'-axiom: Let $S\subseteq P(\Omega)\times P(\Omega)$. Then S satisfies the S'-axiom if there is a natural number k and an indexed set of subsets of Ω, $\{E_{i,j}: 1\leq i<j\leq k\}$ such that

(i) If $(X,Y)\in P(\Omega)\times P(\Omega)\setminus S$ then there are $1\leq i<j\leq k$ such that $X\cap E_{i,j}\neq\emptyset$ and $Y\not\subseteq E_{i,j}$.

(ii) If $(X,Y)\in S$, $1\leq i<j\leq k$ and $X\cap E_{i,j}\neq\emptyset$ then $Y\subseteq E_{i,j}$.

(iii) The same as (iii) of the F'-axiom.

W'-axiom: Let $W\subseteq P(\Omega)\times P(\Omega)$. Then W satisfies the W'-axiom if there is a natural number k and an indexed set of subsets of Ω, $\{E_{i,j}: 1\leq i<j\leq k\}$ such that

(i) If $(X,Y)\in P(\Omega)\times P(\Omega)\setminus W$ then there are $1\leq i<j\leq k$ such that $X\subseteq E_{i,j}$ and $Y\cap E_{i,j}=\emptyset$.

(ii) If $(X,Y)\in W$, $1\leq i<j\leq k$ and $X\subseteq E_{i,j}$ then $Y\cap E_{i,j}\neq\emptyset$.

(iii) The same as (iii) of the F'-axiom.

Remark: Observe that the $E_{i,j}$-s in the F'-axiom are maximal dependent sets i.e. if $(X,Y)\in F$ and $X\subseteq E_{i,j}$ then $Y\subseteq E_{i,j}$.

Theorem 2.2: (i) Let $Y\subseteq P(\Omega)\times P(\Omega)$ and $Y\in\{F,D,S\}$. Then Y satisfies the Y-axiom iff Y satisfies Y'-axiom.

(ii) Let Ω be a finite set, $|\Omega|\geq 3$.
Then there is a $W\subseteq P(\Omega)\times P(\Omega)$ such that W satisfies the W-axiom and W doesn't satisfy the W'-axiom.

<u>Proof:</u> (i) Let first $Y=F$ and suppose that Y satisfies the F-axiom. Write $Y=F$.

For any $(X,Y)\in P(\Omega)\times P(\Omega)\setminus F$ take an $E(X,Y)\subseteq\Omega$ guaranteed by the F-axiom. List these $E(X,Y)$-s as $E_2,\ldots E_k$ the indexes begin with (2!). For $1<j\leq k$ let $E_{1,j}=E_j$ and for $1<i<j\leq k$ let $E_{i,j}=E_i\cap E_j$.

We claim that $\{E_{i,j}:1\leq i<j\leq k\}$ demonstrates that F satisfies the F'-axiom.

The requirement (i) of the F'-axiom holds by $\{E_2,\ldots,E_k\}\subseteq\{E_{i,j}:1\leq i<j\leq k\}$.

We left to the reader to check that (ii) holds too.

To prove (iii) of the F'-axiom let $1\leq i<j<\ell\leq k$. We distinguish two cases:

(a) $i=1$. Then $E_{i,j}=E_j$; $E_{i,\ell}=E_\ell$ and $E_{j,\ell}=E_j\cap E_\ell$. Thus the intersection of any two members of $\{E_{i,j};E_{i,\ell};E_{j,\ell}\}$ is $E_j\cap E_\ell$. This means that $\{E_{i,j};E_{i,\ell};E_{j,\ell}\}$ is a Δ-system.

(b) $1<i$. Then $E_{i,j}=E_i\cap E_j;E_{i,\ell}=E_i\cap E_\ell$ and $E_{j,\ell}=E_j\cap E_\ell$. Thus the intersection of any two members of $\{E_{i,j};E_{i,\ell};E_{j,\ell}\}$ is $E_i\cap E_j\cap E_\ell$. This means that $\{E_{i,j};E_{i,\ell};E_{j,\ell}\}$ is a Δ-system.

If Y satisfies the F'-axiom then Y obviously satisfies the F-axiom.

Now let $Y=D$ and suppose that Y satisfies the D-axiom. Write $Y=D$.

For any $(X,Y)\in P(\Omega)\times P(\Omega)D$ take an $E(X,Y)\subseteq\Omega$ guaranteed by the D-axiom. List these $E(X,Y)$-s as $E_1,\ldots E_k$. For $1\leq i\leq k$ let $E_{2i-1,2i}=E_i$ and if $1\leq i<j\leq 2k$ and $E_{i,j}$ is still undefined then let $E_{i,j}=\emptyset$.

It is easy to see that $\{E_{i,j}: 1\leq i<j\leq 2k\}$ shows the D'-axiom to hold for D.

If D satisfies the D'-axiom then it trivially satisfies the D-axiom.

The case Y=S is an easy modification of the proof worked in the case Y=F.

(ii) For the sake of simplicity suppose that $\Omega=\{a,b,c\}$. (In the general case pick two different elements of Ω, a,b. The role of {c} will be played by $\Omega\setminus\{a,b\}$.)

Let $W=\{(A,B)\in P(\Omega)\times P(\Omega)$: $A\subseteq\{a\} \to a\in B$ and

$A\subseteq\{b\} \to b\in B$.

Then W satisfies the W-axiom while if $(A,B)\in P(\Omega)\times P(\Omega)\setminus W$ then either ($A\subseteq\{a\}$ and $a\notin B$) or ($A\subseteq\{b\}$ and $b\notin B$. For (A,B) $E=\{a\}$ taken in the first case and $E=\{b\}$ in the second shows the W-axiom to hold.

We claim that W doesn't satisfy the W'-axiom. Suppose indirectly that $\mathcal{E}=\{E_{i,j}: 1\leq i<j\leq k\}$ is a system that shows the W'-axiom to hold for W.

Then

(1) $\{a\}\in\mathcal{E}$ and $\{b\}\in\mathcal{E}$

while $(\{a\},\Omega\setminus\{a\})\in P(\Omega)\times P(\Omega)\setminus W$ and

$(\{b\},\Omega\setminus\{b\})\in P(\Omega)\times P(\Omega)\setminus W$ hold.

(2) $\emptyset\notin W$ and $\{c\}\notin W$

while $(\emptyset,\Omega)\in W$ and $(\{c\},\Omega\ \{c\})\in W$ hold.

By the "allocation" of {a} and {b}, we distinguish two cases:

(a) $\{a\}=E_{i,j}$ and $\{b\}=E_{i,\ell}$

Then $\{E_{i,j}; E_{i,\ell}; E_{j,\ell}\}$ is a Δ-system, that is $E_{j,\ell}=\emptyset$ or {c} which contradicts (2).

(b) $\{a\}=E_{i,j}$ and $\{b\}=E_{\ell,m}$, where $|\{i,j,\ell,m\}|=4$.

Now we are interested but in $E_{i,j}$, $E_{i,\ell}$, $E_{i,m}$, $E_{j,\ell}$, $E_{j,m}$ and $E_{\ell,m}$, thus we may suppose that $i=1$, $j=2$, $\ell=3$ and $m=4$.

Investigate what may be $E_{1,3}$.

The cases $E_{1,3} = \{a\}$ or $\{b\}$ arise to (a).
$E_{1,3} \neq \{c\}$ and $E_{1,3} \neq \emptyset$ by (2).

$E_{1,3} \neq \{b,c\}$ while $\{E_{1,2};E_{1,3};E_{2,3}\}$ is a Δ-system hence $E_{1,3} = \{b,c\}$ implies $E_{2,3} = \emptyset$ contradicting (2). Now it is clear that $a\in E_{1,3}$. Thus $a\in E_{2,3}$, while $\{E_{1,2};E_{1,3};E_{2,3}\}$ is a Δ-system.

$\{E_{2,3};\ E_{2,4};E_{3,4}\}$ is a Δ-system, hence $a \notin E_{2,4}$, that is $E_{2,4}\subseteq\{b,c\}$.

$E_{2,4}\neq \emptyset$ and $E_{2,4} \neq \{c\}$ by (2) and $E_{2,4} \neq \{b\}$ by (a). Hence $E_{2,4} = \{b,c\}$.

$\{E_{2,3};\ E_{2,4};E_{3,4}\}$ is a Δ-system, hence $b\in E_{2,3}$.
Finally $E_{1,3} = \{a,c\}$ while $E_{2,3},E_{1,2},E_{1,3}$ form a Δ-system.
Now $\{E_{1,3};E_{1,4};E_{3,4}\}$ is a Δ-system and $E_{1,3}\cap E_{3,4}= \emptyset$ and $E_{1,3}\cup E_{3,4}=\Omega$, hence $E_{1,4}=\emptyset$ which contradicts (2). □

<u>Remark:</u> Theorem 2.2 demonstrates the difference between the weak dependency and the rest.

<u>Theorem 2.3:</u> Let $Y\subseteq P(\Omega)\times P(\Omega)$ satisfy the Y'-axiom for some $Y\in\{F,D,S,W\}$. Then there is a relation R over Ω with $Y = Y_R$. Conversely if R is a relation over Ω then Y_R satisfies the Y'-axiom.

Proof: Let $\mathcal{E} = \{E_{i,j} : 1 \le i < j \le k\}$ show that Y satisfies the Y'-axiom. Then the requirement (iii) of the Y'-axiom and Theorem 2.1 (ii) imply that there is a relation R over Ω such that $\mathcal{E}_R = \mathcal{E}$. By the Y'-axiom it is obvious that $Y = Y_R$. Conversely, if R is a relation over Ω, then writing $R = \{h_1, \ldots h_k\}$, $E_{i,j} = E(h_i, h_j)$; $\{E_{i,j} : 1 \le i < j \le k\}$ shows that Y_R satisfies the Y'-axiom. □

3. Combinatorial Results

Definition 3.1. Let F be a full f-family and let $A \subseteq \Omega$. Then A is a candidate key for F if $(A, \Omega) \in F$ and for any $A' \subset A$ $(A', \Omega) \notin F$ holds. Let R be a relation over Ω, then the set of candidate keys of R is the set of candidate keys of F_R.

Let C denote the set of candidate keys of F. Then C is a Sperner system i.e. $(\forall A, B \in C)(A \subseteq B \rightarrow A = B)$.
We deal with the following question of [11]:

(*) What is the largest number $r(n)$ of rows that is needed for some $C \subseteq P(\Omega)$ being the set of candidate keys of a relation over Ω with $r(n)$ rows, where $|\Omega| = n$ and C is a Sperner system?

In [11] it is shown that for any Sperner system there is a relation with this system as its set of candidate keys and that

$$\sqrt{2\binom{n}{[n/2]}} \le r(n) \le 2\binom{n}{[n/2]}.$$

We give sharper estimations for $r(n)$.

Theorem 3.1: $$\frac{1}{n^2}\binom{n}{[n/2]} < r(n) \le \binom{n}{[n/2]} + 1.$$

Proof: First we prove the upper bound.
Let $C \subseteq P(\Omega)$ be a Sperner system. Let $\mathcal{B}$ consist of the maximal sets that do not contain members of C. Let the members of $\mathcal{B}$ be $B_2, \ldots B_k$. For $1 < j \le k$ let $E_{1,j} = B_j$ and

for $1<i\leq j\leq k$ let $E_{i,j} = B_i \cap B_j$. Then $\{E_{i,j}: 1\leq i<j\leq k\}$ satisfies the requirements of the Theorem 2.1 (ii), hence there is a relation R over Ω with k rows such that $\mathcal{E}_R = \{E_{i,j}: 1\leq i<j\leq k\}$. Then obviously **C** is the set of candidate keys of R. It is trivial that **B** is a Sperner system, and thus $|\mathbf{B}|\leq \binom{n}{[n/2]}$ that is

$k\leq \binom{n}{[n/2]}+1$.

Now let we seen the lower bound. We start with two trivial observations.

1. Let R be a relation over Ω with r rows. Then there is a relation R' over Ω such that R' uses no more than r symbols and $\mathcal{E}_R = \mathcal{E}_{R'}$.
2. Let R be a relation over Ω with r rows and let $r'>r$. Then there is a relation R' over Ω with r' rows such that $\mathcal{E}_R = \mathcal{E}_{R'}$. (Now we allow identical rows.)

By 1. and 2. the number of Sperner systems which may be represented as sets of candidate keys of a relation with r rows is no more then $r^{r\cdot n}$. Hence

$$r(n)^{r(n)\cdot n} > 2^{\binom{n}{[n/2]}} \quad \text{which implies}$$

$$r(n) > \frac{1}{n^2}\binom{n}{[n/2]}. \quad \square$$

If **B** is a Sperner system and R is a relation such that $\mathbf{B} \subseteq \mathcal{E}_R \subseteq \{\cap \mathbf{B}': \mathbf{B}' \subseteq \mathbf{B}\}$ then we can define two graphs on the set of rows of R as follows:

1. the **B** -graph of R is G_R where the vertices of G_R are the rows of R and two rows are connected by an edge if and only if their equality-set is an element of **B** .
2. the colored graph of R is the complet graph on the set of rows of R with the colour $E(f,g)$ on the edge $\{f,g\}$.

The **B** -graph of R has the following property: if G_R is disconnected, then there is a relation R' such that the number of rows of R' is less than that of R and $\mathbf{B} \subseteq \mathcal{E}_{R'} \subseteq \{\cap \mathbf{B}' : \mathbf{B}' \subseteq \mathbf{B}\}$. The colored graph of R contains no circuit the edges of which have the same colour except exactly one.

These two observations may be useful to make an algorithm to find the minimal relation for Sperner systems.

The estimation for $r(n)$ in Theorem is not sharp. If $\mathbf{B} = \{X \subseteq \Omega : |X| = [\frac{n}{2}]\}$, then there is a relation R such that $\mathbf{B} \subseteq \mathcal{E}_R \subseteq \{\cap \mathbf{B}' : \mathbf{B}' \subseteq \mathbf{B}\}$ and the number of rows of R is the least natural number greater than

$$\frac{1}{2}\binom{n}{[n/2]} + \sqrt{2\binom{n}{[n/2]}}.$$

It is natural to ask the following analogon of (*): What is the largest number $R(n)$ of rows that is needed to represent a relation with $\mathbf{F}$ as the set of functional dependencies of it for an $\mathbf{F} \subseteq P(\Omega) \times P(\Omega)$ where $|\Omega| = n$ and $\mathbf{F}$ is a full f-family.

By the proof of Theorem 2.2 (i) it is obvious that $R(n) \leq$ the maximal number of subsets of Ω such that the intersection of any two of them is not a third. Thus, by a theorem of D.Kleitman [13], $R(n) \leq c.\binom{n}{[n/2]}$ where $c = 3/2$. Z.Füredi and J.Pach have shown, that this number is less then $(1 + (c.\log n)/n)\binom{n}{[n/2]}$. It is trivial that $r(n) \leq R(n)$.

Lastly we give the combinatorial characterization - according to §0 - of the sets which are of minimal cardinality with respect to the property that they imply all the dependencies of a given full f-family.

We need some definitions and a lemma.

Definition 3.2: Let $\mathbf{M} \subseteq P(\Omega)$.

(i) We say that M has the intersection property if for any $\mathbf{M}' \subseteq \mathbf{M}$ $\cap \mathbf{M}' \in \mathbf{M}$ holds.

(ii) An $M \in \mathbf{M}$ is irreducible if $M \neq \cap\{M' \in \mathbf{M} : M \subset M'\}$ (recall that $\subset$ means strict inclusion).

(iii) An $\mathbf{N} \subseteq \mathbf{M}$ generates $\mathbf{M}$ if $\mathbf{M} = \{\cap \mathbf{N}' : \mathbf{N}' \subseteq \mathbf{N}\}$.

Lemma 3.1: Let $\mathbf{M}$ have the intersection property and let $\mathbf{N} = \{M \in \mathbf{M}: M \text{ is irreducible}\}$. Then an $\mathbf{N'} \subseteq \mathbf{M}$ generates $\mathbf{M}$ iff $\mathbf{N} \subseteq \mathbf{N'}$.

Proof: The following proof is standard in lattice theory. If $\mathbf{N'}$ generates $\mathbf{M}$, then $\mathbf{N} \subseteq \mathbf{N'}$ is obvious. For the converse we have to prove that $\mathbf{N}$ generates $\mathbf{M}$. Suppose indirectly that there is an $X \in \mathbf{M} \setminus \mathbf{N}$, such that $X \neq \cap\{Y: Y \in \mathbf{N} \& X \subseteq Y\}$. Let X be of minimal cardinality with respect to this property. $X \notin \mathbf{N}$ means that $X = \cap\{Y: Y \in \mathbf{M} \& X \subset Y\}$, hence $X \subset Y$ implies that there is an $\mathbf{N}_Y \subseteq \mathbf{N}$ such that $Y = \cap \mathbf{N}_Y$. Let $\mathbf{N}_X = \cup\{\mathbf{N}_Y: X \subset Y \text{ and } Y \in \mathbf{M}\}$.
Then $\mathbf{N}_X \subseteq \mathbf{N}$ and $X = \cap \mathbf{N}_X$ which is a contradiction.□

Remark: Observe that the proofs of Theorems in [2] are essentially our proof of Lemma 3.1.

Corollary: If $\mathbf{M}$ has the intersection property then there is exactly one $\mathbf{N} \subseteq \mathbf{M}$ which generates $\mathbf{M}$ and has minimal cardinality.

Theorem 3.2: Let $\mathbf{F}$ be a full f-family, let $\mathbf{B}$ be the set of maximal dependent set for $\mathbf{F}$ and let $\mathbf{C}$ be the set which generates $\mathbf{B}$ and has minimal cardinality (in [1] there is shown that $\mathbf{B}$ has the intersection property).
Then for any $\mathbf{F'} \subseteq \mathbf{F}$ we have the following:
$\mathbf{F'}$ implies all the dependencies of $\mathbf{F}$ and $\mathbf{F'}$ has minimal cardinality with respect to this property
if and only if
for any $C \in \mathbf{C}$ there is an $A_C \subseteq \Omega$ such that $\mathbf{F'} = \{(A_C, C) : C \in \mathbf{C}\}$.

We left the easy proof of the Theorem to the reader. We think that it is interesting to compare Theorem 3.2 with the Theorem on pp. 16 of [2].

References

[1] Armstrong, W. W.: Dependency structures of data base relationships. In: Information Processing 74. Proc. IFIP Congress, Stockholm 1974. pp. 580 - 583, Amsterdam 1974

[2] Armstrong, W. W.: On the generation of dependency structures of relational data bases. Publication ≠ 272, Université de Montréal 1977

[3] Beeri, C., Fagin, R., and Howard, J. H.: A complete axiomatization for functional and multivalued dependencies in database relations. In: Proc ACM SIGMOND Int. Conf. on Management of Data, Toronto 1977, 47 - 61

[4] Békéssy, A., and Demetrovics, J.: Contribution to the theory of data base relations. Discrete Math. 27, 1 - 10 (1979)

[5] Békéssy, A., Demetrovics, J., Hannák, L., Frankl, P., and Katona, G.: On the number of maximal dependencies in a data base relation of fixed order. Discrete Math. 30, 83 - 88 (1980)

[6] Codd, E. F.: A relational model of data for large shared data banks. Comm. ACM. 13, 377 - 387 (1970)

[7] Codd, E. F.: Further normalization of the data base relational model. In: Ristin, R. (Ed.): Data Base System. Courant Computer Science Symposium 6, New York 1971. pp. 33 - 64, Englewood Cliffs, N. J., 1972

[8] Czédli, G.: Függőségek relációs adatbázis modellben. Alkalmaz. Mat. Lapok (1980)

[9] Delobel, C.: Normalization and hierarchical Dependencies in the Relational Data Model. ACM Trans. Database Sys. 3, 201 - 222 (1978)

[10] Delobel, C., Casey, R. G., and Bernstein, Ph. A.: Decomposition of a Data Base and the Theory of Boolean Switching Functions. IBM J. Res. Develop. 17, 374 - 386 (1973)

[11] Demetrovics, J.: Candidate keys and antichains, SIAM J. Algebraic Discrete Methods 1, 92 (1980)

[12] Demetrovics, J.: On the equivalence of candidate keys with Sperner systems. Acta Cybernet. 4, 247 - 252 (1979)

[13] Fagin, R.: Multivalued dependencies and a new normal form for relational data-bases. ACM Trans. Database Sys. 2, 262 - 278 (1977)

[14] Kleitman, D.: On a problem of Erdős. Proc. Amer. Math. Soc. 18, 139 - 141 (1966)

[15] Mendelzon, A. O.: On axiomatizing multivalued dependencies in relational databases. J. Assoc. Comput. Mach. 26, 37 - 44 (1979)

[16] Rissanen, J.: Independent components of relations. ACM Trans. Database Sys. 2, 317 - 325 (1977)

[17] Sperner, E.: Ein Satz über Untermengen einer endlichen Menge. Math. Z. 27, 544 - 548 (1928)

[18] Yu, C. T., and Johnson, D. T.: On the complexity of finding the set of candidate keys for a given set of functional dependencies. Inform. Process. Lett. 5, 100 - 101 (1976)

received: March 30, 1981

Author's address:

J. Demetrovics, Gy. Gyepesi
Computer and Automation Institute
of Hungarian Academy of Sciences
H 1502 Budapest, XI. Kende u. 13-17

Rostock. Math. Kolloq. 20, 53 - 60 (1982) 68899

Volker Wedler

Invariant Properties and Normalizations of Pictures

The classificetion of objects given by geometric transformed pictures es caused by shifts, rotations, distortions, or others, represent a serious problem in picture processing. The present paper discueees an unified epproech for the cheracterization of invariant properties end normalizations of pictures formed on Rosenfeld-Kak /1/, where basic definitions to the conception of invariant properties are given, allowing an algebraic treatment of this field. It will be shown that there exists a netural relation between invariant properties and normalizations of pictures.

1. Pictures, Transformations, Properties

We shall considar digital pictures which are generally defined in the following way.

Definition 1: Let m,n be natural numbers end $G \subseteq R^1$, then any mapping $f:\{0,1,\dots,m-1\} \times \{0,1,\dots,n-1\} \longrightarrow G$ is celled a (digital) $m \times n$ p i c t u r e with gray-value range G. The set of all pictures is denoted by B.

Note thet the restriction to pictures with quantisized domains is only a concession to digital picture processing, the generalization of our considerations to pictures defined in the real plane is straight forward.
In practice G ie a finite set of gray levels. If $G = \{0,1\}$ the pictures will be noted as binary pictures.
A picture may be represented by a $m \times n$ matrix with valuee from G.

Definition 2: A mapping T from B into B is called a p i c t u r e t r a n s f o r m a t i o n, $\mathcal{T} := \{T:B \to B\}$.

Definition 3: For a certain fixed set $K \subseteq R^1$ a mapping P from B into K is called a p i c t u r e p r o p e r t y,
$\mathcal{P} := \{P:B \rightarrow K\}$.

If K is a finite set, then a property P may be viewed as a picture classification strategy that maps B into card(K) classes. The most frequently appearing picture transformations are geometrical transformations like translations and rotations. We shall consider translations as cyclic operations within the $m \times n$ domain which is no essential restriction.
Important properties are quantitative features of picture objects like length, breadth, circumference etc., or properties which can be computed by the moments of pictures /2/, /3/ as well.

2. Invariances and Normalizations

The definitions of invariant properties and normalizations by Rosenfeld and Kak /1/ correspond to the intuitive conception of these notions.

Definition 4: Let S be a set of picture transformations. A property P is called i n v a r i a n t in relation to S iff $P(T(f)) = P(f)$ holds for all pictures f and for all transformations $T \in S$. The set of all properties which are invariant in relation to S is called i n v a r i a n c e c l a s s of S, and denoted by Inv(S). A transformation N is called n o r m a l i z a t i o n in relation to S iff $N(T(f)) = N(f)$ holds for all pictures f and for all transformations $T \in S$. The set of all normalizations in relation to S is called n o r m a l i z a t i o n c l a s s of S, and is denoted by Norm(S).

For these classes some important properties can be derived which will be mentioned in the following theorem. S^* denotes the algebraic closure of S in relation to superposition.

Theorem 1:
1. Inv(∅) is the set of all properties,
2. Norm(∅) is the set of all transformations,

3. $Inv(\mathcal{T})$ is the set of all constant properties,
4. $Norm(\mathcal{T})$ is the set of all constant transformations,
5. If $S_1 \subseteq S_2$ then $Inv(S_2) \subseteq Inv(S_1)$,
6. If $S_1 \subseteq S_2$ then $Norm(S_2) \subseteq Norm(S_1)$,
7. $Inv(S_1) \cap Inv(S_2) = Inv(S_1 \cup S_2)$,
8. $Norm(S_1) \cap Norm(S_2) = Norm(S_1 \cup S_2)$,
9. $Inv(S) = Inv(S^*)$,
10. $Norm(S) = Norm(S^*)$,
11. $Inv(S) = \{P'$: there exists a property P so that $P'(f) = P(N(f))$ holds for all pictures $f\}$.

__Proof:__ Only the properties for the invariance classes will be proved, the proofs for normalization classes can be realized analogously. The properties 1,...,4 are obvious.

__5.__ If $P \in Inv(S_2)$ then $P(T(f)) = P(f)$ holds for all pictures f and all transformations $T \in S_2$. Because of $S_1 \subseteq S_2$, this holds for all $T \in S_1$, too. Therefore $P \in Inv(S_1)$.

__7.__ Because of $S_1 \subseteq S_1 \cup S_2$ and property 5, we have $Inv(S_1) \supseteq Inv(S_1 \cup S_2)$. Analogously $Inv(S_2) \supseteq Inv(S_1 \cup S_2)$. Therefore $Inv(S_1) \cap Inv(S_2) \supseteq Inv(S_1 \cup S_2)$ follows.
If $P \in Inv(S_1) \cap Inv(S_2)$ then $P(T_1(f)) = P(f)$ and $P(T_2(f)) = P(f)$ holds for all pictures f, for all $T_1 \in S_1$ and for all $T_2 \in S_2$. Therefore $P(T(f)) = P(f)$ holds for all pictures f and for all transformations $T \in S_1 \cup S_2$. It follows that $Inv(S_1) \cap Inv(S_2)$ is a subset of $Inv(S_1 \cup S_2)$. Altogether these sets are equal.

__9.__ Let $P \in Inv(S)$, $T_1, T_2 \in S$, and f an arbitrary picture, and $f' = T_2(f)$. From the invariance of P it follows that
$P(T_1(T_2(f))) = P(T_1(f')) = P(f') = P(T_2(f)) = P(f)$.
By the way of induction it can be shown that this equality holds for any superposition of transformations of S. On the other side it is clear, that there exists no transformation outside S so that all properties of Inv(S) are invariant in relation to this transformation.

__11.__ If N is a normalization in relation to S, and P an arbitrary property then $P(N(T(f))) = P(N(f))$ holds for all pictures f and all transformations $T \in S$. Thus each property of normalized pictures is an invariant property. On the other side, always there exists a normalization N that is itself a transformation of S so that $P(N(f)) = P(f)$ holds for any property $P \in Inv(S)$ and all pictures f. Therefore any invariant property can be represented in this way. □

In picture processing such normalizations of pictures are of special interest, because the properties of normalized pictures are all invariant properties. If the invariant properties in relation to translations shall be investigated then it suffices to search the properties of such pictures which arise when the centre of gravity is translated into a fixed point. This normalization fulfils the condition that all invariant properties can be derived from the normalized pictures. For a few other classes of transformations, normalizations are known, too /1/.

3. The Algebra of Invariance Classes

The formation of invariance classes represents a mapping from the power set of transformations into the power set of properties. It is possible to construct a mapping that produces from a set D of properties the maximal set of transformations so that D is a subset of the invariance class of this set of transformations. First the following system of sets will be constructed,

$$\Sigma_D = \{S: S \text{ is a set of transformations and } Inv(S) \supseteq D\}.$$

The union of this system of sets is called t r a n s f o r m a t i o n c l a s s, and is denoted by Tr(D).
The production of this transformation class represents a mapping from the power set of properties into the power set of transformations, and has similar properties as the mapping Inv.

Theorem 2:

1. $Tr(\emptyset)$ is the set of all transformations,
2. $Tr(\mathcal{P})$ is the singleton of the identic transformation,
3. If $D_1 \subseteq D_2$ then $Tr(D_2) \subseteq Tr(D_1)$,
4. $Tr(D_1) \cap Tr(D_2) = Tr(D_1 \cup D_2)$,
5. $Tr(D) = (Tr(D))^*$.

This theorem can be proved analogously to Theorem 1.

Therefore the pair of mappings (Inv, Tr) is a Galois connection, and in a natural way the following two hull operators can be defined:

$\Gamma_T(S) = Tr(Inv(S))$, $\Gamma_P(D) = Inv(Tr(D))$.

With the help of these hull operators two equivalence relations $\equiv_P$ on the power set of properties and $\equiv_T$ on the power set of transformations can be defined. Two sets D_1, D_2 are equivalent, $D_1 \equiv_P D_2$, if and only if $\Gamma_P(D_1) = \Gamma_P(D_2)$, for transformation classes, $S_1 \equiv_T S_2$, analogously. If from every equivalence class the greatest element is chosen then two algebras of sets [P] and [T] arise.

It is easy to see that all elements of these algebras are closed in relation to their hull operator, for all $D \in [P]$ and all $S \in [T]$ the equalities $\Gamma_P(D) = D$ and $\Gamma_T(S) = S$ hold.

For the algebra [P] the following natural operations can be defined:

$max_P(D_1,D_2) = \Gamma_P(D_1 \cup D_2)$,

$min_P(D_1,D_2) = D_1 \cap D_2$.

It is obvious that the maximum is again an element of [P], and the minimum by property 7 of Theorem 1, too.
Analogously, for [T] it can be defined,

$max_T(S_1,S_2) = \Gamma_T(S_1 \cup S_2)$,

$min_T(S_1,S_2) = S_1 \cap S_2$,

where maximum and minimum are again elements of [T].
With these operations the algebras [P] and [T] are lattices which are related to each other as follows:

<u>Theorem 3:</u> $[P, max_P, min_P]$ and $[T, max_T, min_T]$ are dual isomorphic lattices; both with maximum and minimum (1 and 0).

<u>Proof:</u> The mapping Inv is defined for any set of transformations, hence for all elements of [T], too.
If $D \in [P]$ then $Inv(Tr(D)) = D$ holds, therefore $Tr(D) \in [T]$ is an origin of D in relation to Inv. Thus, Inv is a mapping from [T] onto [P].

If $S_1, S_2 \in [T]$ and $Inv(S_1) = Inv(S_2)$ then $Tr(Inv(S_1)) = Tr(Inv(S_2))$, and therefore, $\Gamma_T(S_1) = \Gamma_T(S_2)$ holds.
Because of the closedness of all elements of $[T]$, $S_1 = S_2$ holds.
That's why Inv is a one-to-one mapping, the inverse mapping is Tr.
With the help of the points 7 and 8 of Theorem 1

$$Inv(max_T(S_1,S_2)) = Inv(\Gamma_T(S_1 \cup S_2)) = Inv(Tr(Inv(S_1 \cup S_2)))$$
$$= Inv(S_1 \cup S_2) = Inv(S_1) \cap Inv(S_2) = min_P(Inv(S_1), Inv(S_2)) \text{ and}$$

$$max_P(Inv(S_1), Inv(S_2)) = \Gamma_P(Inv(S_1) \cup Inv(S_2))$$
$$= Inv(Tr(Inv(S_1) \cup Inv(S_2))) = Inv(Tr(Inv(S_1)) \cap Tr(Inv(S_2)))$$
$$= Inv(\Gamma_T(S_1) \cap \Gamma_T(S_2)) = Inv(S_1 \cap S_2) = Inv(min_T(S_1,S_2))$$

follows.
Thus the lattices $[P]$ and $[T]$ are dually isomorphic. The minimum of $[P]$ is $\Gamma_P(\emptyset)$, the set of all constant properties. The maximum of $[T]$ is the set of all transformations, and $Inv(1_T) = 0_P$ holds. On the other side, the minimum of $[T]$ is $\Gamma_T(\emptyset)$, the singleton of the identic transformation, and $Inv(0_T) = 1_P$, the set of all properties. □

For the normalizations it is possible to construct analogously a lattice $[N]$, this lattice is isomorphic to the lattice $[P]$.
If a set S of transformations is composed by $S_1, \ldots, S_k$ in the following way, $S = (S_1 \cup \ldots \cup S_k)^*$ then for the invariance classes $Inv(S) = Inv(S_1) \cap \ldots \cap Inv(S_k)$ holds.

4. An Example

Let $S_1 = \{T_1\}$ and $S_2 = \{T_2\}$ with

$$T_1(f)(x,y) = 1/3\ (f(x-1,y) + f(x,y) + f(x+1,y)) \text{ and}$$
$$T_2(f)(x,y) = 1/3\ (f(x,y-1) + f(x,y) + f(x,y+1)).$$

These transformations are the productions of the means in a small neighbourhood of the point, a blurring of the picture; T_1 in the rows and T_2 in the columns. The superpositions of T_1, or T_2 is a blurring within larger domains. The only invariant properties in relation to these transformations are the means of the rows, and functions of them, or the means of the columns, and functions of them. Thus:

$$Inv(S_1) = \left\{P:\ P(f) = g(\mu_1,\dots,\mu_n),\ g:R^n \to R^1,\ \mu_j = 1/m\sum_{k=1}^{m} f(j,k)\right\}.$$

$$Inv(S_2) = \left\{P:\ P(f) = g(\nu_1,\dots,\nu_m),\ g:R^m \to R^1,\ \nu_k = 1/n\sum_{j=1}^{n} f(j,k)\right\}.$$

Any value that can be computed by the means of the rows, and by the means of the columns, too, can be computed by the mean of the whole picture. Therefore

$$Inv(S_1) \cap Inv(S_2)$$

$$= \left\{P:\ P(f) = g(\mu),\ g:R^1 \to R^1,\ \mu = 1/mn \sum_{j=1}^{n} \sum_{k=1}^{m} f(j,k)\right\}.$$

On the other side each transformation of $(S_1 \cup S_2)^*$ is a blurring of the whole picture, and the only invariant properties are the mean of the whole picture and functions of it.
In this case the relation $Inv(S_1) \cap Inv(S_2) = Inv(S_1 \cup S_2)$ is clear immediately.

5. Conclusions

The continuation of this theory can be the finding of the atoms of the lattice [T] and of the dual atoms of [P] and [N], or the searching of algorithms for normalizations for important classes of transformations.

References

/1/ Rosenfeld, A., and Kak, A. C.: Digital Image Processing. New York 1980

/2/ Hu, M.-K.: Visual Pattern Recognition by Moment Invariants. IRE Trans. Inform. Theory IT-8, 179 - 187 (1962)

/3/ Gonzalez, R., and Wintz, P.: Digital Image Processing. Reading, Mass., 1977

received: March 30, 1981

Author's address:

Dipl.-Math. V. Wedler
Friedrich-Schiller-Universität
Sektion Mathematik
Universitätshochhaus 17. OG
DDR-6900 Jena

Rostock. Math. Kolloq. 20, 61 - 71 (1982) 68D20
93A05

Gottfried Wolf

Zur Charakterisierung stabiler Konfigurationen auf binären zellularen Automaten

0. Zusammenfassung

Die Arbeit befaßt sich mit der Charakterisierung von stabilen Konfigurationen auf binären zellularen Automaten. Die (lokale) Überführungsfunktion ist die Implikation. Das Kopplungsmuster ist die von-Neumann-Kopplung. Dabei wird, aufbauend auf vorliegenden Resultaten für die linearen bzw. verallgemeinert linearen Funktionen Antivelenz, Äquivalenz, Konjunktion und Disjunktion, deren Gültigkeit auch auf die nichtlineare Funktion Implikation erweitert.
Für die untersuchte Funktion gibt es auf zellularen Automaten stabile Konfigurationen. Die implikative "Überlagerung" dieser stabilen Konfigurationen ergibt wiederum stabile Konfigurationen.
Der Begriff "Rand einer Konfiguration" als eine Teilmenge des Zellgitters wird eingeführt. Jede stabile Konfiguration wird durch ihre Werte auf dem Rand vollständig charakterisiert. Ein Rand wird für die untersuchte Funktion angegeben.
Die Gültigkeit des Superpositionsprinzips für stabile Konfigurationen ermöglicht - zusammen mit dem Rand - den Aufbau stabiler Konfigurationen aus einfachen Grundelementen. Für die untersuchte Funktion wird eine Basis bestimmt.

1. Grundbegriffe

Die Definitionen lehnen sich an die gebräuchliche Terminologie an (/3/). Ein zellularer Automat besteht aus einer unendlichen Menge identischer endlicher Automaten. Diese Automaten sind in einem zweidimensionalen Gitter angeordnet. Jeder Automat ist in der gleichen Weise mit seinen Nachbarn verbunden. Formal beschreiben wir einen zellularen Automaten durch ein Tripel

$\mathcal{L} = (A, G^2, S)$:

- G^2 ist das Zellgitter. $G^2 = \{r_1, r_2\}$ ist die Menge aller geordneten Paare ganzer Zahlen. Das Paar $\mathfrak{r} = (r_1, r_2) \in G^2$ beschreibt mit den Cartesischen Koordinaten r_1, r_2 einen Punkt $\mathfrak{r}$ in der Ebene des Zellgitters. Jedem dieser Punkte in der Ebene wird ein Zellautomat $A_{\mathfrak{r}}$ zugeordnet. Dieser Zellautomat wird auch kurz Zelle $\mathfrak{r}$ genannt.

- S ist das Kopplungsmuster oder die Nachbarschaftskopplung. Das Kopplungsmuster wird hier durch ein 5-Tupel von Vektoren

$$S = (s_0, s_1, s_2, s_3, s_4)$$

mit $s_0 = (0,0)$, $s_1 = (1,0)$, $s_2 = (0,1)$, $s_3 = (-1,0)$, $s_4 = (0,-1)$ beschrieben. Dieses Kopplungsmuster ist die sogenannte von-Neumann-Kopplung. Die Nachbarn $N(\mathfrak{r})$ einer Zelle sind

$$N(\mathfrak{r}) = \{\mathfrak{r}' \mid \mathfrak{r}' = \mathfrak{r} + s_i \wedge s_i \in S,\ i = 0,1,2,3,4\}.$$

Die Zelle $\mathfrak{r}_0$ bezeichnen wir als Zentrumszelle. Die Outputs der Automaten $A_{\mathfrak{r}'}$, $\mathfrak{r}' \in N(\mathfrak{r})$ sind die Inputs von $A_{\mathfrak{r}}$. Die Nachbarn einer Zentrumszelle $\mathfrak{r}$ zeigt Bild 1.

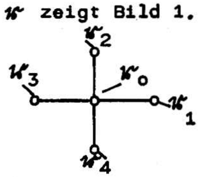


Bild 1

- A ist der Zellautomat mit $A = (X, Y, Z, \delta, \lambda)$, wobei in dieser Arbeit $X = Z^5$, $Y = Z = \{0,1\}$, $\delta: X \times Z \rightarrow Z$, $\lambda: Z \rightarrow Y$, $\lambda(z) = z$ für $z \in Z$ ist. X, Y, Z sind die Input-, Output- und Zustandsmenge, δ und λ die Überführungs- und die Ausgabefunktion - entsprechend der gebräuchlichen Schreibweise. Die verwendete Überführungsfunktion ist die logische Funktion Implikation. Für diese Überführungsfunktion gilt - unter Beachtung der Verkopplung mit den 5 Nachbarzellen - folgendes:

$$\delta\left[z_{\varkappa+s_0}(t),\ z_{\varkappa+s_1}(t),\ z_{\varkappa+s_2}(t),\ z_{\varkappa+s_3}(t),\ z_{\varkappa+s_4}(t)\right]$$

$$=((((z_{\varkappa+s_0}(t)\rightarrow z_{\varkappa+s_1}(t))\rightarrow z_{\varkappa+s_2}(t))\rightarrow z_{\varkappa+s_3}(t))\rightarrow z_{\varkappa+s_4}(t))$$

$$= z_{\varkappa_0}(t+1).$$

Durch die spezielle Klammerung und Reihenfolge der Argumente von δ wird jeweils eine spezielle Funktion δ festgelegt. Die Untersuchung aller möglichen Klammerungen und Reihenfolgen erfolgt im Rahmen dieses Vortrages nicht. Die betrachtete Verknüpfung der Eingangsgrößen soll an einem Schaltschema (Bild 2) verdeutlicht werden.

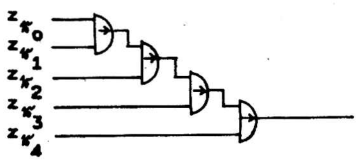


Bild 2

Eine Konfiguration K, $K \subseteq G^2$, ist die Menge aller erregten Zellen: $K = \{\varkappa \mid \varkappa \in G^2 \wedge z_{\varkappa} = 1\}$. K ist endlich (unendlich), wenn die Zahl der erregten Zellen endlich (unendlich) ist.

Die der Konfiguration K zugeordnete Belegung $\mathfrak{B}^K$ ist folgendermaßen definiert:

$$\mathfrak{B}^K : G^2 \rightarrow \{0,1\},$$

wobei $\mathfrak{B}^K(\varkappa) = 1$, wenn $\varkappa \in K$, und $\mathfrak{B}^K(\varkappa) = 0$, wenn $\varkappa \notin K$ ist.

Die Einschränkung der Belegung $\mathfrak{B}^K$ auf $R \subseteq G^2$ des Zellgitters wird mit $\mathfrak{B}^K_R$ bezeichnet.

Die lokale Belegung $\mathfrak{B}(N(\varkappa))$ ist die Belegung einer Zelle und aller ihrer Nachbarn, also

$$\mathfrak{B}(N(\varkappa)) = \{z_{\varkappa'} \mid \varkappa' \in N(\varkappa)\}.$$

Wir wollen diesen Begriff an einigen Beispielen erläutern. Für die von-Neumann-Kopplung sind im Bild 3 einige Beispiele von lokalen Belegungen gegeben.

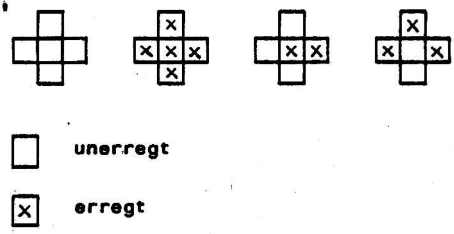


Bild 3

2. Stabile Konfigurationen

Eine Konfiguration K zum Zeitpunkt t wird als K^t bezeichnet. Eine Konfiguration K ist stabil, wenn sie sich zeitlich nicht ändert, d. h., wenn $K = K^t$ für alle $t > 0$ gilt. Offensichtlich folgt aus $K^t = K^{t+1}$ auch $K^{t+T} = K$ für $T > 0$. Damit gilt für jedes $\varkappa \in G^2$ einer stabilen Konfiguration

$$z_{\varkappa}(t+1) = z_{\varkappa}(t). \tag{1}$$

Dieser Gleichung (1) entspricht bei der angenommenen implikativen Verknüpfung

$$z_{\varkappa_0} = ((((z_{\varkappa_0} \rightarrow z_{\varkappa_1}) \rightarrow z_{\varkappa_2}) \rightarrow z_{\varkappa_3}) \rightarrow z_{\varkappa_4}) \tag{2}$$

$$(\varkappa_i \in G^2,\ i=0,\ldots,4).$$

Die Gleichung (2) ist die sogenannte lokale Stabilitätsbedingung, die für alle Zellen $\varkappa_0 \in G^2$ gelten muß.

Mit Gleichung (2) können aus allen lokalen Belegungen die lokal stabilen Belegungen ausgewählt werden. In einer (global) stabilen Konfiguration sind nur lokal stabile Belegungen zugelassen. Bei der untersuchten Funktion Implikation können aber nicht alle möglichen lokal stabilen Belegungen in (global) stabilen Konfigurationen auftreten.

In /1/ und /2/ wurden Stabilitätsbedingungen für die linearen und verallgemeinert linearen Funktionen Antivalenz, Äquivalenz, Konjunktion und Disjunktion bereits untersucht. Für die genannten Funktionen gibt es in jedem Fall stabile Konfigurationen. Für die Zustandsüberführungsfunktion Implikation erhalten wir bei der angenommenen Verknüpfung entsprechend (2)

$$z_0(t+1) = z_0(t)\left[1 + \bigvee_{i=1}^{4} z_i(t)\right] + z_2(t)\left[1 + \bigvee_{i=3}^{4} z_i(t)\right] + z_4(t) \pmod 2, \qquad (3)$$

$$0 = z_0(t) \bigvee_{i=1}^{4} z_i(t) + z_2(t)\left[1 + \bigvee_{i=3}^{4} z_i(t)\right] + z_4(t) \pmod 2. \qquad (4)$$

Die Lösungen von (4) sind die im Bild 4 dargestellten lokal stabilen Belegungen der Zustandsüberführungsfunktion Implikation.

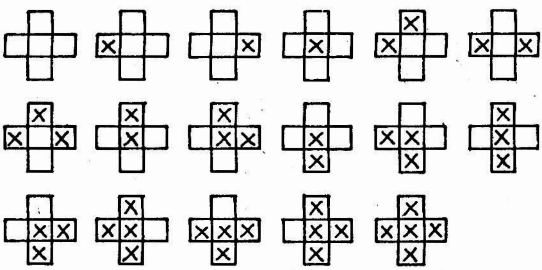


Bild 4

Zur Auswahl und zur Anordnung von lokal stabilen Belegungen, die in (global) stabilen Konfigurationen auftreten können, stellen wir folgende Betrachtungen an:

Wir nehmen zunächst an, daß die Konfiguration nicht leer ist. Es gibt dann also mindestens eine erregte Zelle.

1) Durch Vergleich mit den angegebenen lokal stabilen Belegungen kann man feststellen, daß es keine lokal stabilen Bele-

gungen gibt, bei denen die Zentrumszelle unerregt und die untere Nachbarzelle erregt ist, d. h., es liegt über einer erregten Zelle stets eine weitere erregte Zelle.

2) Liegt eine erregte Zentrumszelle über einer unerregten unteren Nachbarzelle, so lassen sich nur solche Anordnungen von lokal stabilen Belegungen angeben, bei denen mindestens bei einer der seitlichen Nachbarzellen im Widerspruch zu 1) - eine unerregte Zelle über einer erregten Zelle liegt. Es liegt also auch unter einer erregten Zelle stets eine weitere erregte Zelle.

Senkrechte unendliche erregte Zellstreifen sind daher Bestandteil stabiler Konfigurationen. Ebenso ist durch Vergleich mit den lokal stabilen Belegungen leicht zu erkennen, daß auch senkrechte unendliche unerregte Zellstreifen stabil sind. Gibt es in der Konfiguration weitere erregte Zellstreifen mit mindestens einem oberen oder einem unteren Ende, so ist die Konfiguration nicht stabil. Die volle und die leere Konfiguration sind ebenfalls stabil.
Wird eine andere Numerierung der Nachbarzellen vorgenommen, ergeben sich andere stabile Konfigurationen. Als Beispiel sei die im Bild 5 angegebene Numerierung genannt, bei der stabile Konfigurationen aus unendlichen waagerechten unerregten und erregten Zellstreifen bestehen.

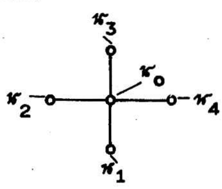


Bild 5

Wird die Klammerung verändert, so ergeben sich weitere stabile Konfigurationen. Beispiel sei die Überführungsfunktion mit der im Bild 5 angegebenen Numerierung und der Klammerung

$$((z_{\varkappa+s_0}(t) \rightarrow z_{\varkappa+s_1}(t)) \rightarrow z_{\varkappa+s_2}(t)) \rightarrow (z_{\varkappa+s_3}(t) \rightarrow z_{\varkappa+s_4}(t)) = z_{\varkappa}(t+1).$$

Bei dieser Oberführungsfunktion sind stabile Konfigurationen alle schachbrettartigen Anordnungen von erregten und unerregten Zellen, die rechts von einem unerregten senkrechten Zellstreifen des Gitters liegen. Links von diesem Zellstreifen gibt es keine erregten Zellen.

3. Oberlagerung von Konfigurationen

Im folgenden Abschnitt untersuchen wir die implikative Oberlagerung von stabilen Konfigurationen auf zellularen Automaten deren (lokale) Zustandsüberführungsfunktion die Implikation ist.

<u>Definition 1:</u> Es seien K_1 und K_2 zwei Konfigurationen und ihre entsprechenden Belegungen $\mathcal{B}^{K_1}$ und $\mathcal{B}^{K_2}$. Dann ist deren Oberlagerung wie folgt erklärt:

$$K = K_1 \rightarrow K_2,$$

$$K = \{\varkappa \mid \varkappa \notin K_1 \vee (\varkappa \in K_1 \wedge \varkappa \in K_2)\},$$

oder

$$\mathcal{B}^{K}(\varkappa) = \mathcal{B}^{K_1}(\varkappa) + \mathcal{B}^{K_2}(\varkappa) \cdot \mathcal{B}^{K_1}(\varkappa) + 1 \pmod 2.$$

Für stabile Konfigurationen gilt ein Superpositionsprinzip, das die Erzeugung von weiteren stabilen Konfigurationen durch die Superposition von stabilen Konfigurationen aus einer gegebenen Menge gestattet.

<u>Satz 2:</u> Sind K_1 und K_2 stabile Konfigurationen eines zellularen Automaten $\mathcal{L}$ mit einer implikativen Zustandsüberführungsfunktion, dann ist K_3 mit $K_1 \rightarrow K_2 = K_3$ eine stabile Konfiguration von $\mathcal{L}$.

<u>Beweis:</u> Aus der Bedingung, daß die lokal stabilen Belegungen in K_1 und K_2 Teile einer (global) stabilen Konfiguration sind, folgt, daß K_3 ebenfalls global stabil ist.

Satz 2 erweitert die in /1/, /2/ erhaltenen Ergebnisse zur Oberlagerung von Konfigurationen mit linearen bzw. verallgemeinert linearen Funktionen auf zellulare Automaten mit einer implikativen Oberführungsfunktion. Wir führen nun den Begriff des Randes einer Konfiguration ein. Dieser Begriff wurde in

Anlehnung an den Begriff des Randes bei Randwertproblemen der Differentialgleichungen gebildet.

<u>Definition 3:</u> Eine Teilmenge $R \subseteq G^2$ der Gitterpunkte wird als Rand einer stabilen Konfiguration K bezeichnet, wenn für jede Belegung $\mathcal{B}_R$ von $R \subseteq G^2$ genau eine Belegung $\mathcal{B}^K$ von $\varkappa \in G^2$ existiert mit

1) $\mathcal{B}^K_R(\varkappa) = \mathcal{B}_R(\varkappa)$ für $\varkappa \in R$, wobei

2) $\mathcal{B}^K(\varkappa)$ die Belegung einer stabilen Konfiguration K ist.

Als Beispiel eines Randes wählen wir einen unendlichen waagerechten Zellstreifen wie im Bild 6. (Die Zellen 3 und 5 gehören nicht zum Zellstreifen.)

```
      3
-------------
    4 1 2
-------------
      5
```

<u>Bild 6</u>

Nun läßt sich für jede Zelle $\varkappa \in R$ eine stabile Belegung finden, die auf den Zellen 1, 2, 4 mit einer lokal stabilen Belegung in einer stabilen Konfiguration übereinstimmt. Damit ist aber auch die Belegung der Zellen 3 und 5 eindeutig bestimmt, d. h., der Zustand der Zellen 1, 3 und 5 ist gleich. Ebenso können die lokalen Belegungen für die anderen Zellen des Randes bestimmt werden. Damit sind die Zustände aller Zellen in den oben und unten an dem Rand anliegenden Zellstreifen eindeutig bestimmt. Bei einer Fortsetzung des Verfahrens wird schrittweise die Belegung der gesamten Konfiguration durch die Belegung des Randes bestimmt.

<u>Satz 4:</u> Für stabile Konfigurationen auf zellularen Automaten mit der von-Neuman-Kopplung und mit der Zustandsüberführungsfunktion Implikation gibt es einen Rand $R \subset G^2$.

<u>Beweis:</u> Für den genannten Fall ist in Bild 3 ein (endlicher) Ausschnitt eines (unendlichen) Randes $R \subset G^2$ gegeben.

Mit der Möglichkeit der Überlagerung von stabilen Konfigurationen zu wiederum stabilen Konfigurationen entsteht die Frage nach einfachen Grundmengen von stabilen Konfigurationen, aus denen durch Überlagerung von Elementen dieser Grundmengen jede stabile Konfiguration erzeugt werden kann. Eine solche Grundmenge nennen wir Erzeugendensystem. Ist sie ein "kleinstes" Erzeugendensystem, so nennen wir sie Basis.

<u>Definition 5:</u> Wir nennen die Menge B aus stabilen Konfigurationen Basis zur Zustandsüberführungsfunktion Implikation des Zellautomaten, wenn

1) für jede stabile Konfiguration K' eine Teilmenge $\tilde{K} \subseteq B$ existiert, so daß $K' = ((K_1 \rightarrow K_m) \rightarrow K_n) \rightarrow \ldots\ldots$ mit $K_i \in \tilde{K}$ $(i=1,m,n \ldots)$ gilt, und

2) für alle $B' \subset B$ Aussage 1) (bei Beschränkung auf Teilmengen $\tilde{K}$ von B') nicht gilt.

Wir geben im folgenden eine Basis an. Dazu nehmen wir an, daß K eine stabile Konfiguration sei und $\mathcal{B}_R^K$ die Belegung dieser Konfiguration auf dem Rand R. Die Zellen des Randes werden in willkürlicher aber fester Weise numeriert. Mit $\mathcal{B}_R^K(j)$ bezeichnen wir die Belegung der j-ten Zelle des Randes. Zu jeder Belegung des Randes gibt es genau eine stabile Konfiguration.
Wir bezeichnen mit K_i die stabile Konfiguration, für die gilt:

$$\mathcal{B}_R^{K_i}(j) = \begin{cases} 1 & \text{für } j = i, \\ 0 & \text{für } j \neq i. \end{cases}$$

Für die stabile Konfiguration B^o gilt

$$\mathcal{B}_R^{B^o}(j) = 0 \text{ für alle } j > 0.$$

<u>Satz 6:</u> Die Menge $B \cup B^o = \{K_i \mid i = 1,2\ldots\} \cup B^o$ ist eine Basis für stabile Konfigurationen auf zellularen Automaten mit der Zustandsfunktion Implikation.

<u>Beweis:</u> Es sei K eine stabile Konfiguration. Ist K die volle Konfiguration, so gilt $K = K_1 \rightarrow K_1$.

Ist K nicht die volle Konfiguration und R ein Rand, so wollen wir zuerst zeigen, daß B ein Erzeugendensystem ist, d. h., jede Konfiguration K ist aus der Überlagerung von Elementarkonfigurationen K_i erzeugbar. Es gilt

$$\mathcal{L}_R^K = (\ldots(((\mathcal{L}_R^{K_l} \rightarrow B^o) \rightarrow \mathcal{L}_R^{K_m}) \rightarrow B^o) \rightarrow \mathcal{L}_R^{K_n} \ldots$$

$(l, m, n, \ldots \in I,\ I = \{i \mid \mathcal{L}_R^K(i) = 1\},\ l, m, n, \ldots$ paarweise verschieden)

und damit wegen der Definition des Randes

$$\mathcal{L}^K = (\ldots(((\mathcal{L}^{K_l} \rightarrow B^o) \rightarrow \mathcal{L}^{K_m}) \rightarrow B^o) \rightarrow \ldots$$

Außerdem ist dieses Erzeugendensystem eine Basis, weil keine Teilmenge ein Erzeugendensystem ist. Es sei K_k mit $k > 0$ eine beliebige stabile Konfiguration, die nicht in $B' \subset B$ enthalten ist. Dann läßt sich aber, da alle anderen Basiskonfigurationen $K_l, K_m, K_n \in B'$ an der Stelle k gleich Null sind, d. h. $K_j(k) = 0$, die Basiskonfiguration K_k mit $K_k(k) = 1$ und $K_k(j) = 0$ nicht durch

$$\mathcal{L}_R^{K_k} = (\ldots(((\mathcal{L}_R^{K_l} \rightarrow B^o) \rightarrow \mathcal{L}_R^{K_m}) \rightarrow B^o) \rightarrow \ldots$$

darstellen.

Die Ergebnisse in diesem Abschnitt stimmen mit den entsprechenden Ergebnissen für lineare und verallgemeinert lineare Funktionen überein (/1/), d. h., auch für die Implikation läßt sich ebenso wie für die linearen bzw. verallgemeinert linearen Funktionen eine Basis angeben.

Abschließend soll noch darauf hingewiesen werden, daß sich die verschiedenen Konfigurationen K_i durch Translation aus einer Konfiguration, z. B. K_1, erzeugen lassen, d. h., die Basis wird durch eine Elementarkonfiguration bestimmt, mit der (durch Translation) alle anderen Elementarkonfigurationen beschrieben werden können. Mit dem Basissystem ist ein einfaches Beschreibungsmittel für stabile Konfigurationen gegeben. Dieses Ergebnis steht in Zusammenhang mit Entscheidungsproblemen, wie sie in der Theorie zellularer Automaten untersucht werden (/4/).

Literatur

/1/ Wolf, G.: Charakterisierung stabiler Konfigurationen auf binären linearen und verallgemeinert linearen zellularen Automaten. Dissertation A, Dresden 1980

/2/ Wolf, G., and Gössel, M.: Stable configurations of linear and generalized linear binary cellular automata. Erscheint in Z. elektr. Informations- und Energietechnik

/3/ Wunsch, G.: Zellulare Systeme. Berlin 1977

/4/ Carstens, H. G.: Entscheidungsprobleme in Zellularräumen. In: Golze, U., und Vollmar, R. (Eds.): Beiträge zur Theorie der Polyautomaten. Braunschweig 1977

eingegangen: 30. 03. 1981

Anschrift des Verfassers:

Dr.-Ing. G. Wolf
Akademie der Wissenschaften der DDR
Zentralinstitut für Kybernetik und
Informationsprozesse
Kurstraße 30/34
DDR-1080 B e r l i n

Rostock. Math. Kolloq. 20, 73 - 83 (1982)

68030
68C25

Bolesław Mikołajczak

On some new NP-complete problems in the theory of finite automata

This paper deals with time computational complexity of several problems associated with simulation of a set of finite automata with help of one automaton.
The concept of realization of one automaton with help of another one assumes that input literals of one automaton can be transformed into input literals of the second one. Here we assume that input words of one automaton can be transformed into input words of the second one. This leads to a wide area of applications in the theory and practice; for instance in a generalized decomposition theory, and also in several problems of simulation of a set of finite automata with help of one automaton.
We present in this paper several results concerning time computational complexity of decision simulation problem for arbitrary automata and also for special subclasses of automata such as: asynchronous, k-asynchronous, planar automata. We assume throughout the paper that automata under consideration are finite, deterministic and completely defined.

Let $A = (S_A, \Sigma_A, M_A)$ and $B = (S_B, \Sigma_B, M_B)$ be two <u>finite heterogenous automata</u> such that $\Sigma_A \cap \Sigma_B = \emptyset$. A pair of functions $\alpha = (\alpha_S, \alpha_\Sigma)$, α_S: S_A into S_B, and α_Σ : Σ_A^* into Σ_B^*, and

$$\alpha_S(M_A(s,x)) = M_B(\alpha_S(s), \alpha_\Sigma(x))$$

for every $s \in S_A$, $x \in \Sigma_A^*$, $\alpha_\Sigma(x) \neq \lambda_B$, $\lambda_B \notin \Sigma_B$ is said to be a <u>generalized homomorphism</u> of A into B. The set of all state homomorphisms will be denoted as SHom(A ⟶ B), and the set of all homomorphisms with $\alpha_\Sigma : \Sigma_A$ into Σ_B will be denoted by Hom(A ⟶ B). We assume that GHom(A ⟶ B) = ∅ denotes the fact that there exists only the <u>trivial generalized homomorphism</u>, i. e. such that there exists a state $t \in S_A$ such that for every $s \in S_A$, it follows $\alpha_S(s) = t$, and in a pair of functions $(\alpha_S, \alpha_\Sigma)$, function α_Σ is arbitrary, respectively chosen for α_S and such that $\alpha_\Sigma(\sigma) = x$, where $M_B(t,x) = t$.
The set of all trivial generalized homomorphisms will be denoted by TGHom(A ⟶ B). A set of all <u>nontrivial generalized homomorphisms will be denoted by</u>

NTGHom(A ⟶ B) = GHom(A ⟶ B) \ Hom(A ⟶ B).

The algorithm of computing all nontrivial generalized homomorphisms NTGHom(A ⟶ B) has been presented in /19/.

An automaton $A' = (S_A', \Sigma_A', M_A')$ is said to be <u>subautomaton</u> of $A = (S_A, \Sigma_A, M_A)$ with input free monoid Σ_A^* iff $S_A' \subseteq S_A$, $\Sigma_A' \subseteq \Sigma_A^*$, M_A' is a restriction of M_A to $S_A' \times \Sigma_A'$, and for arbitrary $s' \in S_A'$ we have $M_A'(s',x) \in S_A'$.

For automaton $A = (S_A, \Sigma_A, M_A)$ two kinds of subautomata are of special interest, namely $A^i = (S_A, \Sigma_A^i, M_A^i)$ and

$A_d^i = (S_A, \Sigma_{d,A}^i, M_{d,A}^i)$,

where $\Sigma_A^i = \{\sigma_1 \sigma_2 \dots \sigma_i : \sigma_1, \sigma_2, \dots, \sigma_i \in \Sigma_A\}$,

$$\Sigma_{d,A}^i = \{\sigma\sigma\dots\sigma = \sigma^i : \sigma \in \Sigma_A\}$$

and $M^i_A : S_A \times \sum^i_A \rightarrow S_A$, and $M^i_{d,A} : S_A \times \sum^i_{d,A} \rightarrow S_A$ are restrictions of M_A. These subautomata are called <u>subautomata associated with the change of operating time</u>.

Now we will formulate a set of problems, called SIMULATION PROBLEMS. Before this we quote several definitions needed in what follows.

A decision problem π is defined as an ordered pair $\pi = (D_\pi, Y_\pi)$, where D_π is a set of instances of a problem called domain, and Y_π is a subset of D_π, i. e., the set of instances for which the answer is "yes". A subproblem sub-π of a given problem $\pi = (D_\pi, Y_\pi)$ is an ordered pair (D'_π, Y'_π), where $D'_\pi \subseteq D_\pi$ and $Y'_\pi = Y_\pi \cap D'_\pi$.

Under P we understand a class of decision problems solvable in polynomial time on deterministic Turing machines. Under NP we understand a class of decision problems solvable in polynomial time on nondeterministic Turing machines.

Problem $\pi = (D_\pi, Y_\pi)$ is polynomially reducible to problem $\pi' = (D'_\pi, Y'_\pi)$ iff there exists a function $f : D_\pi \rightarrow D'_\pi$ such that it transforms every instance I of D_π into instance $f(I) \in D'_\pi$ in polynomial time and such that $I \in Y_\pi$ iff $f(I) \in Y'_\pi$.

We say that problem π is NP-complete iff $\pi \in$ NP and every problem $\pi' \in$ NP is polynomially reducible to π. We can also prove NP-completeness of some problem π through showing that $\pi \in$ NP and besides that there exists another problem $\pi' \in$ NP-complete such that π' is polynomially reducible to π.

Now we quote the concept of a class of problems of intermediate difficulty between the classes P and NP with respect to time computational complexity; namely

$$NPI = NP \setminus (P \cup NPC)$$

with the assumtion that NPI $\neq \emptyset$ if P $\neq$ NP. One of the well-known among NPI problems is the vertex isomorphism of directed or undirected graphs. This problem have withstood the test of time and have certain attributes that seem to distinguish it from the types of problems already known to be NP-complete

(NPC). In this paper we show that the problem of finding generalized isomorphism between two heterogenous automata belongs also to NPI.

A <u>search problem</u> π consists of a set D_π of finite objects called instances and, for each instance $I \in D_\pi$, a set of $S_\pi(I)$ of finite objects called solutions for I. An algorithm is said to solve a search problem π if, given as input any instance $I \in D_\pi$, it returns the answer "no" whenever $S_\pi(I)$ is empty and otherwise it returns some solution s belonging to $S_\pi(I)$.

An enumeration problem based on the search problem π is: "Given I, what is the cardinality of $S_\pi(I)$, that is, how many solutions are there?"

An <u>optimization problem</u> is either a <u>minimization problem</u> or <u>maximization problem</u> and consists of the following three parts:

i) a set of instances, D_π,
ii) for each instance $I \in D_\pi$, à finite set $S_\pi(I)$ of candidate solutions for I, and
iii) a function m_π that assigns to each instance $I \in D_\pi$ and each candidate solution $s \in S_\pi(I)$ a positive rational number $m_\pi(I,s)$, called the solution value for π.

Now we can formulate a fundamental decision problem associated with generalized homomorphisms.

GHOM EMPTINESS PROBLEM

I. There are given two finite automata $A = (S_A, \Sigma_A, M_A)$ and

$B = (S_B, \Sigma_B, M_B)$.

Q. Is the set of nontrivial generalized homomorphisms from A into B empty, i. e. $NTGHom(A \rightarrow B) = \emptyset$?

Of course, this problem is decidable; but the algorithm presented in /19/ is unefficient because it includes a computation of characteristic semigroup of automaton B, and a computation of state homomorphisms from A into (onto) B.

For some subclass of permutation automata there exists an effective method of solution of GHOM EMPTINESS PROBLEM. This follows from the following lemma.

<u>Lemma 1:</u> Let $A = (S_A, \Sigma_A, M_A)$ and $B = (S_B, \Sigma_B, M_B)$ be two permutation automata; then $\overline{I(A)} = \overline{I(B)}$ implies $NTGHCM(A \rightarrow B) = \emptyset$.

Thus in general the automata which generate the same group are not related by isomorphisms or homomorphisms even if we permit a change of the input words. For permutation automata the problem of GHOM EMPTINESS PROBLEM is solvable in pseudoexponential time due to the fact that a number of generators in a group is of order O(log n). Therefore the problem of verifying the equality of $\overline{I(A)}$ and $\overline{I(B)}$ has a time computational complexity of order $O(n^{\log n})$, where n is the number of elements in $\overline{I(A)}$ or $\overline{I(B)}$.
Now we are in a good position to formulate our simulation problems.

1. Automata simulation decision problem

I. Let $\mathfrak{A} = \{A_1 = (S_1, \Sigma_1, M_1),$

$$A_2 = (S_2, \Sigma_2, M_2), \ldots, A_k = (S_k, \Sigma_k, M_k)\}$$

be a set of finite automata.

Q. Does there exist a finite automaton $A = (S, \Sigma, M)$ such that for every automaton $A_j \in \mathfrak{A}$ there exists a subautomaton $A^i = (S, \Sigma^i, M^i)$ of automaton A such that $NTGHom(A_j \rightarrow A^i) \neq \emptyset$?

2. Automata simulation search problem

I. Let $\mathfrak{A} = \{A_1 = (S_1, \Sigma_1, M_1),$

$$A_2 = (S_2, \Sigma_2, M_2), \ldots, A_k = (S_k, \Sigma_k, M_k)\}$$

be a set of finite automata, and let K be a natural number.

Q. Compute all automata $B_1 = (T_1, \Sigma_1, M_1)$ such that $|T_1| \leq K$ and for every automaton $A_j \in \mathcal{A}$ there exists a natural number $i > 1$ such that $NTGHom(A_j \to B_1^i) \neq \emptyset$.

3. Automata simulation enumeration problem

I. Let $\mathcal{A} = \{A_1 = (S_1, \Sigma_1, M_1),$
$A_2 = (S_2, \Sigma_2, M_2), \ldots, A_k = (S_k, \Sigma_k, M_k)\}$
be a set of finite automata.

Q. How many automata B_1 there exist such that $NTGHom(A_j \to B_1^i) \neq \emptyset$?

4. Automata simulation optimization problem

I. Let $\mathcal{A} = \{A_1 = (S_1, \Sigma_1, M_1),$
$A_2 = (S_2, \Sigma_2, M_2), \ldots, A_k = (S_k, \Sigma_k, M_k)\}$
be a set of finite automata.

Q. Compute automaton $B = (T, \Sigma, M)$ such that for every automaton $A_j \in \mathcal{A}$ there exists a natural number $i > 1$ and $NTGHom(A_j \to B^i) \neq \emptyset$ and $|T|$ is minimal (or the cardinality of a characteristic semigroup $\overline{I(B)}$ is minimal).

As a special case of the above presented problems, a set of problems can be defined in which we deal with subautomata A_d^i. In what follows we will discuss simulation problems associated with subautomata A^i only, because problems associated with subautomata A_d^i are subproblems of problems associated with subautomata A^i.

All these problems are of great practical importance because the set of automata can be "realized" by the help of one automaton, an additional multichannel clock with different frequencies being multiplicity of some fundamental frequency, and a sequential transducer mapping input symbols of $\sum_i$ into words of $\sum^+$. From technical point of view these problems can also be treated as generalized decomposition problems for a set of automata with common tail automaton B and different front automata (sequential transducers mapping input symbols into words).

Now we present several results associated with time computational complexity of the above presented problems.

<u>Theorem 1:</u> Automata simulation decision problem is NP-complete.

<u>Proof:</u> For a given set of automata we choose nondeterministically an automaton B with 2,3,...,n,... states, respectively; then we generate all automata of fixed number of states. Especially as an automaton B we can choose the universal automaton, i. e. such automaton that its characteristic semigroup is composed of all possible transformations on the set of states T. From practical point of view we restrict our considerations to such n that $n \le \max\{|S_1|, |S_2|, \ldots, |S_k|\}$.

Then by the help of the algorithm presented in /19/ we verify whether the set of nontrivial homomorphisms $NTGHom(A \to B^i) \neq \emptyset$ for every $j \in \{1, 2, \ldots, k\}$ and such i that $i = \max(T_{min})$, where

$$T_{min}(B) = \bigcup_{j'=1}^{p} \mathrm{Min}\ \bar{\bar{x}}_{j'}, \text{ where } \mathrm{Min}\ \bar{\bar{x}}_{j'} = y, \text{ and}$$

$$y = \min\{|z| : z \in \bar{\bar{x}}_{j'}\};\ \overline{I(B)} = \{\bar{\bar{x}}_1, \bar{\bar{x}}_2, \ldots, \bar{\bar{x}}_p\}.$$

According to /8/ the problem of finding homomorphism between two words on finite alphabets is polynomial in time. Therefore using the restriction technique /10/ we have reduced our problem to the problem of finding state homomorphisms between two finite automata (or equivalently, vertex homomorphisms between two directed graphs). This last problem is NP-complete according to /1/, /9/, /10/, /22/.

Corollary 1: The automata simulation decision problem with generalized isomorphisms from A_j into B^i is also NP-complete.

Corollary 2: The automata simulation decision problem with generalised homomorphisms from A_j onto B^i is also NP-complete.

Theorem 2: The automata sumulation decision problem with generalized isomorphisms from A_j onto B^i is NPI.

Proof: It follows from the fact that the problem of finding isomorphism between two words on finite alphabets is polynomial in time /8/. And therefore our problem has been reduced to the problem of state isomorphism from A_j onto B^i, due to restriction technique /10/; this last problem is NPI.

Theorem 3: The automata simulation search problem with respect to generalized onto isomorphisms is NPI.

Proof: The decision problem whether two directed graphs are state (vertex) onto isomorphic and search problem of isomorphism between such graphs are polynomially equivalent according to /18/.

Theorem 4: The automata simulation decision problem with generalized isomorphisms from A_j into B^i is NP-complete.

Proof: Through a restriction technique from the CLIQUE problem.

Now we restrict our problems to some subproblems as a way of looking for time polynomial subproblems of given problems. We will discuss three special classes of automata: asynchronous, k-asynchronous and planar.

Automaton $A = (S, \sum, M)$ is said to be asynchronous if for every $s \in S$ and for every $\sigma \in \Sigma$, $M(s, \sigma) = M(s, \sigma\sigma)$.

Automaton $A = (S, \Sigma, M)$ is said to be <u>k-asynchronous</u> if for every $s \in S$ and every $\sigma \in \Sigma$ there exists natural number $k > 1$ such that $M(s, \sigma^k) = M(s, \sigma^{k+1})$.

Automaton $A = (S, \Sigma, M)$ is said to be <u>planar</u> if its transition graph is planar.

<u>Lemma 2:</u> Let $A = (S, \Sigma, M)$ be k-asynchronous or an asynchronous automaton with $|S| = n$; then the cardinality of the characteristic semigroup $\overline{I(A)}$ has a time computational complexity equal to $O(n^k)$ and $O(n)$, respectively.

<u>Proof:</u> Follows from the results included in /19/.

<u>Lemma 3:</u> Let $A = (S, \Sigma, M)$ be a planar automaton; then every subautomaton A^i of A is also planar.

Now we are in a position to state our next result.

<u>Theorem 5:</u> Automata simulation decision and search problems for asynchronous, k-asynchronous and planar automata are polynomial in time.

<u>Proof:</u> It follows from lemmas 2 and 3, and besides from results included in /7/, /8/, /14/, /15/.

<u>References</u>

/1/ Barrow, H. G., and Burstall, R. M.: Subgraph isomorphism, matching relations and maximal cliques. Inform. Process. Lett. <u>4</u>, 83 - 84 (1975)

/2/ Booth, K. S.: Isomorphism testing for graphs, semigroups and finite automata are polynomially equivalent problems. SIAM J. Comput. <u>7</u>, 273 - 279 (1978)

/3/ Buda, A.: Generalized sequential machine maps. Inform. Process. Lett. 8, 38 - 41 (1979)

/4/ Busacker, R., and Saaty, T.: Finite Graphs and Networks. New York 1965

/5/ Colbourn, M. J., and Colbourn, C. J.: Graph isomorphism and selfcomplementary graphs. SIGACT News 10, 25 - 30 (1978)

/6/ Cook, S.: The complexity of theorem proving procedures. In: Proc. 3rd Annual ACM Symposium on Theory of Computing, pp. 151 - 158, New York 1971

/7/ Corneil, D. G.: Graph isomorphism. Tech. Report N°18. Department of Computer Science, University of Toronto, Toronto 1970

/8/ Ehrenfeucht, A., and Rozenberg, G.: Finding a homomorphism between two words is NP-complete. Inform. Process. Lett. 9, 86 - 88 (1979)

/9/ Garey, M. R., and Johnson, D. S.: Some simplified NP-complete graph problems. Theoret. Comput. Sci. 1, 237 - 267 (1976)

/10/ Garey, M. R., and Johnson, D. S.: Computers and intractability, A guide to the theory of NP-completeness. San Francisco 1979

/11/ Grzymała-Busse, J. W.: Operation-preserving functions and autonomous factors of finite automata. J. Comput. System Sci. 5, 465 - 474 (1971)

/12/ Hirschberg, D., and Edelberg, M.: On the complexity of computing graph isomorphism. Tech. Report N° 130, Computer Science Laboratory, Department of Electrical Engineering, Princeton University, N. J. 1973

/13/ Holt, R. C., and Reingold, E. M.: On the time required to detect cycles, and connectivity in graphs. Math. Systems Theory 6, 103 - 106 (1972)

/14/ Hopcroft, J., and Wong, A.: Linear time algorithm for isomorphism of planar graphs. In: Constable, R. L. (Ed.): Sixth Annual ACM Symposium on Theory of Computing, Seattle 1974, pp. 172 - 184, New York 1976

/15/ Hopcroft, J., and Tarjan, R.: Isomorphism of planar graphs. In: Miller, R. E., and Thatcher, J. W. (Eds.): Complexity of Computer Computations, pp. 131 - 152, New York 1972

/16/ Kozen, D.: A clique problem equivalent to graph isomorphism. Private communication

/17/ Levin, L. A.: Universal sorting problems. Problemy Peredači Informacii 9, 111 - 116 (1973)

/18/ Mathon, R.: A note on the graph isomorphism counting problem. Inform. Process. Lett. 8, 131 - 132 (1979)

/19/ Mikołajczak, B.: Generalized functions preserving operations of finite automata. Found. Control Engrg. to appear

/20/ Miller, G. L.: Graph isomorphism, general remarks. In: Proc. 9th Annual ACM Symposium on Theory of Computing, pp. 143 - 150, New York 1977

/21/ Miller, G. L.: On the $n^{\log n}$ isomorphism technique, a preliminary report. In: Proc. 10th Annual ACM Symposium on Theory of Computing, pp. 51 - 58, New York 1978

/22/ Nieminen, J.: On homomorphic images of transition graphs. Inform. Process. Lett. 4, 14 - 15 (1975)

received: March 30, 1981

Author's address:

Dr. habil. B. Mikołajczak
Institut of Control Engineering
TU Poznań
Poznań, VR Polen

Rostock. Math. Kolloq. 20, 85 - 96 (1982) 68D90
68C15

Hans-Dieter Burkhard

On Fairness in Petri Nets

Fairness is to be understood as performing actions in the order of their announcements. Related problems are studied in the Petri net model where actions correspond to the firings of transitions. An action is announced if a transition becomes firable, it is performed if this transition fires. Fairness in Petri nets means firing of transitions in the order of their enabling. Therefore, fairness is a property of firing sequences: some of them satisfy fairness conditions while others do not and have to be excluded. To ensure fairness the firings must be controlled by a queue regime giving temporary priorities to the transitions with longest waiting time. The queue policy can be performed by finite automata controlling the firings of the Petri net.

While in other fairness considerations the performability of a waiting action is not influenced by other actions, the situation in Petri nets may be different: a firable transition may loose its concession by firings of other transitions, i. e., an announced action may be non-performable when it should be executed with respect to its waiting period. This leads to certain constraints for the net (to permit fair firings) or to modified fairness conditions.

The problems like reachability, boundedness, liveness etc. are considered for nets firing under fairness conditions. They can be proved to be undecidable in the case of unbounded nets since we can simulate counter machines by Petri nets where the firings are constrained by fairness. Thus fairness gives the Petri nets more computational power and hence in general it is not possible to modify a Petri net in such a way that the possible firings are exactly the fair firings of the original net.

1. Basic notions

A (generalized initial) Petri net is given by

$$N = (P, T, \{t^- \mid t \in T\}, \{t^+ \mid t \in T\}, m_o)$$

with finite disjoint sets P and T of places and transitions, respectively, and with the initial marking $m_o \in \mathbb{N}^P$. The vectors t^-, $t^+ \in \mathbb{N}^P$ determine the change of markings by firings of transitions:

A transition $t \in T$ is firable at the marking $m \in \mathbb{N}^P$ iff $t^- \leq m$, after its firing the new marking is $m + \Delta t$ with $\Delta t := t^+ - t^-$.

A sequence $u = t_1 t_2 \ldots t_n \in T^*$ is firable at m iff each t_i is firable at $m + \Delta t_1 + \Delta t_2 + \ldots + \Delta t_{i-1}$ $(i=1,\ldots,n)$. The firing of u leads to the new marking $m + \Delta u$ with $\Delta u := \Delta t_1 + \Delta t_2 + \ldots + \Delta t_n$.

L_N denotes the set of all transition sequences firable at the initial marking m_o ("firing sequences").

R_N denotes the set of reachable markings: $R_N := \{m_o + \Delta u \mid u \in L_N\}$.

A place $p \in P$ is bounded $:=_{DF} \exists k \in \mathbb{N} \;\; \forall m \in R_N : m(p) \leq k$.

A transition $t \in T$ is live $:=_{DF} \forall u \in L_N \; \exists v \in T^* : uvt \in L_N$.

The Petri net N is bounded (live) iff all places are bounded (all transitions are live).

N is persistent $:=_{DF} \forall u \in T^* \; \forall t_1, t_2 \in T : ut_1, ut_2 \in L_N \rightarrow ut_1 t_2 \in L_N$.

2. Fairness definitions

The waiting time w(u,t) of the transition $t \in T$ after the firing of $u = t_1 \ldots t_n \in L_N$ describes the maximum number of firings after a point of time where t was enabled but not fired in u for the subsequent time:

$$w(u,t) := \mathrm{Max}(\{k \mid t^- \leq m_o + \Delta t_1 \ldots t_{n-k} \wedge \bigwedge_{j=0}^{k-1} t \neq t_{n-j}\} \cup \{-1\}).$$

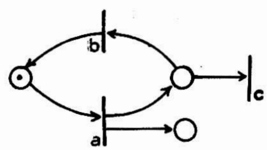
The "-1" denotes the fact that t is not waiting: t was not enabled after its last firing or even from the beginning.

Let $t \in T$ be firable in $m_o + \Delta u$ for $u \in L_N$. Then the firing of t after u is called fair iff $w(u,t) \geq w(u,t')$ for all $t' \in T$.

Since the transitions with maximal waiting time might be non-firable at $m_o+\Delta u$ (cf. example 1) we consider a "weak fairness", too. The firing of t after u is called weakly fair iff $w(u,t) \geq w(u,t')$ for all $t' \in T$ which are firable in $m_o+\Delta u$.

A firing sequence $u = t_1 \ldots t_n \in L_N$ is called (weakly) fair iff the firing of t_i after $t_1 \ldots t_{i-1}$ is (weakly) fair for all $i = 1, \ldots, n$. The set of all (weakly) fair firing sequences of the net N is denoted by L_N^f (L_N^{wf}).

Example 1:



$L_N = (ab)^* \cdot \{a, ac, \lambda\}$,

$L_N^{wf} = \{a, ab, aba, abac, ac\}$,

$L_N^f = \{a, ab, ac\}$.

Observation 1: There are nets N with $L_N^f \subsetneq L_N^{wf} \subsetneq L_N$.

Proposition 1: $L_N^f \subseteq L_N^{wf} \subseteq L_N$ for each Petri net N.

The first part can be proved by induction over the length of the sequences, while the second part is obvious. Furthermore we have by the definitions:

Proposition 2: The sets L_N^f, L_N^{wf}, L_N are decidable.

3. Temporary priorities

(Weakly) Fair firings might be considered as firings under temporary priorities where the priorities depend on the history of firings which are described by a firing sequence u. Thus we have a binary priority relation $\underset{u}{\succ}$ over T with

$$t \underset{u}{\succ} t' \text{ iff } w(u,t) > w(u,t')$$

and an equivalence relation $\underset{u}{\sim}$ over T with

$$t \underset{u}{\sim} t' \text{ iff not } t \underset{u}{\succ} t' \text{ and not } t' \underset{u}{\succ} t \text{ (iff } w(u,t) = w(u,t')).$$

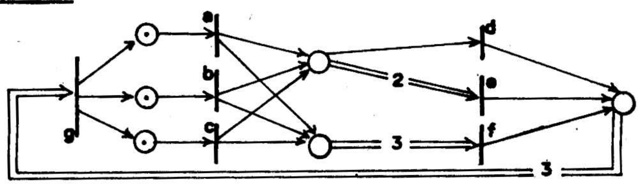
By the factorization $T/\underset{u}{\sim}$ we have classes of transitions with the same priority (with the same waiting time). Fair firing means firing of a transition from the highest priority class of $T/\tilde{u}$. For weak fairness we have to regard only the set T_m of transitions which are firable at the actual marking m:

$T_m := \{t \mid t^- \leq m\}$.

Weakly fair firing means firing of a transition from the highest priority class of $T_{m_o + \Delta u}/\tilde{u}$.

<u>Observation 2:</u> The number of priority classes may increase and decrease during the firing of a (weakly) fair firing sequence:

<u>Example 2:</u>



In this example we have (λ denotes the empty word):

$T/\tilde{\lambda}$: $\{a,b,c\} \succ_{\lambda} \{d,e,f,g\}$,

$T/\tilde{u}$: $\{d\} \succ_{u} \{e\} \succ_{u} \{f\} \succ_{u} \{a,b,c,g\}$,

$T/\widetilde{uv} = T/\tilde{\lambda}$, where u is a firing sequence which contains a,b,c in an arbitrary order, and v = defg.

Moreover, with respect to (weak) fairness we have a "non-deterministic" situation in the beginning: there are three firable transitions (a,b,c) which are allowed to fire in an arbitrary order. But the situation is a "deterministic" one after their firings: the three firable transitions d,e,f are allowed to fire in the order d - e - f, only. Note that (weak) fairness leads to this fixed ordering in each case - independently of the chosen order for the firings of a,b,c. Thus we have:

<u>Observation 3:</u> By (weakly) fair firings it is possible to enforce a fixed ordering for the firable transitions.

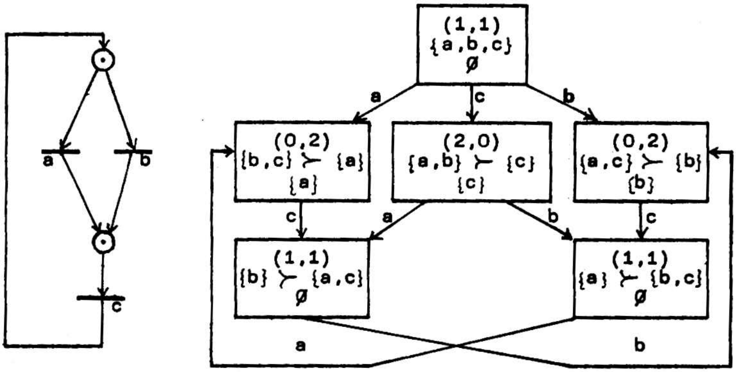
4. (Weak) Fairness marking graph

Possible firings in Petri nets are determined by the actual marking. To satisfy (weak) fairness, possible firings must regard the actual priorities, too. Hence the usual marking graph is not sufficient to study the behaviour. Instead of the marking graph we consider a (weak) fairness marking graph where the nodes are triples $(m,\succ,U)$ such that m is the actual marking, $\succ$ is the actual priority relation and U is the set of "unannounced actions", i. e., of transitions with the waiting time -1. The initial node is given by $(m_o,\succ_o,U_o)$ with $U_o = T \setminus T_{m_o}$ and $t' \succ_o t''$ for $t' \in T_{m_o}$, $t'' \in U_o$. An arc labelled by t leads from $(m,\succ,U)$ to $(m',\succ',U')$ iff

- t is a firable transition from the highest priority class of $T/\sim$ (of $T_m/\sim$ in the case of weak fairness),
- $m' = m + \Delta t$,
- $U' = (U \cup \{t\}) \setminus T_{m'}$,
- $\succ' = (\succ \setminus (\{t\} \times T)) \cup ((T \setminus U) \times \{t\}) \cup ((T \setminus U') \times U')$.

Note that the information "U" is necessary to identify the not announced transitions (the class with lowest priority is identical to U only if not all transitions are announced).
The (weak) fairness marking graph of N contains all nodes which are reachable by the directed arcs from the initial node.
A sequence $u \in L_N^f$ (L_N^{wf}) corresponds to a path in the (weak) fairness marking graph which starts in the initial node and which is labelled by u. The (weak) fairness marking graphs may be infinite just like usual marking graphs, but they may be finite in some cases where the usual marking graphs are infinite (cf.5.). Note that a "converability tree" /KM/ with related properties like for usual firings does not exist for (weakly) fair firings, cf.5.

Example 3 with (weak) fairness marking graph:



5. Reachability, Boundedness, Liveness, Persistency

We consider these problems for nets in which only (weakly) fair firings are permitted. Thus the set of firing sequences is restricted to L_N^f (L_N^{wf}), and therefore the sets of reachable markings are restricted, too: $R_N^f = \{m_o + \Delta u \mid u \in L_N^f\}$, $R_N^{wf} = \{m_o + \Delta u \mid u \in L_N^{wf}\}$. Boundedness, liveness and persistency with respect to (weak) fairness are then defined like in 1., where L_N^f/R_N^f (L_N^{wf}/R_N^{wf}) are used instead of L_N/R_N.

Proposition 3: 1. $R_N^f \subseteq R_N^{wf} \subseteq R_N$.

2. If N is bounded, then N is bounded with respect to (weak) fairness.
3. If N is persistent, then N is persistent with respect to (weak) fairness.

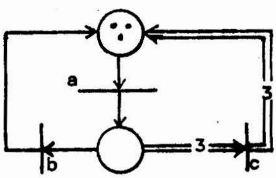
Proof: 1., 2. follow from proposition 1. Persistency: If we have $ut', ut'' \in L_N^f$ (L_N^{wf}), then t', t" are in the same priority class of $T/\underset{u}{\sim}$. Hence $ut't'' \notin L_N^f$ is only possible if $ut't'' \notin L_N$. If $ut't'' \notin L_N^{wf}$ then $ut't'' \notin L_N$ or there must exist some t with

$t \succ_u t'$, $t \succ_u t''$, t not firable at $m_o + \Delta u$ (but again firable at $m_o + \Delta ut'$). Since t is announced but not firable at $m_o + \Delta u$, it must have lost its concession by other firings and hence N is not persistent.

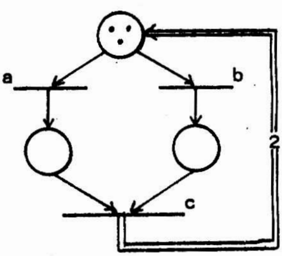
<u>Observation 4:</u>

1. There are nets N with $R_N^f \subsetneq R_N^{wf} \subsetneq R_N$ (exemple 1).
2. There are nets bounded with respect to (weak) fairness which are not bounded (example 1).
3. There are nets persistent with respect to (weak) fairness which are not persistent (example 4).
4. There are nets live with respect to (weak) fairness which are not live (exemple 5).
5. There are nets not live with respect to (weak) fairness which are live (example 4).

<u>Example 4:</u>



<u>Example 5:</u>



<u>Proposition 4:</u> The net N is bounded with respect to (weak) fairness iff the (weak) fairness marking graph is finite.

<u>Proof:</u> The graph is finite iff R_N^f (R_N^{wf}) is finite.

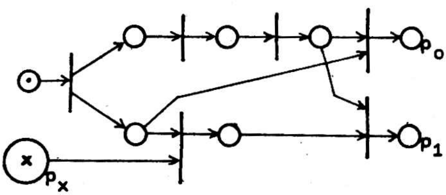
Now we consider four nets where only (weakly) fair firings are permitted: the reachability problem: "$m \in R_N^f$ (R_N^{wf})?", the boundedness problem: "p/N bounded with respect to (weak) fairness?", the liveness problem: "t/N live with respect to (weak) fairness?" and the persistency problem: "N persistent with respect to (weak) fairness?".

Proposition 5: The reachability, boundedness, liveness and persistency problems are decidable for Petri nets where only (weakly) fair firings are permitted if the (weak) fairness marking graphs are finite.

Proof: The corresponding graphs can be checked.

Proposition 6: The reachability, boundedness, liveness and persistency problems are in general undecidable for Petri nets where only (weakly) fair firings are permitted.

Proof: Satisfying (weak) fairness in a Petri net N it is possible to simulate deterministic counter machines. For lack of space we must omit details of the proof (cf./B1/ where "Queue 3" corresponds to weak fairness). The most important point is the simulation of zero-testing for unbounded places:



In this construction by (weakly) fair firings a token reaches place p_o if and only (!) if the counterplace p_x is empty (if $x=0$). A token comes to p_1 iff $x > 0$.

The undecidability of the halting problem for deterministic counter machines implies the undecidability of the mentioned problems (for corresponding constructions cf. /B1/).

Since they are able to simulate deterministic counter machines, Petri nets satisfying (weak) fairness have more computational power than usual Petri nets (cf. /B1/). Hence we have:

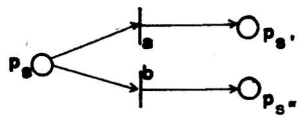
Proposition 7: In general it is not possible to simulate (weakly) fair firings by usual firings in a modified Petri net, exactly.

Remarks: 1) By the extended computational power, there are sets L_N^f (L_N^{wf}) different from all sets $L_{N'}$ of arbitrary Petri nets N'. On the other hand there are sets L_N which are different from all sets $L_{N'}^f$ ($L_{N'}^{wf}$) like $L_N = \{a,b\}^*$ for instance.

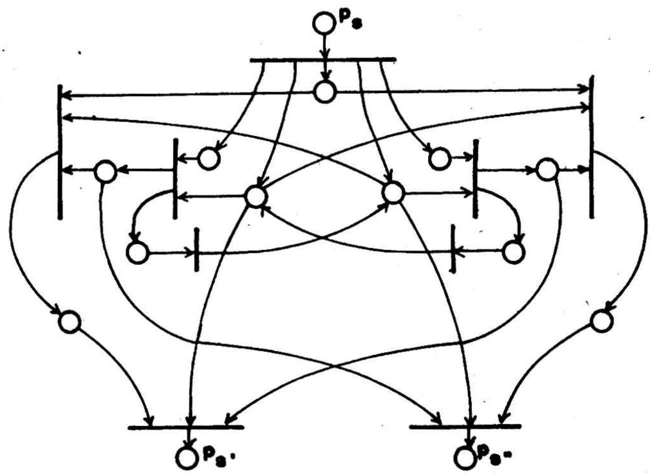
2) Using transition labelling functions (homomorphisms $h: T \to \Sigma \cup \{\lambda\}$, Σ a finite alphabet) and a terminal marking in Petri nets under (weakly) fair firings it is possible to generate all recursively enumerable languages by the simulation of non-deterministic counter machines (cf. /B1/). Note that the simulation of the non-deterministic choice instruction

s: goto s' or s"

cannot be performed by two conflict transitions a,b



since (weak) fairness would enforce alternation of a and b while the choice should be performable arbitrarily often. Nevertheless the non-deterministic choice can be realized in Petri nets where only (weakly) fair firings are permitted by the following construction:



6. Control

To ensure (weak) fairness the usual firing of a Petri net must be restricted. Therefore the firings have to be controlled by a queue regime, for instance, corresponding to the changing priorities (cf. /B1/). The control can be performed by a finite automaton where the states are pairs $(\succ, U)$ of actual priority relations $\succ$ and actual unannounced transition sets U. The input coming from the net with the actual marking m is the set T_m of transitions firable at m. The output is a transition which is to be fired in the net: for ensuring (weak) fairness the output transition is a firable transition from the highest priority class of T (T_m). If such a transition does not exist, the automaton has to stop the work of the net. The necessary information is known to the automaton by its state and the input. To reflect the new situation after the firing of the selected transition the automaton changes its state regarding the old state, the input and the output. Such automata are in general non-deterministic. Special (weakly) fair firing sequences are realized by deterministic automata with related properties /B2/. We have:

Proposition 8: (Weakly) Fair firings can be ensured in a Petri net under the control of finite automata. Such automata can be constructed.

7. Blocking by fairness

We say that the net N is blocked by a fair firing sequence $u \in -_N^f$ iff there exist some transitions $t \in T$ with $ut \in L_N$ but $ut' \notin L_N^f$ for all $t' \in T$. Thus blocking means that firing is possible but only by violating fairness (cf. the example 1: the net is blocked by $u = ab$). A net is blockable iff there exists a blocking sequence $u \in L_N^f$.

Proposition 9: $L_N^f = L_N^{wf}$ iff N is not blockable.

Proof: If the net is blocked by u, then we have some $t \in T$ with $ut \in L_N^{wf} \setminus L_N^f$ since certain weakly fair firings must be possible if some firings are possible.

If $L_N^f \neq L_N^{wf}$, then there must exist $ut \in L_N^{wf} \setminus L_N^f$ with $u \in L_n^{wf} \cap L_N^f$. It means that the firing of t after u is weakly fair but not fair. Hence all transitions in the highest priority class of $T/\widetilde{u}$ are not firable in $m_o + \Delta u$, such that fair firings are impossible and the net is blocked by u.

Proposition 10: If N is persistent with respect to fairness, then it is not blockable.

Proof: For blocking the appearance of conflicts is necessary.

Observation 5: Persistency with respect to fairness is not necessary for a net to be not blockable (see example 3 where $a,b \in L_N^f$ but $ab \notin L_N^f$).

Using appropriate constructions in connection with the simulation of deterministic counter machines (cf.7.) it can be shown:

Proposition 11: It is not decidable whether a net is blockable. (It is decidable for nets with finite fairness marking graphs.)

8. Conclusions

Problems in Petri nets become harder under the restrictions by (weakly) fair firings: the simulation of deterministic counter machines is possible even if no conflicts appear - the construction given in 5. is persistent with respect to (weak) fairness. Thus (weak) fairness is one of the examples where a control of firings extends the computational power of Petri nets.

References

/KM/ Karp, R. M., and Miller, R. E.: Parallel Program Schemata. J. Comput. System Sci. 3, 147 - 195 (1969)

/B1/ Burkhard, H.-D.: Ordered firing in Petri nets. To appear in Elektron. Informationsverarb. Kybernet.

/B2/ Burkhard, H.-D.: Environment relations between Petri nets and finite deterministic automata. (Manuscript)

received: March 30, 1981

Author's address:

Dr. H.-D. Burkhard
Humboldt-Universität Berlin
Sektion Mathematik
Unter den Linden 6
DDR-1086 Berlin

Rostock. Math. Kolloq. 20, 97 - 115 (1982) 68G10
68C25

Reinhard Klette

Time Complexity of Digital Image Processing Problems

In the present paper, for selected problems drawn from the digital image processing area (/10/, /11/) different possible ways of computer realization are discussed where the time needed by the programs is used as a measure for effectiveness. For these discussions two models of computation are underlain: the random access machine (RAM) in the sense of /1/ and a special array processor called parallel matrix processing system (PMS), for further architectures of parallel processing systems reference is made to /7/. - The present paper reports about some parts of /5/ in a brief way.

1. Time Complexity of Computational Problems

The models of computation, RAM and PMS, which are yet defined in the following are characterized by a certain set of basic operations (basic instructions) which are available in these models, in both cases. We assume each basic operation needs one unit of time, the *uniform cost criterion*. Thus the time of a computation is equal to the number of basic instructions that has to be carried out where we agree upon that input or output operations are leaved out of count, however.
The time complexity of a program is taken as the maximum complexity over all inputs of a given size known as *worst-case complexity*. The input size may be the size N of the considered quadratic NxN images, or the number G of gray levels. Throughout the paper, N is assumed to be a power of two. *Gray value images* are NxN matrices with elements drawn from the set $\{0, 1, \ldots, G-1\}$ of linear ordered gray levels, for $G \geq 2$. *Binary images* are Boolean matrices of size NxN. For the analysis of

time complexities of programs we restrict ourselves to asymptotic statements. To this, we use the common notations

$$g = O(f) \text{ iff } (\exists c > 0)(\overset{\infty}{\forall} n) \quad g(n) \leq c \cdot f(n),$$

$$g = \Omega(f) \text{ iff } (\exists c > 0)(\overset{\infty}{\forall} n) \quad g(n) \geq c \cdot f(n),$$

for functions g, f which are defined on the natural numbers, and which have values in the set of non-negative reals. Furthermore, for a real number x, let $\lfloor x \rfloor$ be the greatest integer equal to or less than x, and $\lceil x \rceil$ denotes the least integer equal to or greater than x. For a natural number $n \geq 1$, log n denotes $\lceil \log_2 n \rceil$. The time complexity of a certain computational problem is equal to $\Theta_{RAM}(f)$ or $\Theta_{PMS}(f)$ iff optimal programs for the solution of this problem require $\Omega(f)$ and $O(f)$ time where the RAM or the PMS are used as models for computation, respectively. Furthermore, for a given computational problem the time complexities $\Theta_{RAM}(f_1)$ and $\Theta_{PMS}(f_2)$ imply a speed-up of $\Theta(f_1/f_2)$ from RAM to PMS.

An essential intention of report /5/ was the analysis of possible speed-ups from the RAM to different kinds of SIMD parallel processing systems where the PMS represents one of the discussed parallel models. For these comparisons between different hardware conceptions the derivation of meaningfull lower bounds is the crucial point. The following Data Transfer Lemma was used in /5/ for a quite general approach to obtain lower bounds for parallel computers in the matrix processing area.

2. Data Transfer Lemma

In the present paper, a *d a t a   u n i t* is a real number. Any *r e g i s t e r* may store one data unit at a time. For a certain model of computation, by the execution of a sequence of basic instructions some well-defined *d a t a   d e p e n d e n c e s* are marked out: for each register R, after the execution of a certain sequence of basic instructions a collection of registers is fixed such that the contents of these registers at the beginning of this execution affect the reached contents of register R at the end of this execution in an direct or indirect way, by the syntax of the given sequence of basic instructions only.

<u>Example 1:</u> Suppose the contents of N registers R_1, R_2,..., R_N are to be added. A certain parallel processing system, say PAR_ADD, is assumed to consist of N/2 processing elements, and to possess the ability of performing just log N different basic instructions for the parallel addition of register contents. For PAR_ADD, the meaning of these log N basic instructions may be as follows:

<u>for</u> $j \leftarrow 1$ <u>step</u> 2^i <u>until</u> N-1 <u>do in parallel</u>
$R_j \leftarrow R_j + R_{j+2^{i-1}}$ <u>od</u>,

for i = 1, 2,..., log N. This system PAR_ADD models the well-known parallel method of addition by an addition tree, for N = 8 the characteristic data flow is outlined in Fig. 1.

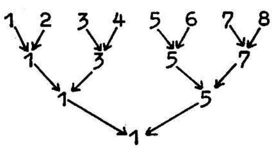


Fig. 1. Addition tree.

Obviously, after the execution of log N basic instructions in the order i = 1, 2,..., log N, in register R_1 the desired sum is computed.

Now, let $sub_t(n)$ be the set of indices of all those registers whose contents at the beginning of computation affect the contents of register R_n after the first t basic operations, direct or indirect, for t = 0, 1, 2,..., log N and n = 1, 2,..., N. Obviously, $sub_o(n) = \{n\}$ for n = 1, 2,..., N. If within step t an operation $R_n \leftarrow R_n + R_{n+2^{t-1}}$ is executed then

$$sub_t(n) = sub_{t-1}(n) \vee sub_{t-1}(n + 2^{t-1})$$

holds, and $sub_t(n) = sub_{t-1}(n)$ otherwise, for t = 1, 2,..., log N and n = 1, 2,..., N. After performing all the log N basic instructions we obtain

$sub_{\log N}(1) = \{1, 2, 3, \ldots, N\}$, $sub_{\log N}(2) = \{2\}$, $sub_{\log N}(3) = \{3, 4\}$, $sub_{\log N}(4) = \{4\}$ and so on. □

Example 2: Let us consider the following theoretical square array processor: for all integers i, j, module (i, j) possesses one register and in one step, each module transforms the contents of its register in dependences upon the previous contents in its own register, and in the registers of its four neighbor modules. This model corresponds to the structure of ILLIAC IV, cp. /7/, that is indicated in Fig. 2.
For this model, the sets $sub_t(i,j)$ are defined in an analogous way as the sets $sub_t(n)$ in Example 1, for $t = 0, 1, 2, \ldots$ and integers i, j. Then, for any module (i,j) after t steps of simultaneous parallel computation the result will be that $card\,(sub_t(i,j)) = 2t^2 + 2t + 1$.

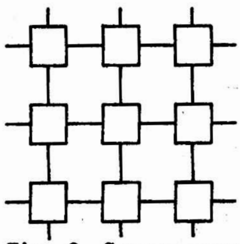


Fig. 2. Square array.

For a square array processor with both horizontal / vertical and diagonal connections of the modules, we obtain $4t^2 + 4t + 1$, for $t = 0, 1, 2, \ldots$, where such a connection scheme corresponds to the structure of CLIP 3, cp. /7/.

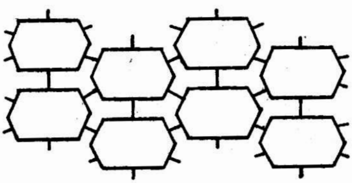


Fig. 3. Hexagonal array.

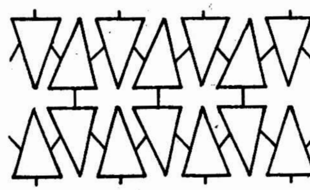


Fig. 4. Triangular array.

For a hexagonal or triangular array of modules, the data transfer is characterized by $3t^2 + 3t + 1$ or $\frac{3}{2}t^2 + \frac{3}{2}t + 1$, respectively, for $t = 0, 1, 2, \ldots$, where these connection schemes may be used for the design of VLSI algorithms, for example. □

For a model SYS of computation and $t \geq 0$, let $\chi_{SYS}(t)$ be the maximal number of registers of SYS whose contents at the beginning of a computation affect the contents of a certain register after t steps of computation, direct or indirect by test operations, indirect addressing, or others.
For example, in the case of Example 1 we can deduce
$\chi_{PAR\_ADD}(t) = 2^t$, for $t = 0, 1, 2, \ldots, \log N$, because
$\chi_{PAR\_ADD}(t) = \max \{card(sub_t(n)): n = 1, 2, \ldots, N\}$ for these t-values. From Example 2 we know that

$\chi_{ILLIAC\ IV}(t) = 2t^2 + 2t + 1$, for $t \leq n_0$, and $\chi_{CLIP\ 3}(t) = 4t^2 + 4t + 1$, for $t \leq n_1$, where n_0, n_1 are specific parameters given by the real size of these systems.
The function χ_{SYS} is called the *data transfer restriction* of system SYS, and this function offers a general approach to analyse the power of SYS for the solution of certain computational problems. In /2/ a similar function was proposed for the characterization of data transfer between processors of multiprocessor systems. According to /2/, we have for χ_{SYS} and the important computational problem of multiplication of NxN-matrices the following result.

Lemma 1: The multiplication of two NxN matrices (over the field of the real numbers) to produce the NxN product matrix requires at least t basic operations of SYS, where $\chi_{SYS}(2t) \geq N^2$.

For example, by the given result for $\chi_{ILLIAC\ IV}$ we know that the multiplication of NxN matrices on ILLIAC IV requires at least $\frac{1}{2}(N^2/2 - \frac{1}{4})^{1/2} - \frac{1}{4}$ basic operations, cp. /2/. For the application of χ_{SYS} to the consideration of lower bounds for further problems than matrix multiplication, we define:
For $n, m \geq 1$ let f be a function defined on n-tuples of real numbers, and with m-tuples of real numbers as values. Then, for $d \geq 1$, f is *d-dependent* iff
(a) there exist an $i \in \{1, 2, \ldots, m\}$ and d pairwise different integers $i_1, i_2, \ldots, i_d$ drawn from $\{1, 2, \ldots, n\}$ such that the

value of the i-th component of a certain vector $f(x_1, x_2, \dots, x_n)$ has to be computed under consideration of the values $x_{i_1}, x_{i_2}, \dots, x_{i_d}$, or

(b) there exist an $i \in \{1, 2, \dots, n\}$ and d pairwise different integers $i_1, i_2, \dots, i_d$ drawn from the set $\{1, 2, \dots, m\}$ such that the values in a certain vector $f(x_1, x_2, \dots, x_n)$ in positions $i_1, i_2, \dots, i_d$ are influenced by the value x_i.

Example 3: Let us consider the function

$$f(x_1, x_2, x_3, x_4, x_5) = \begin{cases} (x_1 + x_2, x_3), & \text{if } x_5 < 0, \\ (x_1, x_2), & \text{if } x_5 = 0, \\ (x_4, x_1 + x_3), & \text{if } x_5 > 0. \end{cases}$$

According to (a) in our definition, this function f is 1-, 2-, and 3-dependent but not 4- or 5-dependent. According to (b) in our definition, this function is 1- and 2-dependent but not 3-, 4-, or 5-dependent. Altogether, f is a 3-dependent function. The notion of d-dependence for vectors can be expanded to matrices in a natural way. Let INIT(x) be a NxN matrix with all elements equal to x, for arbitrary real x. For a NxN matrix

$A = (a_{ij})_{i,j\,=\,0,1,\dots,N-1}$ let
$\mathrm{COUNT}_x(A) = \mathrm{card}\,\{(i,j) : a_{ij} = x \,\&\, 0 \le i,\ j \le N-1\}$,

for an arbitrary real x. Then, INIT is a N^2-dependent function according to (b), and COUNT_x is a N^2-dependent function according to (a), for any real x. □

From the given definition of χ_{SYS} and of that of d-dependence, immediately follows

Lemma 2 (Data Transfer Lemma): The computation of a d-dependent function requires in the worst case at least t basic operations of SYS, where $\chi_{SYS}(t) \ge d$.

For example, the multiplication of two NxN matrices represents a 2N-dependent function what means that Lemma 1 is to be prefered rather than Lemma 2 for lower bounds, if χ_{SYS} is a polynom in t and N a sufficient large integer. Further examples: the inversion of an NxN matrix represents a N^2-dependent function, the solution of a system of N linear equations in N unknowns represents a $(N^2 + N)$-dependent function, and the Fourier transform of NxN matrices (over the field of complex numbers) represents a $2N^2$-dependent function. For such matrix processing tasks, the Data Transfer Lemma yields non-trivial lower bounds within the parallel processing area.
It must be mentioned that the data transfer restrictions $\chi_{CLIP\ 3}$ and $\chi_{ILLIAC\ IV}$ as given in this Section are approximations to the real functions only because the basic instruction sets of these systems were simplified in Example 2. In fact, the precise derivation of data transfer restrictions from complex computer systems is quite a time-consuming attempt. As an illustration of the general way we shall compute the data transfer restriction for a simple model of computation, the RAM as defined in /1/, in the next Section. On the other side, for the principal understanding of new hardware systems it seems to be sufficient to have some good approximations to the data transfer restriction as demonstrated for ILLIAC IV and CLIP 3.

3. Random Access Machine

The RAM as used in this paper matches the model defined in /1/ with the only modification that real number arithmetic replaces integer arithmetic, and that input and output operations have no influence to the discussions on time complexity of programs. The RAM disposes of an infinite sequence of registers R_0, R_1, R_2, R_3,... where R_0 denotes the accumulator.

Each basic RAM instruction consists of two parts - an operation code and an address. The basic RAM instructions are summarized in Tab. 1 where x denotes a real number, and i, b denote natural numbers. The address *i denotes indirect addressing. Fore a more detailed description of the RAM the reader is refered to /1/.

	operation code	address
(1)	LOAD, ADD, SUB, MULT, DIV, WRITE	= x, i, *i
(2)	STORE, READ	i, *i
(3)	JUMP, JGTZ, JZERO	b
(4)	HALT	-

Tab. 1. Basic instructions of the RAM.

Lemma 3: $\chi_{RAM}(t) = 2t + 1$, for $t \geq 0$.

Proof: Let $sub_t(n)$ be the set of indices of all those registers whose contents at the beginning of a certain computation affect the contents of register R_n after t basic RAM instructions, direct or indirect, for $n \geq 0$ and $t \geq 0$. We have $sub_o(n) = \{n\}$, for $n \geq 0$. In Tab. 2 for basic instructions of types (1) and (2) those cases are stated where, after performing a (t+1)-st basic operation, a set $sub_{t+1}(n)$ may differ from $sub_t(n)$ for a certain $n \geq 0$. In this table, c(i) denotes the contents of register R_i at the beginning of step t+1. The operations SUB, MULT, DIV lead to analogous results as ADD. If in certain steps t_o basic instructions JGTZ b or JZERO b have to be carried out then the set $sub_{t_o}(0)$ has to be joined to all sets $sub_t(n)$, for $t > t_o$ and if in step t a certain basic operation produces a value in register R_n, for $n \geq 0$. In step t_o itself all sub_{t_o}-sets remain unchanged.

instruction in step t+1	change opposite to sub_t-sets
LOAD = x	$sub_{t+1}(0) = \emptyset$
LOAD i	$sub_{t+1}(0) = sub_t(i)$
LOAD *i	$sub_{t+1}(0) = sub_t(i) \cup sub_t(c(i))$
STORE i	$sub_{t+1}(i) = sub_t(0)$
STORE *i	$sub_{t+1}(c(i)) = sub_t(0) \cup sub_t(i)$
ADD i	$sub_{t+1}(0) = sub_t(0) \cup sub_t(i)$
ADD *i	$sub_{t+1}(0) = sub_t(0) \cup sub_t(i) \cup sub_t(c(i))$
READ i	$sub_{t+1}(i) = \emptyset$
READ *i	$sub_{t+1}(c(i)) = sub_t(i)$

Tab. 2. Changes of sub_{t+1}-sets opposite to sub_t-sets.

Assume that the register contents c(1), c(2), ..., c(t) are pairwise different natural numbers greater than t. By the sequence

ADD *1, ADD *2, ..., ADD *t

of basic instructions we get the sets
$sub_t(0) = \{0, 1, 2, \ldots, t, c(1), c(2), \ldots, c(t)\}$, and
$sub_t(n) = sub_o(n) = \{n\}$ for $n \geq 1$. Thus, $\varkappa_{RAM}(t) = 2t + 1$ for $t \geq 0$.
On the other side, the maximal number of data dependences may be realized in register R_o obviously. By JGTZ b or JZERO b in R_o no additional data dependences are reached. In conformity with Tab. 2 the maximal growth in dependences may be attained by ADD *i (or SUB *i, MULT *i, DIV *i) instructions. □

Theorem 1: For a d-dependent function f a computation on the RAM requires at least $\lfloor d/2 \rfloor$ basic operations in the worst case.

This result follows immediately by Lemma 2 and Lemma 3. For example, by Theorem 1 we know that the two-dimensional Fourier transform requires at least N^2 basic operations which have to be performed by the RAM.

4. Parallel Matrix Processing System

The PMS represents an enlargement of the RAM by a two-dimensional processor array of size NxN and by countably many two-dimensional memory arrays MR_0, MR_1, MR_2, ... which we call m a t r i x   r e g i s t e r s , each of them consists of NxN registers. The matrix register MR_0 is called the accumulator array where all parallel processing takes place. In Tab. 3 all basic PMS instructions are summarized which are available in the PMS in addition to the RAM instructions. In this table, i and j denote natural numbers and the address (i,j) denotes the operand mc(0,i,j), mc(n,i,j) is defined to be the contents of register $MR_n(i,j)$ for $n \geq 0$ and $1 \leq i,j \leq N$. Addresses i, *i in Tab. 3, (5) refer to matrix registers MR_i, $MR_{c(i)}$, respectively. The instructions of type (6) guarantee the communication between the RAM part and the parallel processing part of the PMS. LOAD (i,j) means c(0)$\leftarrow$mc(0,i,j) and STORE (i,j) means mc(0,i,j)$\leftarrow$c(0).

	operation code	address
(5)	MLOAD, MADD, MSUB, MMULT, MDIV MWRITE, MSTORE, MREAD	i, *i
(6)	LOAD, STORE	(i,j)
(7)	MSHR, MSHL, MSHD, MSHU	i
(8)	MABS, MSIGN	-

Tab. 3. Additional basic instructions of the PMS.

By instructions of types (5), (7), (8), in the accumulator array MR_0 in all N^2 positions the designated operation will be performed (SIMD mode):
MLOAD i means mc(0,j,k)$\leftarrow$mc(i,j,k),
MADD i means mc(0,j,k)$\leftarrow$mc(0,j,k) + mc(i,j,k),
MSUB i means mc(0,j,k)$\leftarrow$mc(0,j,k) - mc(i,j,k),
MMULT i means mc(0,j,k)$\leftarrow$mc(0,j,k) times mc(i,j,k), i. e., the cross product of two NxN matrices,
MDIV i means mc(0,j,k)$\leftarrow$mc(0,j,k)/mc(i,j,k), MWRITE i means the output of the MR_i contents, MSTORE i means mc(i,j,k)$\leftarrow$mc(0,j,k), and MREAD i means an input into the matrix register MR_i, for all j,k = 1, 2,..., N in parallel.

Indirect addresses *i denote the operands $MR_{c(i)}$ and their meaning is analogous to direct addresses.
MABS means $mc(0,j,k) \leftarrow |mc(0,j,k)|$ and
MSIGN means $mc(0,j,k) \leftarrow sign(mc(0,j,k))$, where

$$sign(x) = \begin{cases} 0, & \text{if } x \le 0 \\ 1, & \text{if } x > 0, \end{cases}$$

for all $j,k = 1, 2, \ldots, N$ in parallel.
Array processors as the PMS are often equiped with a certain permutation network in practice, cp. /12/. Within the PMS, the parallel data transfer happens by power-of-two shifts in MR_0 only: the instructions MSHR i, MSHL i, MSHD i, or MSHU i mean that the contents of MR_0 has to be shifted by c(i) rows or columns to the right, to the left, down, or up, respectively, if c(i) denotes a power of two less or equal to N (otherwise these shift operations are not defined), where the rows or columns shifted out of array MR_0 will be discarded, and the rows or columns shifted into array MR_0 are assumed to be identical zero.

Example 4: To give an impression about the available parallelism within the PMS we consider the PMS realization of the Roberts gradient on gray value images $A = (a_{ij})_{i,j} = 0,1,\ldots,N-1$. This transform is defined by the replacement rule

$$a_{ij} \leftarrow \max\{|a_{ij} - a_{i+1,j+1}|, |a_{i+1,j} - a_{i,j+1}|\},$$

for $i,j = 0, 1, \ldots, N-1$ in parallel, and represents a well-known edge operator in digital image processing.
For two NxN matrices A and B, let $\max(A,B) = (\max\{a_{ij},b_{ij}\})_{i,j}$. Then we have

$$\max(A,B) = A \times sign(A-B) + B \times sign(B-A) + A - A \times sign|A-B|.$$

Thus, the matrix max(A,B) can be computed on PMS within constant time. Finally, it holds

$$ROBGRA(A) = \max(|A-shl\ 1(shu\ 1(A))|, |shu\ 1(A)-shl\ 1(A)|).$$

That means, the Roberts gradient has time complexity $\Theta_{RAM}(N^2)$ as well as $\Theta_{PMS}(1)$. □

Example 5: For a demonstration of the available parallel data transfer by power-of-two shifts we return to function INIT defined in Example 3. By the following program,

```
procedure INIT(A,x):
begin A ← A - A; a_{N-1,N-1} ← x; i ← 1;
    while i ≠ N do A ← A + shu i(A); A ← A + shl i(A);
                   i ← 2i od;
    return A
end,
```

the function INIT can be computed on PMS in O(log N) time. □

Lemma 4:

$$\chi_{PMS}(t) = \begin{cases} 2t + 1, \text{ for } t = 0, 1, \text{ or } 2, \\ 4\cdot 2^{m-1}+2m+1, \text{ for } t = 3m \text{ and } 1 \le m \le 1 + 2\cdot\log N, \\ 5\cdot 2^{m-1}+2m+2, \text{ for } t = 3m+1 \text{ and } 1 \le m \le 1 + 2\cdot\log N, \\ 6\cdot 2^{m-1}+2m+3, \text{ for } t = 3m+2 \text{ and } 1 \le m \le 1 + 2\cdot\log N, \\ (t-6\cdot\log N + 1)N^2+t-2\cdot\log N, \text{ for } t \ge 6 + 6\cdot\log N. \end{cases}$$

The proof of Lemma 4 may be accomplished analogously to that of Lemma 3. Of course, the parallel structure of the PMS causes additional difficulties contrary to the straightforward data transfer analysis in the RAM case. By combination of Lemma 2 and Lemma 4 we have

Theorem 2: For a d-dependent function f a computation on the PMS requires at least

$3\cdot\log d - 5$, if $8 \le d \le 8N^2$, and

$\lfloor d/N^2 \rfloor$, if $(5 + 6\cdot\log N)N^2 < d \le (6\cdot\log N - 1)N^4$

basic operations in the worst case.

In practice an economic way in computer design consists in simulation of an aspired system on the available one. In this sense, the simulation of PMS on a RAM is both of practical and theoretical interest. Any input-output relation that may be performed on PMS in time f(n) can be realized on RAM in time $k\cdot N^2\cdot f(n)$, for a certain universal $k > 0$. Thus, the attainable speed-up from RAM to PMS is upward restricted by $O(N^2)$ in each case. As in Example 4 shown, there exist computational problems with this optimal speed-up from RAM to PMS.

5. Computational Problems of Digital Image Processing

In this Section, to some fields of digital image processing computational problems are to be mentioned which were discussed in /5/ relative to different computer realizations. In Section 6 for these computational problems lower and upper bounds will be summarized. With the proposed selection of computational problems in digital image processing we attach the hope that some representative examples of low level image processing (i. e. receptive processing) are comprehended.

(i) For the uniform treatment of the whole of the NxN gray value image $A = (a_{ij})_{i,j=0,1,\dots,N-1}$ by certain global transform methods, we consider the histogram computation, $hist(A) = (h_0, h_1, h_2, \dots, h_{G-1})$ with $h_k = card\{(i,j) : a_{ij} = k \;\&\; 0 \le i,j \le N-1\}$, for $k = 0, 1, 2, \dots, G-1$, the (common) matrix multiplication, the Fourier transform

$$F(A)(u,v) = \frac{1}{N}\sum_{i=0}^{N-1}\sum_{j=0}^{N-1} a_{ij}\cdot \exp(-2\pi\sqrt{-1}\,(iu + jv)/N),$$

for $u,v = 0, 1, 2, \dots, N-1$, the Walsh transform

$$W(A)(u,v) = \frac{1}{N}\sum_{i=0}^{N-1}\sum_{j=0}^{N-1} a_{ij}\cdot \prod_{k=0}^{n-1}(-1)^{b_k(i)b_{n-1-k}(u)+b_k(j)b_{n-1-k}(v)},$$

for $u,v = 0, 1, 2, \dots, N-1$, $n = \log N$, and $b_k(i)$ denotes the k-th bit in the binary representation of i, and, finally, pattern matching: if a given pattern of size MxM appears in the image A in any position, $M \le N$, then this match is signaled by a special output.

(ii) For the uniform treatment of the whole of the NxN gray value image $A = (a_{ij})_{i,j=0,1,\dots,N-1}$ by certain local transform methods, we consider the median filter (each pixel gets the median of its MxM neighborhoud as its new value), and the Roberts gradient as defined in Example 4.

(iii) As examples for geometric transforms of a NxN gray value image $A = (a_{ij})_{i,j=0,1,\dots,N-1}$, we shall consider shifts in row or column direction, where the shift distance may be an arbitrary integer drawn from the set $\{0,1,2,3,\dots,N\}$, cyclic

shifts in row or column direction with arbitrary shift distances, the matrix transposition, the vertical and the horizontal invertation, and rotations by multiples of $\pi/2$.
(iv) As problems in computational geometry, we consider the denotation of the convex hull for ordered sequences of contour points of length n, testing digital straight line segments for digital arcs of length n, and testing convexity for digital objects with contour sequences of length n; for the definition of these notions reference is made to /4/, /10/, /11/.
(v) For simple recognition tasks in the case of NxN binary images we shall consider decisions whether a given image exactly contains one solid rectangle, one solid square, or one solid isosceles triangle.
(vi) As simple examples of feature extraction for a NxN binary image A, we consider the functions number (A) = card $\{(i,j) : a_{ij} = 1 \ \& \ 0 \le i,j \le N-1\}$, zero(A) = 1 iff A is identical zero, zero(A) = 0 otherwise, one(A) = 1 iff in A exactly one pixel has value 1, one(A) = 0 otherwise, comp(A) = number of different 8-components in A, and, finally, the computation of the Euler numbers $E_1(A)$ = number of 8-components in A minus the number of 4-holes in A, as well as $E_2(A)$ = number of 4-components in A minus the number of 8-holes in A, cp. /10/ for different approaches to compute such functions.

6. Time Complexities

First of all we state the time complexities of those computational problems mentioned in Section 5 for which both Θ_{RAM} and Θ_{PMS}, and thus the speed-up from RAM to PMS, are known. Here as in the following it should be clear that the results given in Theorems 1 and 2 are useful tools for the statement of lower bounds for quite a lot of the mentioned problems.

	Θ_{RAM}	Θ_{PMS}
(1) pattern matching	N^2	log N
(2) Roberts gradient	N^2	1
(3) rectangle	N^2	log N
(4) square	N^2	log N
(5) isosceles triangle	N^2	log N
(6) number	N^2	log N
(7) zero	N^2	log N
(8) one	N^2	log N
(9) E_1 and E_2	N^2	log N

Now, all those problems will be listed for which Θ_{RAM} is known but not Θ_{PMS}.

	Θ_{RAM}	Ω_{PMS}	O_{PMS}
(10) histogram	N^2	log N	G log N
(11) shifts	N^2	1	log N
(12) cyclic shifts	N^2	1	log N
(13) matrix transposition	N^2	1	log N
(14) vertical/horizontal invertation	N^2	1	log N
(15) rotations by multiples of $\pi/2$	N^2	1	log N

Adequate we add those problems with known Θ_{PMS} but unknown Θ_{RAM}.

	Ω_{RAM}	O_{RAM}	Θ_{PMS}
(16) two-dimensional Fourier transform	N^2	$N^2 \log N$	log N
(17) two-dimensional Walsh transform	N^2	$N^2 \log N$	log N

Finally, the most indeterminate cases among the mentioned problems are:

	Ω_{RAM}	O_{RAM}	Ω_{PMS}	O_{PMS}
(18) matrix multiplication	N^2	$N^{2.609}$	log N	N log N
(19) median filtering	N^2	$M N^2$	log N	$G(\log M)^2 + \log N$

For the computational problems given in point (iv) of Section 5, in /4/ optimal algorithms are stated with $\Theta_{RAM}(n)$ time in each case.

Thus, the determination of the convex hull for an ordered contour point sequence represents, as a matter of principle, a more simple problem than the computation of the convex hull of n arbitrary placed points in the plane (this problem possesses the time complexity $\Theta_{RAM}(n \cdot \log n)$ as shown in /9/). In /6/ it is proved that certain quantitative recordings of convexity of digital objects may be done on PMS real well (time $\Theta_{PMS}(\log N)$ for digital objects given by NxN binary images).
To the given summary for problems (1) ... (19), four notes have to be added:

(a) For the results given for pattern matching in (1), we have assumed that the pattern size M is viewed as a constant.

(b) To the two-dimensional Fourier transform, in /3/ a parallel algorithm for the PMS is explained such that all arithmetical operations of this transform are performed within O(log N) basic PMS operations, however, for the obtained matrix all elements have yet to change their positions according to the bitwise inversion of their indices. To this transform a permutation network would be well designed but the PMS as defined in Section 4 offers no adapted data transfer operations. In this sense, the results $O_{PMS}(\log N)$ as used in (16) and (17) are restricted to the performance of arithmetical operations only.

(c) In /8/ it is shown that under certain restrictions, $O_{RAM}(n \cdot \log n)$ is required for the discrete one-dimensional Fourier transform.

(d) It has to be expected that the upper bound given for the multiplication of NxN matrices on a RAM does not reflect the state of the art according to the fast growing number of recent papers to this subject.

7. Conclusions

By the mentioned results in Section 6 it should be clear that several computational problems as in digital image processing can be essentially speed up by using the PMS opposite to the RAM as model of computation, and especially those problems, where the whole of the NxN array is submitted to an uniform treatment. It is remarkable that this includes transforms as the

median filtering, i. e., transforms with local decisions. Furthermore it is noticed that not only for simple local operations as the Roberts gradient but also for the arithmetical operations of two-dimensional Fourier or Walsh transforms the speed-up from RAM to PMS is equal to or closed to the maximal possible speed-up $O(N^2)$.
Altogether, this paper may be understood as a contribution to the current research on interactions between

1. the structure of favourable parallel hardware with special attention to the possibilities of data transfer in these systems,
2. the possible parallizations of solution processes for computational problems in different fields of science and engineering, and
3. the practicability of the proposed hardware-software complexes.

For the design of optimal parallel algorithms, in hardware or for a programmable parallel computer, these three points must be taken into consideration as the general framework. For example, matrix multiplication may be done by O(log N) parallel operations but the corresponding hardware demands are highly unrealistic at present, cp. /5/. Thus, the design of parallel algorithms is quite different from the common design of sequential algorithms for a von Neumann computer, where algorithm design and hardware design are nearly independent processes.
Furthermore, the investigations in /5/ have teached us that for the design of efficient parallel algorithms a programmable parallel computer appears n o t as the best solution neither for an universal image processor nor for specialized image processors. In our opinion, a modular system would be the economic way where besides a central sequential processor to the data bus some parallel algorithms, realized in hardware, may be connected. Generally speaking, the aim should be that efficient solutions for parallel algorithms are realized in hardware to obtain the optimal enlargement of the central processor.

Acknowledgement

The author is specially obliged to Albrecht Hübler and Egbert Creutzburg, both from Jena University, for useful discussions and critical hints.

References

/1/ Aho, A. V., Hopcroft, J. E., and Ullman, J. D.: The Design and Analysis of Computer Algorithms. Reading, Mass., 1974

/2/ Gentleman, W. M.: Some Complexity Results for Matrix Computations on Parallel Processors. J. Assoc. Comput. Mach. 25, 112 - 115 (1978)

/3/ Hübler, A., and Klette, R.: On Parallel Realization of the Fast Fourier Transform of NxN Matrices. Elektron. Informationsverarb. Kybernet. 15, 599 - 609 (1979)

/4/ Hübler, A., Klette, R., and Voß, K.: Determination of the Convex Hull of a Finite Set of Planar Points within Linear Time. Elektron. Informationsverarb. Kybernet. 17 (1981)

/5/ Klette, R.: Zeitkompliziertheiten von Berechnungsproblemen der digitalen Bildverarbeitung. Entwurf Dissertation B, Sektion Mathematik der Friedrich-Schiller-Universität Jena 1980

/6/ Klette, R., and Krishnamurthy, E. V.: Algorithms for Testing Convexity of Digital Polygons. TR-841, Computer Science Center, University of Maryland 1979

/7/ Kotov, V. L., and Mikloško, J. (Eds.): Algorithms, Software and Hardware of Parallel Computers. Bratislava (to appear)

/8/ Morgenstern, J.: Note on a Lower Bound of the Linear Complexity of the Fast Fourier Transform. J. Assoc. Comput. Mach. 20, 305 - 306 (1973)

/9/ Preparata, F. B., and Hong, S. J.: Convex Hulls of Finite Sets of Points in Two and Three Dimensions. Comm. ACM 20, 87 - 93 (1977)

/10/ Rosenfeld, A., and Kak, A.: Digital Picture Processing. New York 1976

/11/ Simon, H., Kunze, K. D., Voß, K., und Herrmann, W. R. (Eds.): Automatische Bildverarbeitung in Medizin und Biologie. Dresden 1975

/12/ Sykora, O.: The Generalization of the Perfect Shuffle Principle and its Application. Schr. WBZ MKR Dresden 39, 92 - 100 (1979)

received: March 30, 1981

Author's address:

Dr. R. Klette
Friedrich-Schiller-Universität
Sektion Mathematik
Universitätshochhaus 17. OG
DDR-6900 Jena

Hinweise für Autoren

Manuskripte (in deutscher, ggf. auch in russischer oder englischer Sprache) bitten wir, an die Schriftleitung zu schicken. Die gesamte Arbeit ist linksbündig zu schreiben. Eine Ausnahme hiervon bilden hervorzuhebende Formeln und das Literaturverzeichnis. Der Kopf der Arbeit soll folgende Form haben: Rostock. Math. Kolloq./ Leerzeile/ Vorname Name/ Leerzeile/ Titel der Arbeit/ 1 Zeilenumschaltung/ Unterstreichung/ Leerzeile. Der Text der Arbeit ist eineinhalbzeilig (= 3 Zeilenumschaltungen) zu schreiben mit maximal 63 Anschlägen je Zeile und maximal 37 Zeilen je Seite. Zwischenüberschriften sind wie folgt einzuordnen: 6 Zeilenumschaltungen/ Zwischenüberschrift/ Unterstreichung (ohne Zeilenumschaltung)/ 5 Zeilenumschaltungen. Hervorhebungen sind durch Unterstreichen und Sperren möglich. Ankündigungen wie Satz, Definition, Bemerkung, Beweis u. ä. sind zu unterstreichen und mit einem Doppelpunkt abzuschließen. Vor und nach Sätzen, Definitionen u. ä. ist ein Zeilenabstand von 5 Umschaltungen zu lassen. Fußnoten sind möglichst zu vermeiden. Sollte doch davon Gebrauch gemacht werden, so sind sie durch eine hochgestellte Ziffer im Text zu kennzeichnen und innerhalb des oben angegebenen Satzspiegels unten auf der gleichen Seite anzugeben. Formeln und Bezeichnungen sollen möglichst mit der Schreibmaschine zu schreiben sein. Hervorzuhebende Formeln sind drei Leerzeichen einzurücken und mit 6 Umschaltungen zum übrigen Text zu schreiben. Formelzähler sollen am rechten Rand stehen. Der Platz für Abbildungen ist beim Schreiben auszusparen; die Abbildungen selbst sind in der dem ausgesparten Platz entsprechenden Größe gesondert nach TGL-Vorschrift auf Transparentpapier beizufügen. Der zugehörige Begleittext ist im Manuskript mitzuschreiben. Sein Abstand nach unten beträgt 5 Umschaltungen. Literaturzitate im Text sind durch laufende Nummern in Schrägstrichen (vgl. /8/, /9/ und /10/) zu kennzeichnen und am Schluß der Arbeit unter der Zwischenüberschrift Literatur zusammenzustellen.

Beispiele: (Zeitschriftenabkürzungen nach Math. Reviews)

/8/ Zariski, O., and Samuel, P.: Commutative Algebra. Priceton 1958

/9/ Steinitz, E.: Algebraische Theorie der Körper. J. Reine Angew. Math. 137, 167 - 309 (1920)

/10/ Gnedenko, B. W.: Über die Arbeiten von C. F. Gauß zur Wahrscheinlichkeitsrechnung. In: Reichard, H. (Ed.): C. F. Gauß, Gedenkband anläßlich des 100. Todestages. S. 193 - 204, Leipzig 1967

Die Angaben sollen in Originalsprache erfolgen; bei kyrillischen Buchstaben soll die bibliothekarische Transkription (Duden) verwendet werden.

Am Ende der Arbeit stehen folgende Angaben zum Autor und zur Arbeit: eingegangen: Datum/ Leerzeile/ Anschrift des Verfassers:/ Titel Initialen der Vornamen Name/ Institution/ Struktureinheit/ Straße Hausnummer/ Land Postleitzahl Ort.

Der Autor wird gebeten, eine Korrektur des Durchschlags vom Offsetmanuskript zu lesen und dabei die mathematischen Symbole einzutragen. Ferner sollte er 1 - 2 Klassifizierungsnummern (entsprechend der "1980 Mathematics Subject Classification" der Math. Reviews) zur inhaltlichen Einordnung seiner Arbeit angeben.