Tomus 2. Fasciculus 3.

ACTA CYBERNETICA

FORUM CENTRALE PUBLICATIONUM CYBERNETICARUM HUNGARICUM

Szeged, 1975

Curat: Universitas Szegediensis de Attila József nominata

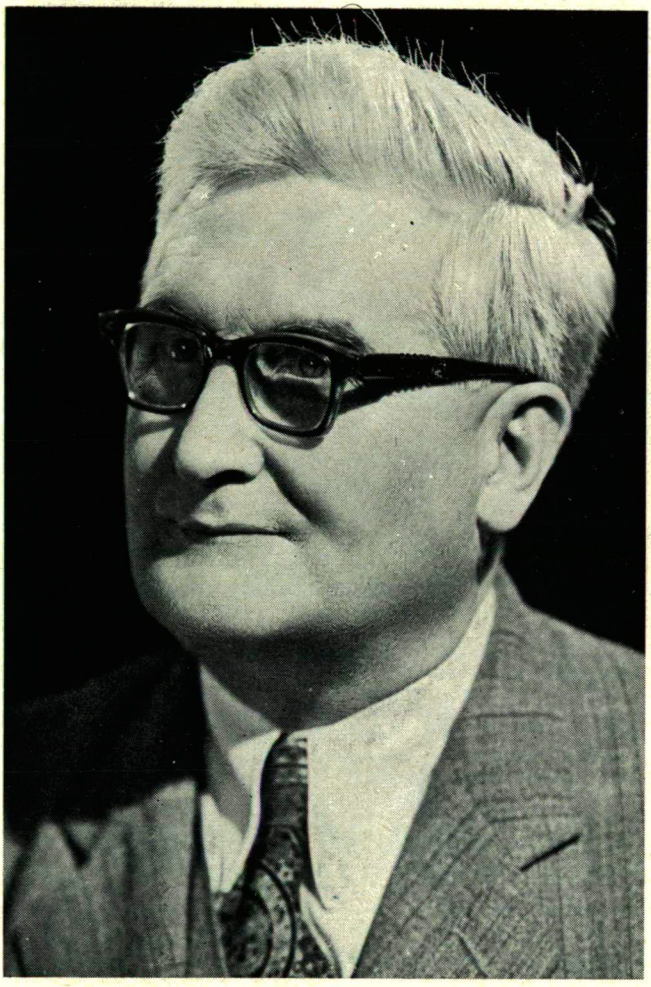
2. kötet 3. füzet

ACTA CYBERNETICA

A HAZAI KIBERNETIKAI KUTATÁSOK KÖZPONTI PUBLIKÁCIÓS FÓRUMA

Szeged, 1975. december

A Szegedi József Attila Tudományegyetem gondozásában



PROF. L. KALMÁR

The work of Academician László Kalmár in the field of computer science

(On the occasion of his 70th birthday)

By M. ARATÓ

Having completed my secondary school studies I was admitted to the Szeged University where I got acquainted with the inner rules of development and mysteries of mathematics. I am firstly indebted to this launching if I had achieved results and acquired knowledge in the field of mathematics. I learned from László KALMÁR mathematical analysis and it was under his auspices that I took my first steps towards scientific activity. I learned in Szeged the basic concepts which have led me to problems related to measure theory and mathematical statistics. 20 years ago we celebrated to 50th birthday of Laszló Kalmár, a young man at that time, and commemorated of his achievements in the most diverse mathematical disciplines. He established a scientific school and it is a pity that I could return to this workshop only now and then. I had the opportunity to get an insight into the mysteries of other large schools so that I have ground for comparison as regards the impact of Kalmár on science and scientific organization. Youthfulness, relentless search of novelty, these criteria of a genuine scientist are all preserved by him while he strives with incredible tenacity to promote the triumph of what is new.

During the last 20 years an important change was brought about in his life due to the appearance of computational techniques, of modern computers. We met again in this field, — that is in the domain of computer science — although Kalmár had started from more distant bases and deeper roots in approaching the problem from the side of mathematical logic. Also at this juncture could I be impressed by his assessment of other trends and by his esteem for other attitudes. The activity of László Kalmár for the last 20 years has resulted in an addition to the values of computer science. His first paper was published in 1958 about digitally operated machines, logical machines. Since then he has dealt in further 36 papers and studies with the most various research issues. Over and above the topics of linguistics, biology and medicine, he has been engaged in the everyday problems and teaching of cybernetics. His widespread work embraces every small detail of research and organization. Of the mathematicians of our country he is the one who follows recent results with the greatest attention and knows more than many of us of the situation and potentialities of computer science. There will be many to work further on the important problems of research trends delineated by Kalmár. Instead of their detailed enumeration and the general characterization of Kalmár's activity may I

remind the reader to his two papers by highlighting their impact on home and international research.

Let us consider Kalmár's paper, presented in Esztergom in 1968, on modern programming languages. At that time we here in Hungary got acquainted with recent programming languages, with the novelty of Algol 68, PL/1, Almo. Kalmár suggested, besides educational issues, the implementation of the partlanguages existing at that time. Out of his suggestions related to issues which had not been then put forward elsewhere or had hardly been investigated, may I point to the following ones:

1. Having discovered that the substantial time interval between the writing and running of a program endangered the possibility of full machine utilization, he suggested, in order to reduce this interval, the automatic error correction of programs similarly as in the introduction of error correcting codes as it had been developed in information theory. This keen insight of Kalmár has been justified by the way of later development and it is only to be regretted that in the wake of his suggestion no adequate research has been undertaken in Hungary.

2. He was already at that time well aware of the importance of non-numerical applications with respect to the works put on the computer. He drew the attention to the role played by COBOL. At that time not only the mathematicians but the majority of computer experts were unable to give the matter due consideration. Kalmár's guidelines are important also in this field.

3. He suggested to exploit in the programming languages the possibilities of getting closer to human language. He felt that the development of algorithmic programming languages got too much out of touch with the users. His suggestions conformed to the concepts developed worldwide since then in the field of man-machine interaction.

4. In the field of setting theoretical tasks he urged the establishment of closer contacts with mathematical logic. This was responsible for the substantial progress achieved since then not only in the field of the description of machine languages but also in that of data bases. There were likewise important results in the description of two stage grammars achieved in mathematical linguistics, considered also as an important task by László Kalmár.

As to his second paper to be reminded here László Kalmár did a pioneering work in 1962 by publishing an article in the Communications of the Math. Phys. Section of the Hungarian Academy of Sciences in the subject of interrelationship between information theory and logic. Working on issues of qualitative information he pointed out that not only in the field of communication engineering, but also in that of computers it is possible to achieve another kind of introduction of information which requires logical building up. In this area important moves were made in mathematical research also independently from Kalmár. The necessity of logical foundation was investigated also by Kolmogorov in his research activities and this tendency has led to a lot of very interesting results in the last decade. It suffices to refer here to the basic problems and results of the complexity.

In the various mathematical disciplines as well as in their applications László Kalmár discovers, moreover forecasts the appearance of the most essential forthcoming new trends. The appreciation of his mathematical activities and work is proven by the fact that László Kalmár was the first Hungarian mathematician

in the wake of whose work a circle of applied mathematicians has come about in this country. Either Jenő Egerváry, or Alfréd Rényi had not been able to achieve this to such an extent. The applications of neither the differential equations, nor of mathematical statistics could give such a reliable foundation for the education of mathematicians as is assured by the direction of programming and computing techniques suggested by Kalmár.

As a teacher and educator of computer science, he not only teaches his students the knacks of the science but supports them, too, in their work. He provides his students not only with problems for a long period, but also with useful advices. Let us do our best availing ourselves of these counsels.

From the realization of his ideas he hopefully will derive also hereinafter much pleasure not only in Szeged and Hungary, but also outside of our country. To this effect we wish him good health and much success.

Die Rekursivität der Programmierungsprache „Lisp 1.5“[1] in Spezialfällen der angeordneten freien holomorphen Mengen

Von R. PÉTER

1. In 1959 habe ich[2] den allgemeinen Begriff der angeordneten freien holomorphen Mengen (Mengen mit „zahlenartig aufbaubaren“ Elementen) eingeführt, und die Theorie der rekursiven Funktionen für solche abstrakte Mengen als Definitionsbereiche verallgemeinert; ferner als wichtigsten Spezialfall den Fall der „Wortemengen“ (Mengen der endlichen Ketten, „Worte“ genannt, aus Elementen je einer gegebenen Menge, „Alphabet“ genannt) über Alphabete beliebiger Mächtigkeiten ausgearbeitet. Auch von anderen Verfassern wurde besonders dieser Spezialfall des allgemeinen Begriffes vielseitig untersucht und angewandt. Als einen davon abweichenden Spezialfall habe ich[3] den Begriff der PAIRschen „freien Binoiden“ in dieser Hinsicht untersucht. Als ein anderer Spezialfall bietet sich die Menge der „symbolischen Ausdrücke“ in der Programmierungsprache „Lisp 1.5“, worum es sich in dieser Arbeit handelt.

2. Zuerst gebe ich die notwendigen Kenntnisse über die angeordneten freien holomorphen Mengen an.

Sei H eine beliebige nicht leere Menge, H_0 eine nicht leere Teilmenge von H (alle Elemente von H_0 werden die Rolle von 0 spielen) und F eine nicht leere Teilmenge der auf H definierten und nur Werte aus H annehmenden Funktionen beliebig vieler Variablen (alle Elemente von F werden die Rolle der Nachfolgerfunktion spielen). Ferner sei H_n für $n=1, 2, \ldots$ folgendermaßen angegeben: sind die Mengen $H_0, H_1, \ldots, H_n$ bereits definiert, dann sei H_{n+1} die Menge jener Elemente von H, welche als Werte der Funktionen aus F angenommen werden, falls für ihre Argumente Elemente aus $H_0 \cup H_1 \cup \ldots \cup H_n$ gesetzt werden, und dabei mindestens für ein Argument ein Element aus H_n. Wird nun H durch die Union sämtlicher Teilmengen $H_0, H_1, H_2, \ldots$ erschöpft, dann wird H eine *holomorphe Menge* genannt.

[1]Siehe: *LISP 1.5 Programmer's Manual*, The Computation Center and Research Laboratory of Electronics, Massachusetts Institute of Technology (1962). Zitiert als [1].

[2]Siehe: R. PÉTER: *Über die Varellgemeinerung der Theorie der rekursiven Funktionen für abstrakte Mengen geeigneter Stuktur als Deflnitionsbereiche*, Acta Math. Ac. Sci. Hung., Teil I in **12** (1961), S. 271-314 (mit Angabe in der ersten Fußnote der Geschichte dieses Themenkreises); Teil II in **13** (1962), S. 1-24 (mit einigen Berichtigungen zum Teil I; betrefs weiterer Berichtigungen berufe ich mich auf [3]). Zitiert als [2].

[3]Siehe: R. PÉTER: *Die* PAIR*schen freien Binoiden als Spezialfälle der angeordneten freien holomorphen Mengen*, Acta Math. Ac. Sci. Hung. **21** (1970), S. 297-313. Zitiert als [3].

Und zwar eine *freie holomorphe Menge,* falls die Mengen H_n paarweise disjunkt sind, und für jedes nicht zu H_0 gehörige Element von H eindeutig bestimmt ist, aus welcher zu F gehörigen Funktion und durch Einsetzen welcher Elemente für deren Argumente es entsteht. Im weiteren wird H immer eine freie holomorphe Menge bezeichnen, und auch die Bezeichnungen $F, H_0, H_1, H_2, \ldots$ werden beibehalten.

$h \in H_i$ wird auch so ausgedrückt, daß die Ordnung $o(h)$ von h gleich i ist.

Es ist auch die Einführung des Begriffes der *Vorgänger* je eines Elementes von H notwendig; diese kann aber in verschiedenen Anwendungen verschiedenartig angegeben werden; jedoch (mit der Bezeichnung „$\prec$" für echte Vorgänger) so, daß die folgenden Forderungen erfüllt seien:

(1) $x \leqq x$.

(2) Ist x der Form $f(y_1, \ldots, y_n)$, wo $f \in F$ ist, so gehören die „unmittelbaren Konstituenten" $y_1, \ldots, y_n$ von x zu den echten Vorgängern von x.

(3) Aus $x \prec y$ und $y \prec z$ folgt $x \prec z$.

(4) Ein echter Vorgänger von x ist immer kleinerer Ordnung als x.

Der ebenfalls notwendige Begriff der *unmittelbaren Vorgänger* der Elemente von X kann ebenfalls verschiedenartig angegeben werden, allein mit den Forderungen:

(5) Es gebe für jedes festgewählte $f \in F$ eine von den Argumenten unabhängige obere Schranke für die Anzahl der unmittelbaren Vorgänger von Elementen der Struktur

$$x = f(\ldots).$$

(6) Jeder echte Vorgänger eines x sei Vorgänger eines unmittelbaren Vorgängers von x.

Eine freie holomorphe Menge H mit einem den Forderungen (1)—(6) genügenden Vorgängerbegriff wird eine (partiell) *angeordnete freie holomorphe Menge* genannt. Von nun an sei durch H immer eine solche Menge bezeichnet.

Auf Grund einer beliebigen solchen Menge H als Definitionsbereich kann eine rekursive Theorie aufgebaut werden.

3. In der Programmierungsprache „Lisp 1.5" werden „Listen" (endliche geordnete Mengen) untersucht, die auf Grund gewisser „Atome" aufgebaut werden. Die Atome sind Elemente einer Wortemenge über ein endliches Alphabet A_0, das Buchstaben, Ziffern, und einige spezielle Charaktere enthält. Die Länge der verwendbaren Worte wird aber eingeschränkt; so gibt es nur endlich viele Atome. Die Glieder der Listen sind Atome oder Listen. Die beliebigvielgliedrigen Listen können auf Paare abgebaut werden: als Gegenstück des ersten Gliedes der Liste kann die Liste der übrigen Glieder gewählt werden; dieses letztere kann ebenfalls auf diese Weise als ein Paar aufgezeichnet werden, usw.; als Gegenstück des letzten Gliedes kann die durch „NIL" bezeichnete leere Liste aufgenommen werden, die auch als ein Atom betrachtet wird. So werden statt Listen sog. „S-Ausdrücke" („symbolische Ausdrücke") behandelt, wobei erstens die Atome als S-Ausdrücke gelten, dann aus je zwei beliebigen S-Ausdrücken s_1 und s_2 als ein Paar der folgenderweise bezeichnete S-Ausdruck s gebildet wird:

$$s = (s_1 \cdot s_2).$$

Gehört s zu einer Liste, so gehört s_1 zum ersten Glied („head") dieser Liste, und s_2 zu jener (eventuell leeren) Liste („tail"), die aus der ursrünglichen Liste durch Weglassen des ersten Gliedes entsteht.

s_1 bzw. s_2 als Funktionen von s werden durch $s_1 = \text{car}(s)$ bzw. $s_2 = \text{cdr}(s)$ bezeichnet, und s als Funktion von s_1 und s_2 durch $s = \text{cons}(s_1, s_2)$.

Man zeigt leicht,[4] daß diese Funktionen, und viele mittels diesen definierten weiteren Begriffe des „Lisp 1.5" in der Wortemenge sowohl über das Alphabet A_0 als auch über das aus den zugelassenen Atomen bestehenden Alphabet A (beidenfalls mit Hinzunahme der Klammern und der Trennzeichen) primitiv-rekursiv sind.

4. Doch die rekursive Theorie der S-Ausdrücke kann nicht nur in eine Wortemenge eingebettet behandelt werden, sondern sie bietet sehr natürlicher Weise ein Beispiel für einen anderen Fall der angeordneten freien holomorphen Mengen. Darin spielt jedes Atom (von nun an — NIL inbegriffen — nicht als Zeichenkette, sondern als ein einziges Zeichen betrachtet) die Rolle von 0, und $\text{cons}(x, y)$ die Rolle der „Nachfolgerfunktion"; ferner besitzt jedes nicht-Atom Element x zwei „unmittelbare Vorgänger":

$$\text{car}(x) \quad \text{und} \quad \text{cdr}(x).$$

Der Aufbau der Theorie der rekursiven Funktionen für eine derartige Menge H wird dadurch erleichtert, daß die (einzige) Nachfolgerfunktion hier zweistellig ist. Für Fälle, wobei es unter den Nachfolgerfunktionen auch eine mindestens zweistellige gibt, habe ich in der Bemerkung (I) auf S. 299 von [2] eine Zuordnung geeigneter Elemente zu je einer Elementenfolge angegeben, und diese auch in [3] benutzt. Hier bietet sich nun eine ähnliche Zuordnung eines S-Ausdrucks zu je einer S-Ausdruck-Folge, wie je einer Liste ein S-Ausdruck zugeordnet wurde; und eine derartige Zuordnung kann auch auf beliebige solche H-Mengen verallgemeinert werden, bei welchen die Menge F auch eine mindestens zweistellige Nachfolgerfunktion enthält.

5. Nun sei die H-Menge der S-Ausdrücke näher betrachtet.

Dabei ist H_0 die (endliche) Menge der Atome, und davon ausgehend entstehen für $n = 1, 2, \ldots$ die Elemente von H_n wie im allgemeinen Fall, durch Anwendungen der einzigen Nachfolgerfunktion $\text{cons}(x, y)$. Jedes nicht-Atom Element x von H hat die Form

$$(x_1 \cdot x_2),$$

wobei x_1 und x_2 eindeutig bestimmte Zeichenketten sind:

$$x_1 (= \text{car}(x))$$

ist jene Zeichenkette, die man aus der Zeichenkette x erhält, wenn man nach Weglassen der ersten Anfangsklammer nach rechts gehend die Zeichen bis zum ersten solchen Punkt kopiert, bis welchem die Anzahl der aufgetretenen Anfangs- und Endklammern übereinstimmt; dann ist

$$x_2 (= \text{cdr}(x))$$

[4] Siehe das Buch: R. PÉTER: *Rekursive Funktionen in der Komputer-Theorie* (im Erscheinen).

jene Zeichenkette, die aus dem übriggebliebenen Teil der Zeichenkette x durch Weglassen des ersten Punktes und der letzten Endklammer entsteht.

Als festgesetzte Elemente je einer der Mengen

$$H_0, H_1, H_2, \ldots$$

können

$$h_0, h_1, h_2, \ldots$$

gewählt werden, wobei

$$h_0 = \text{NIL}$$

und für jedes n

$$h_{n+1} = \text{cons}\,(h_n, h_n)$$

ist.

Jedes Atom besitzt als einzigen (unechten) Vorgänger sich selber. Als unmittelbare Vorgänger eines Elementes $x = \text{cons}\,(x_1, x_2)$ gelten seine unmittelbare Konstituenten x_1 und x_2, und als echte Vorgänger von x gelten die Vorgänger dieser Konstituenten. Offenbar werden für diese Begriffe die Forderungen (1)—(6) erfüllt.

Nach alldem ist unser H eine angeordnete freie holomorphe Menge.

6. Für dieses H, wobei die Menge der Atome

$$A = \{a_1, a_2, \ldots, a_t\}$$

sei, lautet das Schema der primitiven Rekursion:

$$\begin{cases} f(a, u_1, \ldots, u_n) = g_a(u_1, \ldots, u_n), \text{ falls } a \in H_0 (= A) \\ f(\text{cons}\,(x_1, x_2), u_1, \ldots, u_n) = g(x_1, x_2, u_1, \ldots, u_n, f(x_1, u_1, \cdots, u_n), f(x_2, u_1, \ldots, u_n)), \end{cases}$$

wobei

$$g_{a_1}, \cdots, g_{a_t}, g$$

bereits definierte Funktionen bezeichnen.

Als Ausgangsfunktionen gelten unbedingt die Atome, d. h. die Elemente von H_0 (als Konstanten), das einzige Element $(\text{cons}\,(x_1, x_2))$ von F und eventuell noch weitere hinzuzunehmende Funktionen. Die in H primitiv-rekursiven Funktionen sind jene, die aus den Ausgangsfunktionen ausgehend durch endlich viele Substitutionen und primitive Rekursionen entstehen. Dabei darf jede Funktion so betrachtet werden, als eine Funktion ihrer Argumente und beliebig (endlich) vieler „hinzugenommenen" Argumente, von denen sie nicht tatsächlich abhängt.

Eine Relation $R(x_1, \ldots, x_n)$ ist primitiv-rekursiv, falls ihre „charakteristische Funktion"

$$r(x_1, \ldots, x_n) = \begin{cases} h_0, \text{ falls } R(x_1, \ldots, x_n) \\ h_1 \text{ sonst} \end{cases}$$

primitiv-rekursiv ist.

Zum Beispiel sind car (x) und cdr (x), die nur dann eine Rolle spielen, wenn x kein Atom ist, primitiv-rekursiv, da sie durch die folgenden primitiven Rekursionen definiert werden können:

$$\begin{cases} \text{car}\,(a) = \text{NIL} = h_0, \text{ falls } a \in H_0 \\ \text{car}\,(\text{cons}\,(x_1, x_2)) = x_1 \end{cases}$$

bzw.

$$\begin{cases}\mathrm{cdr}\,(a) = h_0, \text{ falls } a \in H_0\\ \mathrm{cdr}\,(\mathrm{cons}\,(x_1, x_2)) = x_2.\end{cases}$$

Bereits in diesen Definitionen wurde die Primitiv-Rekursivität der Identitätsfunktion $f(x)=x$ benutzt. Diese kann durch

$$\begin{cases}\mathrm{id}\,(a) = a, \text{ falls } a \in H_0\\ \mathrm{id}\,(\mathrm{cons}\,(x_1, x_2)) = \mathrm{cons}\,(x_1, x_2)\end{cases}$$

definiert werden, wobei jede Konstante $a \in H_0$ und die Nachfolgerfunktion „cons" zu den Ausgangsfunktionen gehört.

Die festgelegten Elemente

$$h_0, h_1, h_2, \ldots$$

sind primitiv-rekursiv, da dies für das Atom h_0 gilt, und gilt es bereits für h_n, so vererbt es sich auch auf das durch die Substitution cons (h_n, h_n) entstehende h_{n+1}.

Die charakteristische Funktion

$$\mathrm{atom}\,(x) = \mathrm{sig}\,(x)$$

der Eigenschaft „ein 0-Element", d. h. „ein Element von H_0", d. h. „ein Atom zu sein", und das entgegengesetzte $\overline{\mathrm{sig}}\,(x)$ können durch folgende primitive Rekursionen definiert werden:

$$\begin{cases}\mathrm{sig}(a) = h_0, \text{ falls } a \in H_0\\ \mathrm{sig}(\mathrm{cons}(x_1, x_2)) = h_1,\end{cases} \qquad \begin{cases}\overline{\mathrm{sig}}\,(a) = h_1, \text{ falls } a \in H_0\\ \overline{\mathrm{sig}}\,(\mathrm{cons}(x_1, x_2)) = h_0.\end{cases}$$

Wird im letzteren für x die charakteristische Funktion $r(u_1, \ldots, u_n)$ einer Relation $R(u_1, \ldots, u_n)$ eingesetzt, so erhält man in $\overline{\mathrm{sig}}\,(r(u_1, \ldots, u_n))$ die charakteristische Funktion der Negation $\bar{R}(u_1, \ldots, u_n)$ dieser Relation.

Auch die zweistelligen logischen Operationen (die Disjunktion $R_1 \vee R_2$, die Konjunktion $R_1 \& R_2$, die Implikation $R_1 \rightarrow R_2$) sind primitiv-rekursiv, wie das in [2] allgemein bewiesen wurde.

Ebenfalls wurde dort allgemein bewiesen, daß samt den Funktionen $g_1, g_2, \ldots, g_{m+1}$ und den Relationen $R_1, R_2, \ldots, R_m$, wobei an jeder Stelle höchstens eine der genannten Relationen gilt, auch die aus ihnen folgenderweise „zusammengeflickte" Funktion f in H primitiv-rekursiv ist:

$$f(x_1, \ldots, x_n) = \begin{cases}g_1(x_1, \ldots, x_n), \text{ falls } R_1(x_1, \ldots, x_n)\\ \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\\ g_m(x_1, \ldots, x_n), \text{ falls } R_m(x_1, \ldots, x_n)\\ g_{m+1}(x_1, \ldots, x_n) \text{ sonst}\end{cases}$$

(wobei „sonst" auch mit dem Bestehen einer primitiv-rekursiven Relation

$$R_{m+1}(x_1, \ldots, x_n) = \bar{R}_1(x_1, \ldots, x_n) \& \ldots \& \bar{R}_m(x_1, \ldots, x_n)$$

gleichbedeutend ist).

Hier bemerke ich, daß die Fallunterscheidungsform (die „konditionelle" Definition) von McCarthy, die in [1] benutzt wird, der folgende Spezialfall dieser Definition ist:

$$f(x_1, \ldots, x_n) = \begin{cases} g_1(x_1, \ldots, x_n), \text{falls } R_1(x_1, \ldots, x_n) \\ g_2(x_1, \ldots, x_n), \text{ falls } \bar{R}_1(x_1, \ldots, x_n) \,\&\, R_2(x_1, \ldots, x_n) \\ \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \\ g_m(x_1, \ldots, x_n), \text{falls } \bar{R}_1(x_1, \ldots, x_n) \,\&\, \ldots \,\&\, \bar{R}_{m-1}(x_1, \ldots, x_n) \\ \qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad \&\, R_m(x_1, \ldots, x_n). \end{cases}$$

Sind hier nicht alle Möglichkeiten erschöpft, so ist f im „Sonstfall" undefiniert; könnte aber beliebig, z. B. als h_0 definiert werden, da dies in den Anwendungen immer belanglos ist.

7. Zu den weiteren wird auch die primitiv-rekursive Definition der charakteristischen Funktion der Gleichheit:

$$\text{equal}(x, y) = \begin{cases} h_0, \text{falls } x = y \\ h_1 \text{ sonst} \end{cases}$$

gebraucht.

Zuerst können die charakteristischen Funktionen der Eigenschaften: „mit je einem Atom a_i gleich zu sein" durch die primitiven Rekursionen

$$\begin{cases} f(a_i) = h_0 \\ f(a_j) = h_1 \text{ für jedes von } a_i \text{ verschiedene } a_j \in H_0 \\ f(\text{cons}(x_1, x_2)) = h_1 \end{cases}$$

definiert werden.

Dann kann die charakteristische Funktion $\text{eq}(x, y)$ der Relation, daß x ein mit y übereinstimmendes Atom ist, durch die „Zusammenflickung"

$$\text{eq}(x, y) = \begin{cases} h_0, \text{falls } (x = a_1 \,\&\, y = a_1) \vee \ldots \vee (x = a_t \,\&\, y = a_t) \\ h_1 \text{sonst} \end{cases}$$

definiert werden.

Damit läßt sich $\text{equal}(x, y)$ wie folgt definieren:

Sei

$$g(x, y) = \begin{cases} h_0, \text{falls } x = h_0 \,\&\, y = h_0 \\ h_1 \text{ sonst} \end{cases}$$

(da h_0 ein Atom ist, ist g eine primitiv-rekursive Funktion), dann gilt

$$\begin{cases} \text{equal}(a, y) = \text{eq}(a, y), \text{falls } a \in H_0 \\ \text{equal}(\text{cons}(x_1, x_2), y) = g\big(\text{equal}(x_1, \text{car}(y)), \text{equal}(x_2, \text{cdr}(y))\big). \end{cases}$$

Das ist aber keine primitive Rekursion, da dabei der Parameter y nicht unverändert bleibt, sondern dafür einmal $\text{car}(y)$ einmal $\text{cdr}(y)$ eingesetzt wird.

Ich habe zwar in [2] allgemein gezeigt, daß — bei eventueller Hinzunahme gewisser neuen Ausgangsfunktionen — eine derartige Definition auf die sog.

„Wertverlaufsrekursion", und diese auf primitive Rekursion (nebst Substitution) zurückgeführt werden kann; aber dabei wurde auch die charakteristische Funktion der Gleichheit benutzt. Diese (also equal (x, y)) wird daher auch hier zu den Ausgangsfunktionen hinzugenommen. Die Aufnahme weiterer Ausgangsfunktionen ist aber in unserem Spezialfall überflüssig; dies zeige ich in den folgenden.

8. Ich schicke voraus, daß es überflüssig ist in den einzelnen Fällen der „Zusammenflickung" immer wieder derartige Hilfsfunktionen einzuführen, wie g in der Definition von equal (x, y) war. Man kann sich z. B. ein für allemal klarlegen, daß auch die folgende „erweiterte Zusammenflickung" (wobei auch Parameter auftreten können), wo sich die verwendeten Relationen gegenseitig ausschließen, nicht von der Klasse der primitiv-rekursiven Funktionen hinausführt:

$$\begin{cases} f(a) = g_a, \text{falls } a \in H_0 \\ f(\text{cons}(x_1, x_2)) = \begin{cases} g_1'(x_1, x_2, f(x_1), f(x_2)), \text{falls } R_1(x_1, x_2, f(x_1), f(x_2)) \\ \dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots \\ g_m'(x_1, x_2, f(x_1), f(x_2)), \text{falls } R_m(x_1, x_2, f(x_1), f(x_2)) \\ g_{m+1}'(x_1, x_2, f(x_1), f(x_2)) \text{ sonst;} \end{cases} \end{cases}$$

denn mit

$$g(x_1, x_2, v_1, v_2) = \begin{cases} g_1'(x_1, x_2, v_1, v_2), \text{falls } R_1(x_1, x_2, v_1, v_2) \\ \dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots \\ g_m'(x_1, x_2, v_1, v_2), \text{falls } R_m(x_1, x_2, v_1, v_2) \\ g_{m+1}'(x_1, x_2, v_1, v_2) \text{ sonst} \end{cases}$$

kann $f(x)$ durch die folgende primitive Rekursion definiert werden:

$$\begin{cases} f(a) = g_a, \text{falls } a \in H_0 \\ f(\text{cons}(x_1, x_2)) = g(x_1, x_2, f(x_1), f(x_2)). \end{cases}$$

Als Spezialfälle der „erweiterten Zusammenflickung" wurden in [2] bei primitivrekursivem $R(x, u_1, \dots, u_n)$ auch die Relationen: „Es gibt ein Vorgänger von x, für den R besteht":

$$(\exists y)[y \leqq x \,\&\, R(y, u_1, \dots, u_n)],$$

und „für alle Vorgänger von x gilt R":

$$(\forall y)[y \leqq x \to R(y, u_1, \dots, u_n)],$$

ferner die Funktion:

$$\mu_y[y \leqq x \,\&\, R(y, u_1, \dots, u_n)]$$

als primitiv-rekursiv definiert, wobei das letzte den ersten in einer bestimmten Reihenfolge der Vorgänger von x bedeutet, für den R besteht, falls ein solcher Vorgänger von x existiert; und sonst als h_0 definiert wird. (Ist x eine natürliche Zahl, dann bedeutet so y die kleinste natürliche Zahl gewünschter Art bis x.)

Als jene bestimmte Folge (mit erlaubten Wiederholungen) der Vorgänger von x ist es in unserem Spezialfall zweckmässig die Folge

$$(*) \qquad \bar{x}_0, \bar{x}_1, \dots, \bar{x}_b$$

zu wählen, worin für $x \in H_0$ natürlich

$$b = 0 \quad \text{und} \quad \bar{x}_0 = x$$

gilt, und für ein $x = \text{cons}(x_1, x_2)$ erst die Vorgänger von x_2 (in der betreffenden Reihenfolge), dann die Vorgänger von x_1 (in der betreffenden Reihenfolge) auftreten, und endlich

$$x = \bar{x}_b.$$

9. Mit Benutzung von equal (x, y) kann nun die charakteristische Funktion vorg (x, y) der Relation $y \leqq x$ wie folgt als eine primitiv-rekursive Funktion definiert werden:

$$\begin{cases} \text{vorg}(a, y) = \text{eq}(y, a), \text{ falls } a \in H_0 \\ \text{vorg}(\text{cons}(x_1, x_2), y) = \begin{cases} h_0, \text{ falls } \text{cons}(x_1, x_2) = y \vee \text{vorg}(x_1, y) = h_0 \vee \\ \qquad\qquad \vee \text{vorg}(x_2, y) = h_0 \\ h_1 \text{ sonst.} \end{cases} \end{cases}$$

Damit ist auch die Relation

$$y \prec x \equiv y \leqq x \,\&\, y \neq x$$

primitiv-rekursiv.

Werden nun die natürlichen Zahlen

$$0, 1, 2, \ldots$$

der Reihe nach mit

$$h_0, h_1, h_2, \ldots$$

identifiziert, so gilt für jede natürliche Zahl i:

$$o(h_i) = h_i = i = o(i),$$

ferner ist

$$o(x) \prec o(y)$$

mit

$$o(x) \prec o(y)$$

gleichbedeutend, und aus

$$x \leqq o(y)$$

folgt, daß x eine natürliche Zahl, also

$$x = o(x)$$

ist; und $o(x)$ kann wie folgt als eine primitiv-rekursive Funktion definiert werden:

$$\begin{cases} o(a) = h_0, \text{ falls } a \in H_0 \\ o(\text{cons}(x_1, x_2)) = \begin{cases} \text{cons}(o(x_1), o(x_1)), \text{ falls } o(x_2) \leqq o(x_1) \\ \text{cons}(o(x_2), o(x_2)) \text{ sonst.} \end{cases} \end{cases}$$

Für $x = \text{cons}(x_1, x_2)$ wurde $o(x)$ eigentlich mit Verwendung des früheren Wertes $o(x^{-1})$ definiert, wobei durch x^{-1} unter x_1, x_2 ein bestimmtes von nicht-kleinerer Ordnung bezeichnet wird (dessen Ordnung um 1 kleiner als $o(x)$ ist).

In den weiteren wird auch benutzt, daß das Schema (wo auch Parameter auftreten können)

$$\begin{cases} f(a) = g_a, \text{ falls } a \in H_0 \\ f(\text{cons}(x_1, x_2)) = g\big(x_1, x_2, f(x_1), f(x_2), f(\text{cons}^{-1}(x_1, x_2))\big) \end{cases}$$

nicht von der Klasse der in H primitiv-rekursiven Funktionen hinausführt. Denn mit Verwendung der primitiv-rekursiven Hilfsfunktion

$$g'(x_1, x_2, v_1, v_2,) = \begin{cases} g(x_1, x_2, v_1, v_2, v_1), \text{ falls } o(x_2) \leqq o(x_1) \\ g(x_1, x_2, v_1, v_2, v_2) \text{ sonst} \end{cases}$$

kann $f(x)$ durch die primitive Rekursion

$$\begin{cases} f(a) = g_a, \text{ falls } a \in H_0 \\ f(\text{cons}(x_1, x_2)) = g'(x_1, x_2, f(x_1), f(x_2)) \end{cases}$$

definiert werden.

Im Spezialfall $g_a = a$ und $g(x_1, x_2, v_1, v_2, v_3) = v_3$ ergibt sich selbst x^{-1} als primitiv-rekursive Funktion.

10. Mit Verwendung von $o(x)$ wurde in [2] der — im folgenden als *Satz in* [2] zitierte — Satz erhalten, daß jede zahlentheoretische primitiv-rekursive Funktion zu einer primitiv-rekursiven Funktion in H erweitert werden kann. Genauer: Es gibt zu jeder primitiv-rekursiven zahlentheoretischen Funktion $\varphi(m_1, \ldots, m_n)$ eine primitiv-rekursive Funktion $f(x_1, \ldots, x_n)$ in H, für welche

$$o(f(x_1, \ldots, x_n)) = \varphi(o(x_1), \ldots, o(x_n))$$

gilt. Daraus folgt, daß die in H primitiv-rekursive Funktion

$$g(x_1, \ldots, x_n) = o(f(x_1, \ldots, x_n))$$

und die zahlentheoretische primitiv-rekursive Funktion φ für natürliche Zahlen als Argumente übereinstimmen. Derartige „Vertreter" der zahlentheoretischen primitiv-rekursiven Funktionen in H werden hier ebenso bezeichnet, wie die Funktionen die sie vertreten.

Statt einer beliebigen natürlichen Zahl n ist in H passender $o(x)$ zu gebrauchen. Z. B. wird die $o(x)$-te Iterierte einer primitiv-rekursiven Funktion $f(y)$ folgenderweise als primitiv-rekursive Funktion definiert:

$$\begin{cases} f^{(o(a))}(y) = y, \text{ falls } a \in H_0 \\ f^{(o(\text{cons}(x_1, x_2)))}(y) = f(f^{(o(\text{cons}^{-1}(x_1, x_2)))}(y)). \end{cases}$$

Ich bemerke hier, daß die nicht 0-ten Iterierten von $o(y)$ alle gleich $o(y)$ sind, da $o(y)$ immer eine natürliche Zahl ist, und so

$$o(o(y)) = o(y)$$

gilt.

11. Zur Behandlung der sog. Wertverlaufsrekursion hat man den endlichen Folgen der Elemente von H je ein Element derart zuzuordnen, daß daraus die Glieder der Folge wiedererkannt werden können. Für den Fall, wobei unter den Nach-

folgerfunktionen auch ein mindestens zweistelliger vorkommt, habe ich in [2] eine verhältnismäßig einfache derartige Zuordnung angegeben. In unserem Spezialfall liegt es aber an der Hand ein anderes Verfahren (das auch verallgemeinert werden kann) anzuwenden: die endlichen Elementenfolgen als Listen (von S-Ausdrücken) zu betrachten, welchen in den Früheren bereits je ein S-Ausdruck zugeordnet wurde:

$$\begin{aligned}
&\text{der aus } s_0 \quad \text{bestehenden Liste } (s_0 \cdot \mathrm{NIL}),\\
&\text{der aus } s_0, s_1 \quad \text{bertehenden Liste } (s_0 \cdot (s_1 \cdot \mathrm{NIL})),\\
&\text{der aus } s_0, s_1, s_2 \text{ bestehenden Liste } \big(s_0 \cdot (s_1 \cdot (s_2 \cdot \mathrm{NIL}))\big),\\
&\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\\
&\text{und der leeren Liste etwa} \qquad \mathrm{NIL} = h_0.
\end{aligned}$$

So wird einer Elementenliste

$$s_0, s_1, \ldots, s_n$$

ein Element

$$s = c_n(s_0, s_1, \ldots, s_n) = \mathrm{cons}\,(s_0, \mathrm{cons}\,(s_1, \ldots, \mathrm{cons}\,(s_n, h_0)\ldots))$$

zugeordnet. Man sieht, daß dabei

$$n = o(n) \leqq o(s)$$

gilt.

Aus s können die Glieder der Liste als primitiv-rekursive Funktionen zurückerhalten werden:

$$s_0 = \mathrm{car}\,(s),\ s_1 = \mathrm{car}\,(\mathrm{cdr}\,(s)),\ s_2 = \mathrm{car}\,(\mathrm{cdr}\,(\mathrm{cdr}\,(s))),\ \ldots,\ s_n = \mathrm{car}\,(\mathrm{cdr}^{(n)}\,(s)),$$

und es gilt

$$\mathrm{cdr}^{(n+1)}(s) = h_0.$$

Nach den Bisherigen kann die charakteristische Funktion der Eigenschaft „x entspricht einer Elementenliste“ wie folgt primitiv-rekursiv definiert werden:

$$\mathrm{list}(x) = \begin{cases} h_0, \text{ falls } x = h_0 \vee (\exists y)[y \leqq o(x) \;\&\; \mathrm{cdr}^{(o(y))}(x) \notin H_0 \;\&\; \mathrm{cdr}^{(o(y)+1)}(x) = h_0] \\ h_1 \text{ sonst} \end{cases}$$

und

$$\mathrm{long}(x) = \mu_y[y \leqq o(x) \;\&\; \mathrm{cdr}^{(o(y))}\,(x) \notin H_0 \;\&\; \mathrm{cdr}^{(o(y)+1)}(x) = h_0]$$

ergibt für ein S-Ausdruck x, der einer Liste zugeordnet ist, die „Länge“ dieser Liste (eigentlich um 1 weniger als die Gliederanzahl einer nicht leeren Liste, da das erste Glied der Liste den Index 0 erhalten hat). Für andere x gilt

$$\mathrm{long}\,(x) = h_0.$$

Jedenfalls ist long (x) eine natürliche Zahl, so daß

$$\mathrm{long}\,(x) = o\,(\mathrm{long}\,(x))$$

besteht.

Ist die Abhängigkeit einer primitiv-rekursiven Funktion f von x derartig, daß ihre Werte für jedes x dergleichen Ordnung übereinstimmen, so wird f (das auch

von anderen Variablen abhängen kann) auch als $f_{o(x)}$ geschrieben, und eine primitiv-rekursive Folge genannt.

Nun kann nach den Vorherigen eine primitiv-rekursive — und nur Vorgänger von x enthaltende — Folge, welche für solches x, das einer Liste entspricht, bei $o(y) \leqq \text{long}(x)$ das $o(y)$-te Glied dieser Liste liefert, z. B. wie folgt definiert werden:

$$k_{o(y)}(x) = \begin{cases} \text{car}\left(\text{cdr}^{(o(y))}(x)\right), \text{ falls } \text{list}(x) = h_0 \;\&\; o(y) \leqq \text{long}(x) \\ x \text{ sonst.} \end{cases}$$

Es gilt auch der folgende (zur Ausschaltung der Wertverlaufsrekursionen entscheidende) *Satz:* nicht nur die einzelnen Funktionen

$$c_n(s_0, s_1, \ldots, s_n) \quad \text{für} \quad n = 0, 1, 2, \ldots$$

sind primitiv-rekursiv, sondern für jede primitiv-rekursive Folge $s_{o(x)}$ (die auch von Parametern abhängen kann) ist

$${}^{(s)}c(x) = c_{o(x)}(s_0, s_1, \ldots, s_{o(x)})$$

eine primitiv-rekursive Funktion. Diese bezeichnet ja den S-Ausdruck

$$\text{cons}\,(s_0, \text{cons}\,(s_1, \ldots, \text{cons}\,(s_{o(x)-1}, \text{cons}\,(s_{o(x)}, h_0))\ldots)).$$

Deren Werte können von innen nach aussagen gehend berechnet werden. Dabei erhält man den sich sukzessiv verändernden Zwischenwert g zuerst (im „0-ten Schritt") als

$$h_0,$$

dann mit diesem g als

$$\text{cons}\,(s_{o(x)}, g),$$

dann mit diesem neuen g als

$$\text{cons}\,(s_{o(x)-1}, g),$$

usw.; im $i+1$-ten Schritt mit dem im i-ten Schritt erhaltenen g als

$$\text{cons}\,(s_{o(x)-i}, g),$$

für $i=0, 1, 2, \ldots, o(x)$.

Daher kann der Zwischenwert im $o(y)$-ten Schritt wie folgt als eine primitiv-rekursive Funktion definiert werden:

$$\begin{cases} g_{o(a)}(x) = h_0, \text{ falls } a \in H_0 \\ g_{o(\text{cons}(y_1, y_2))}(x) = \text{cons}\left(s_{o(x) \dot{-} o(\text{cons}^{-1}(y_1, y_2))}, g_{o(\text{cons}^{-1}(y_1, y_2))}(x)\right), \end{cases}$$

und damit auch

$${}^{(s)}c(x) = g_{\text{cons}\,(o(x),\, o(x))}(x)\,(= g_{o(x)+1}(x)).$$

Der Index

$$o(x) \dot{-} o\left(\text{cons}^{-1}(y_1, y_2)\right)$$

in der Definition von $g_{o(y)}(x)$ ist exakt folgenderweise zu verstehen: Die zahlentheoretische Funktion $m \dot{-} n$ bedeutet die Differenz $m-n$, falls $m \geqq n$ ist, und 0 sonst (wo der letztere Fall hier ohne Belang ist). Nach *Satz in* [2] gibt es dazu eine in H primitiv-rekursive Funktion $f(x, y)$, so daß

$$o\left(f(x, y)\right) = o(x) \dot{-} o(y)$$

gilt. So kann der betreffende Index von s durch

$$o(f(x, \mathrm{cons}^{-1}(y_1, y_2)))$$

vertreten werden. (In ähnlichem Sinn werden auch in den weiteren zahlentheoretische Funktionen verwendet.)

12. Die allgemeine Form des in unserem Spezialfall eben bewiesenen *Satzes* wurde in [2] zur Zurückführung der „Wertverlaufsrekursion" auf primitive Rekursion benutzt. Da handelt es sich um ein Rekursionschema, wobei zur Definition des Funktionswertes an einer Stelle x Funktionswerte an beliebigen echten, nicht nur an unmittelbaren Vorgängern von x verwendet werden.

Hierzu wird eine bestimmte Liste aus sämtlichen Vorgängern je eines Elementen x in Betracht genommen; in unserem Spezialfall die in Nr. **8** angegebene Folge

$$(*) \qquad \bar{x}_0, \bar{x}_1, \ldots, \bar{x}_b = x.$$

Das der aus den Gliedern

$$f(\bar{x}_0, u_1, \ldots, u_n),\ f(\bar{x}_1, u_1, \ldots, u_n),\ \ldots,\ f(\bar{x}_b, u_1, \ldots, u_n)$$

bestehenden Liste zugeordnete

$$f^*(x, u_1, \ldots, u_n) = c_b(f(\bar{x}_1, u_1, \ldots, u_n), \ldots, f(\bar{x}_b, u_1, \ldots, u_n))$$

kann die „Wertverlaufsfunktion" der Funktion $f(x, u_1, \ldots, u_n)$ genannt werden; daraus ergibt sich für jedes $i \leqq b$

$$f(\bar{x}_i, u_1, \ldots, u_n) = k_i(f^*(x, u_1, \ldots, u_n)),$$

und das Schema der Wertverlaufsrekursion lautet hier:

$$\begin{cases} f(a, u_1, \ldots, u_n) = g_a(u_1, \ldots, u_n), \text{ falls } a \in H_0 \\ f(\mathrm{cons}(x_1, x_2), u_1, \ldots, u_n) = g(x_1, x_2, u_1, \ldots, u_n, f^*(x_1, u_1, \ldots, u_n), f^*(x_2, u_1, \ldots, u_n)), \end{cases}$$

wobei die Funktionen g_a, g primitiv-rekursiv sind.

Wird bewiesen, daß

$$f^*(x, u_1, \ldots, u_n) \quad \text{und} \quad b = b(x)(=o(b(x)))$$

primitiv-rekursiv sind, so ist auch

$$f(x, u_1, \ldots, u_n) = k_{b(x)}(f^*(x, u_1, \ldots, u_n))$$

primitiv-rekursiv.

Überblicke man noch einmal die folgenden: Für ein $x = \mathrm{cons}(x_1, x_2)$ folgen in $(*)$ erst die Vorgänger von x_2 nacheinander; die letzte unter diesen erhält den Index $b(x_2)$. Danach folgen die Vorgänger von x_1, der Reihe nach mit Indizes

$$b(x_2)+1, \ldots, b(x_2)+b(x_1)+1.$$

Dann folgt noch x, das den Index

$$b(x) = b(x_2)+b(x_1)+2$$

erhält. So kann $b(x)$ (mit Anwendung des *Satzes in* [2]) durch folgende primitive Rekursion definiert werden:

$$\begin{cases} b(a) = 0, \text{ falls } a \in H_0 \\ b(\text{cons}(x_1, x_2)) = b(x_2) + b(x_1) + 2. \end{cases}$$

Jeder echte Vorgänger eines $x = \text{cons}(x_1, x_2)$ ist Vorgänger entweder von x_1 oder von x_2. Bezeichne für solches x und für $j < b(x)$ (andere Fälle sind belanglos) $i(x, j)$ den Wert 2 oder 1, je nachdem $\bar{x}_j \leqq x_2$ oder $\bar{x}_j \leqq x_1$ gilt; und $l(x, j)$ einen solchen Index, mit dem $\bar{x}_j$ in der zu $x_{i(x,j)}$ gehörigen Liste $(*)$ vorkommt. In [2] wurde mit allgemeiner Fassung der hier verwendeten Begriffe bewiesen, daß falls der Entsprechende des *Satzes* in Nr. **11** besteht, die Primitiv-Rekursivität der Entsprechenden von $i(x, j)$ und $l(x, j)$ die Primitiv-Rekursivität der entsprechenden Wertverlaufsrekursion nach sich zieht (deshalb sind diese notwendigerfalls zu den Ausgangfunktionen hinzuzunehmen).

In unserem Spezialfall können aber diese wie folgt als primitiv-rekursive Funktionen definiert werden:

$$i(x, j) = \begin{cases} 2, \text{ falls } x = \text{cons}(x_1, x_2) \,\&\, o(j) \leqq b(x_2) \\ 1 \text{ sonst (also auch für den uns allein} \\ \quad \text{interessierenden Fall } b(x_2) \prec o(j) \prec b(x) \end{cases}$$

und

$$l(x, j) = \begin{cases} o(j) \dot{-} (b(x_2) + 1), \text{ falls } x = \text{cons}(x_1, x_2) \,\&\, i(x, j) = 1 \\ o(j) \text{ sonts (also auch für } i(x, j) = 2). \end{cases}$$

So führt die Wertverlaufsrekursion nicht von der Klasse der in H primitiv-rekursiven Funktionen hinaus.

13. In Nr. **7** wurde erwähnt, daß eine derartige Rekursion, wie dort für equal angegeben war:

$$\text{(D)} \qquad \begin{cases} \text{equal}(a, y) = \text{eq}(a, y), \text{ falls } a \in H_0 \\ \text{equal}(\text{cons}(x_1, x_2), y) = g\big(\text{equal}(x_1, \text{car}(y)), \text{equal}(x_2, \text{cdr}(y))\big), \end{cases}$$

wobei für den Parameter y Einsetzungen erfolgten, nach eventueller Aufnahme neuer Ausgangsfunktionen auf Wertverlaufsrekursionen zurückgeführt werden kann.

Obwohl zu den hinzuzunehmenden Ausgangsfunktionen auch equal (x, y) selber gehört, möchte ich den Gedankengang des Beweises an diesem einfachen Beispiel andeuten.

Das erste Ziel ist eine Funktion $f(x, y)$ mit folgenden Eigenschaften zu definieren:

(1) Zu jedem y gibt es ein x mit

$$f(x, y) = y.$$

(2) Zu jedem $a \in H_0$, y und z gibt es ein x mit

$$f(x, y) = \text{eq}\,(a, f(z, y)).$$

(3) Zu jedem y und z gibt es ein x mit

$$f(x, y) = \text{car}\,(f(z, y)).$$

(4) Zu jedem y und z gibt es ein x mit

$$f(x, y) = \mathrm{cdr}\,(f(z, y)).$$

(5) Zu jedem y_1, z_1 und z_2 gibt es ein x mit

$$f(x, y) = g(f(z_1, y), f(z_2, y)).$$

Ein derartiges $f(x, y)$ nimmt sozusagen sämtliche „Bausteine" an, die in der Bestimmung je eines durch (D) definierten Wertes von equal (x, y) teilnehmen.

Nun kann ein solches $f(x, y)$ z. B. mit Verwendung der in Nr. **11** definierten primitiv-rekursiven Funktionen

$$k_i(u),\ c_n(u_0, \ldots, u_n) \quad \text{für} \quad i = 0, 1, 2;\ n = 1, 2,$$

wobei für $i \leqq n$

$$k_i(c_n(u_0, \ldots, u_n)) = u_i$$

gilt, wie folgt definiert werden:

$$\text{(D*)} \qquad \begin{cases} f(a, y) = y, \text{ falls } a \in H_0 \\ f(\mathrm{cons}(x_1, x_2), y) = \begin{cases} \mathrm{eq}\big(k_0(x_1), f(k_1(x_1), y)\big), \text{ falls } o(k_0(x_1)) = h_0 \\ \mathrm{car}\big(f(k_1(x_1), y)\big), \text{ falls } o(k_0(x_1)) = h_1 \\ \mathrm{cdr}\big(f(k_1(x_1), y)\big), \text{ falls } o(k_0(x_1)) = h_2 \\ g\big(f(k_1(x_1), y), f(k_2(x_1), y)\big) \text{ sonst.} \end{cases} \end{cases}$$

Tatsächlich werden so die Forderungen (1)—(5) der Reihe nach z. B. mit

$$\begin{aligned} x &= h_0, \\ x &= \mathrm{cons}\,(c_1(a, z), x_2), \quad \text{falls} \quad a \in H_0, \\ x &= \mathrm{cons}\,(c_1(h_1, z), x_2), \\ x &= \mathrm{cons}\,(c_1(h_2, z), x_2), \\ x &= \mathrm{cons}\,(c_2(h_3, z_1, z_2), x_2) \end{aligned}$$

bei beliebigem x_2, z. B. $x_2 = h_0$ erfüllt.

Definitionen der Art (D*) können zu Wertverlaufsrekursionen umformuliert werden.

Nach einem Hilfsatz in [2] kann nun durch eine ähnliche Definition eine Funktion $w(x, z)$ der Eigenschaft

$$f(x, f(z, y)) = f(w(x, z), y)$$

angegeben werden. (Diese hat besonders dann eine entscheidende Rolle, wenn es sich statt einer so einfachen Definition wie (D) um eine sog. „eingeschachtelte Rekursion" handelt, wobei für Parameter von (an Vorgänger-Stellen angenommenen) Werten der zu definierenden Funktion abhängige Ausdrücke eingesetzt werden; der genannte Hilfsatz ermöglicht die Auflösung der „Einschachtelungen".)

Mit Benutzung dieser Funktion $w(x, z)$ kann endlich durch eine primitive Rekursion eine Funktion $q(x)$ definiert werden, welche aus den Werten vom durch (D*)

definierten $f(x, y)$ die Werte von equal (x, y) sozusagen „herausschöpft", d.h., für welche bei jedem x, y

$$\text{equal}\,(x, y) = f(q(x), y)$$

gilt.

Nach der Definition von $k_{o(y)}(x)$ sind die in (D*) auftretenden

$$k_i(x_1) \qquad (i=0, 1, 2)$$

Vorgänger von x_1. Ist $h_i(x)$ die $v(x, i)$-te in der Liste $(*)$ der Vorgänger von x, und wird die Wertverlaufsfunktion von $f(x, y)$ mit $f^*(x, y)$ bezeichnet, so gilt nach Nr. **12**

$$f(k_i(x_1), y) = k_{v(x_1, i)}(f^*(x_1, y)).$$

Werden für $i=0, 1, 2$ die rechten Seiten dieser Identitäten für ihre linken Seiten in (D*) eingesetzt, so sieht man, daß falls $v(x, i)$ primitiv-rekursiv ist, (D*) zu einer Wertverlaufsrekursion umformuliert werden kann. Deshalb wurde in [2] allgemein gefordert, das Entsprechende von $v(x, i)$ notwendigerfalls zu den Ausgangsfunktionen hinzuzunehmen.

In unserem Spezialfall kann aber $v(x, i)$ (von natürlichen Zahlen auf sämtliche S-Ausdrücke für i in belangloser Weise erweitert) als eine primitiv-rekursive Funktion definiert werden.

Die Liste $(*)$ der Vorgänger von x besteht für $x \in H_0$ aus dem einzigen Glied $\bar{x}_0 = x$, und für ein $x = \text{cons}\,(x_1, x_2)$ aus den Gliedern

$$\bar{x}_0, \ldots, \bar{x}_{b(x_2)} = x_2, \bar{x}_{b(x_2)+1}, \ldots, \bar{x}_{b(x_2)+b(x_1)+1} = x_1, \bar{x}_{b(x_2)+b(x_1)+2} = x,$$

wo die ersten $b(x_2)+1$ Glieder die Liste der Vorgänger von x_2, und die darauf folgenden $b(x_1)+1$ Glieder die Liste der Vorgänger von x_1 bilden.

So gilt erstens immer

$$v(x, i) = 0, \quad \text{falls} \quad x \in H_0.$$

Sei nun

$$x = \text{cons}\,(x_1, x_2).$$

Wenn x zu keiner Liste gehört, oder wenn $i \leqq \text{long}(x)$ nicht besteht, so gilt

$$k_i(x) = x,$$

also

$$v(x, i) = b(x_2)+b(x_1)+2.$$

Sonst ist x der Form:

$$x = \text{cons}\Big(\underbrace{s_0}_{x_1}, \underbrace{\text{cons}\big(s_1, \ldots, \text{cons}\,(s_{\text{long}(x)}, h_0)\ldots\big)}_{x_2}\Big),$$

daher gilt

$$k_0(x) = s_0 = x_1 = \bar{x}_{b(x_2)+b(x_1)+1},$$

also

$$v(x, 0) = b(x_2)+b(x_1)+1;$$

und für $0 < i \leqq \text{long}\,(x)$

$$k_i(x) = k_{i-1}(x_2),$$

also

$$(D_0) \qquad v(x, i) = v(x_2, i-1).$$

Dies bietet für $i>1$ eine Rekursion, wobei für den Parameter i die Einsetzung $i-1$ erfolgt. Doch so einfache Fälle lassen sich leicht auflösen: Für $i>1$ kann (D_0) wiederholt werden. Da

$$x_2 = \mathrm{cdr}\,(x) \qquad (\text{und } x_1 = \mathrm{car}\,(x))$$

ist, ergibt sich so

$$v(x, i) = v\big(\mathrm{cdr}(x), i-1\big) = v\big(\mathrm{cdr}^2(x), i-2\big) = \ldots = v\big(\mathrm{cdr}^{(i)}(x), 0\big) =$$

$$= b\big(\mathrm{cdr}^{(i+1)}(x) + b\big(\mathrm{car}\,(\mathrm{cdr}^{(i)}(x))\big) + 1.$$

Daher ergibt sich $v(x, i)$ durch die Fallunterscheidung

$$v(x, i) = \begin{cases} h_0, \text{ falls } x \in H_0 \\ b\big(\mathrm{cdr}^{(o(i)+1)}(x)\big) + b\big(\mathrm{car}\,(\mathrm{cdr}^{(o\,(i))}(x))\big) + 1, \\ \quad \text{falls } x \notin H_0 \;\&\; \mathrm{list}\,(x) = h_0 \;\&\; o(i) \leqq \mathrm{long}(x) \\ b\big(\mathrm{cdr}(x)\big) + b\big(\mathrm{car}(x)\big) + 2 \text{ sonst} \end{cases}$$

als eine primitiv-rekursive Funktion.

Nach dem angedeuteten Gedankengang führen daher die Rekursionen, wobei für die Parameter Einsetzungen erfolgen (sogar die derart zustandekommenden „eingeschachachtelten Rekursionen") nicht von der Klasse der primitiv-rekursiven Funktionen von H hinaus.

14. Als eine einfache Verwendung der Funktionen $c_{o\,(x)}$ und $k_{o\,(y)}(x)$ erwähne ich noch die in [2] behandelte Zurückführung der „simultan-rekursiven" Definition mehrerer Funktionen

$$f_0, f_1, \ldots, f_n$$

auf die rekursive Definition der einzigen Funktion

$$f = c_n(f_0, f_1, \ldots, f_n)$$

woraus die einzelnen Funktionen durch die folgenden Substitutionen zurückerhalten werden können:

$$f_0 = k_0(f),\ f_1 = k_1(f),\ \ldots, f_n = k_n(f).$$

Z. B. ergibt

$$\begin{cases} f_0(a) = g_a, \text{ falls } a \in H_0 \\ f_0\big(\mathrm{cons}(x_1, x_2)\big) = g\big(x_1, x_2, f_0(x_1), f_0(x_2), f_1(x_1), f_1(x_2)\big) \end{cases}$$

und

$$\begin{cases} f_1(a) = g'_a, \text{ falls } a \in H_0 \\ f_1\big(\mathrm{cons}(x_1, x_2)\big) = g'\big(x_1, x_2, f_0(x_1), f_0(x_2), f_1(x_1), f_1(x_2)\big) \end{cases}$$

eine simultan-rekursive Definition der Funktionen $f_0(x), f_1(x)$.

(Des würde auf dasselbe hinausgehen, wenn statt der letzten Definitionsgleichung

$$f_1\,(\mathrm{cons}\,(x_1, x_2)) = g''\big(x_1, x_2, f_0(x_1), f_0(x_2), f_0\big(\mathrm{cons}\,(x_1, x_2)\big), f_1(x_1), f_1(x_2)\big)$$

stehen würde; man hätte dann nur g' als

$$g'(x_1, x_2, u_1, u_2, u_3, u_4) = g''(x_1, x_2, u_1, u_2, g(x_1, x_2, u_1, u_2, u_3, u_4), u_3, u_4)$$

zu definieren.)

In diesem Fall kann

$$f(x) = c_1(f_0(x), f_1(x))$$

durch folgende primitive Rekursion definiert werden:

$$\begin{cases} f(a) = c_1(g_a, g'_a), \text{ falls } a \in H_0 \\ f(\text{cons}(x_1, x_2)) = c_1\Big(g\big(x_1, x_2, k_0(f(x_1)), k_0(f(x_2))\big), g'\big(x_1, x_2, k_1(f(x_1)), k_1(f(x_2))\big)\Big), \end{cases}$$

und daraus erhält man durch Substitutionen:

$$f_0(x) = k_0(f(x)) \quad \text{und} \quad f_1(x) = k_1(f(x)).$$

15. In [2] wurde auch gezeigt, daß die allgemein-rekursiven Funktionen (im allgemeinen H) ähnlich wie im zahlentheoretischen Fall definiert werden können, als Funktionen, deren Werte aus Definitionsgleichungsystemen — durch endlichmal verwendeter Einsetzung von Elementtermen (in unserem Spezialfall von S-Ausdrücken) für Variablen, und durch Ersetzung einer Gleichungseite die als ein Teilausdruck auftritt, durch die andere Seite der betreffenden Gleichung — überall eindeutig erhalten werden können; mit Weglassung der Forderung „überall" entstehen derart die partiell-rekursiven Funktionen in H. (All diese können auch durch primitive Rekursionen und „unbeschränkte μ-Operationen" definiert werden.)

16. Einige Beispiele für Definitionen von im „Lisp 1.5" verwendeten primitivrekursiven Funktionen in unserem H (diese sind in etwas abweichender Form auch in [1] zu finden):

① Der Ausdruck, der aus x entsteht, wenn für jedes Vorkommen von y darin z gesetzt wird (in [1] wurde das nur im Fall $y \in H_0$ definiert, doch ohne Belang kann die Definition für beliebige y erweitert werden; ähnliches gilt auch in den folgenden Definitionen):

$$\begin{cases} \text{subst}(a, y, z) = \begin{cases} z, \text{ falls } a \in H_0 \,\&\, a = y \\ a, \text{ falls } a \in H_0 \,\&\, a \neq y \end{cases} \\ \text{subst}(\text{cons}(x_1, x_2), y, z) = \text{cons}(\text{subst}(x_1, y, z), \text{subst}(x_2, y, z)). \end{cases}$$

② Der Ausdruck, der jener Liste entspricht, die aus einer Liste durch Danachfügung der Reihe nach der Glieder einer anderen Liste entsteht (der Fall

$$\text{list}(x) = h_1 \vee \text{list}(y) = h_1$$

ist für uns ohne Belang; und für $a \in H_0$ interessiert uns nur $a = h_0$ als das Entsprechende der leeren Liste):

$$\begin{cases} \text{append}(a, y) = y, \text{ falls } a \in H_0 \\ \text{append}(\text{cons}(x_1, x_2), y) = \text{cons}(x_1, \text{append}(x_2, y)). \end{cases}$$

③ Die charakteristische Funktion der Relation „unter den Gliedern einer gewissen Liste vorkommen" (für list $(x)=h_1$ ohne Belang):

$$\begin{cases}\text{member}(a, y) = h_1, \text{ falls } a \in H_0 \\ \text{member}\big(\text{cons}(x_1, x_2), y\big) = \begin{cases} h_0, \text{ falls } x_1 = y \\ \text{member}(x_2, y) \text{sonst.}\end{cases}\end{cases}$$

④ Der Ausdruck, der jener Liste entspricht, die durch linkseitiger Hinzufügung zu einer Liste der Glieder einer „Paarliste" entsteht, die aus den cons-Funktionen („Paare" genannt) der ersten, zweiten, ... Glieder zweier Listen derselben Länge entsteht:

$$\begin{cases}\text{pairlis}(a, y, u) = u \text{ falls } a \in H_0 \\ \text{pairlis}\big(\text{cons}(x_1, x_2), y, u\big) = \text{cons}\big(\text{cons}(x_1, \text{car}(y)), \text{pairlis}(x_2, \text{cdr}(y), u)\big).\end{cases}$$

Wegen dem Vorkommen von

$$\text{pairlis}\,(x_2, \text{cdr}\,(y), u)$$

auf der rechten Seite haben wir hier bereits mit einer solchen Rekursion zu tun, wobei für den Parameter y eine Einsetzung erfolgt ist; worüber es sich in Nr. **12—13** handelte.

Noch zwei Definitionen bezüglich Paarlisten:

⑤ Das erste Paar einer Paarliste, dessen erstes Glied (d. h. car-Funktion) x ist (und etwa h_0, falls kein solches Paar vorhanden ist):

$$\begin{cases}\text{assoc}(a, x) = h_0, \text{ falls } a \in H_0 \\ \text{assoc}\big(\text{cons}(u_1, u_2), x\big) = \begin{cases} u_1, \text{ falls car}(u_1) = x \\ \text{assoc}(u_2, x) \text{ sonst.}\end{cases}\end{cases}$$

⑥ Gehört u zu einer Paarliste, worin die ersten Glieder Paare Atome sind, so bedeutet

$$\text{sublis}\,(x, u)$$

den Ausdruck, der aus dem Ausdruck x entsteht, falls jedes Vorkommen der ersten Glieder der Paare in x durch das zweite Glied (d. h. cdr-Funktion) des betreffenden Paars ersetzt wird.

Erst kann dies mit der Bezeichnung „suba" für $x \in H_0$ durch Rekursion nach u definiert werden:

$$\begin{cases}\text{suba}(a, x) = x, \text{ falls } a \in H_0 \\ \text{suba}\big(\text{cons}(u_1, u_2), x\big) = \begin{cases} \text{cdr}(u_1), \text{ falls car}(u_1) = x \\ \text{suba}(u_2, x) \text{ sonst.}\end{cases}\end{cases}$$

Damit ergibt sich die primitive Rekursion nach x:

$$\begin{cases}\text{sublis}(a, u) = \text{suba}(u, a), \text{ falls } a \in H_0 \\ \text{sublis}\big(\text{cons}(x_1, x_2), u\big) = \text{cons}\big(\text{sublis}(x_1, u), \text{sublis}(x_2, u)\big).\end{cases}$$

Рекурсивность программного языка "Lisp 1,5" в специальных случаях упорядоченных свободных голоморфных множеств

Понятие, имеющееся в заглавии (понятие «множеств, которые могут быть построены по методу натуральных чисел») автор ввёл на варшавском симпозиуме в 1959 году; и для таких множеств он обобщил теорию рекурсивных функций; разрабатывая как частный случай, теорию рекурсивных функций во множествах, находящихся выше любого алфавита. И другие авторы исследовали и применяли в основном этот специальный случай. Легко показать, что основные понятия программного языка „Lisp 1.5" являются примитивно-рекурсивными во множестве, находящемся выше подходящегося алфавита. В то же время множество симболических выражений языка „Lisp 1.5", построенное исходя из некоторых «атомов» с помощью единственного (двухпеременного) действия, само собой даётся как специальный случай упорядоченных свободных голоморфных множеств, не совпадающих с множествами слов. В статье даётся рекурсивная теория для этого случая.

EÖTVÖS LORÁND UNIVERSITÄT
BUDAPEST, UNGARN

(Eingegangen am 1. Februar 1973)

Cybernetic methods in clinical neurophysiology

By L. GYERGYEK and J. K. TRONTELJ

This paper gives a short review of some of our recent activities which have been developed in our collaboration with the Institute of Clinical Neurophysiology at Ljubljana.

A part of our work was concerned with the development of new computer-oriented research techniques to implement the methods of electromyography (EMG) and microelectromyography (micro EMG).

In the second part of our paper we wish to present some of the results of our attempts to study memory and learning in simple neural networks in man.

Frequency analysis of EMG

EMG (Fig. 1) is an important diagnostic tool in clinical neurophysiology. As is well known, it shows a complex series of electrical events in the millivolt range produced by depolarization of muscle fibre preceding their contraction. It is an empirical knowledge that the time course and amplitudes of these voltage changes

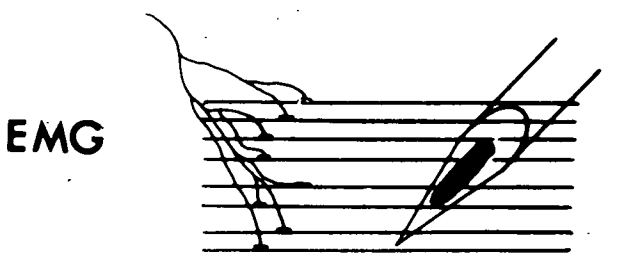


Fig. 1
Coaxial needle electrode recording motor unit action potentials

reflect the spatial and temporal patterns in which the muscle fibres are arranged in the motor units; and from these patterns conclusions can be drawn as to the normal or pathological innervation types.

While it is usually not difficult to recognize clearly abnormal cases, there is a large range of borderline changes which are left to subjective interpretation.

Frequency analysis based on several principles has already proved to be valuable in such cases.

Our aim was to develop an improved method of frequency analysis of the EMG signals.

We have chosen the principle of pattern recognition whereby the frequency content of the EMG signals is represented by a set of twelve-dimension vectors. The EMG signals are fed to twelve band-pass filters covering the whole frequency

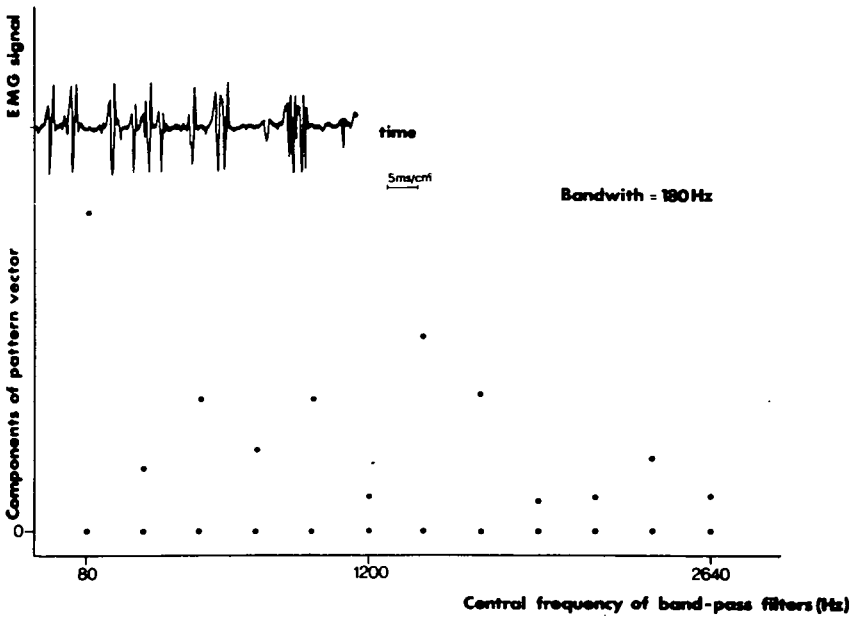


Fig. 2
Recorded EMG and its pattern vector

range; the output of these filters constitutes the components of the pattern vector (Fig. 2). We use a general purpose computer which allows an on-line classification with a pattern recognition method.

We have chosen a linear discriminant function

$$g(x, w) = w_1 x_1 + w_2 x_2 + \ldots + w_n x_n + w_{n+1}$$

where w_i are the weights of the threshold logic unit and x_i are components of the pattern vector.

A nonparametric training was used for determination of the weight vector W

$$W' = W + cX$$

where W' is a new value of W and c is the correction coeffcient.

This method is still under evaluation. Nevertheless it seems to be rather promising.

Computerized Micro EMG

Micro EMG is another method in clinical neurophysiology which has been improved by the introduction of on-line statistical calculations performed by a digital computer.

The basic method here is micro EMG which allows the study of the action potentials of single muscle fibres as well as of several muscle fibres innervated by one motor neurone. The interesting parameters in this method are the time intervals between the individual spike potentials, as well as their respective amplitudes, which serve for their recognition.

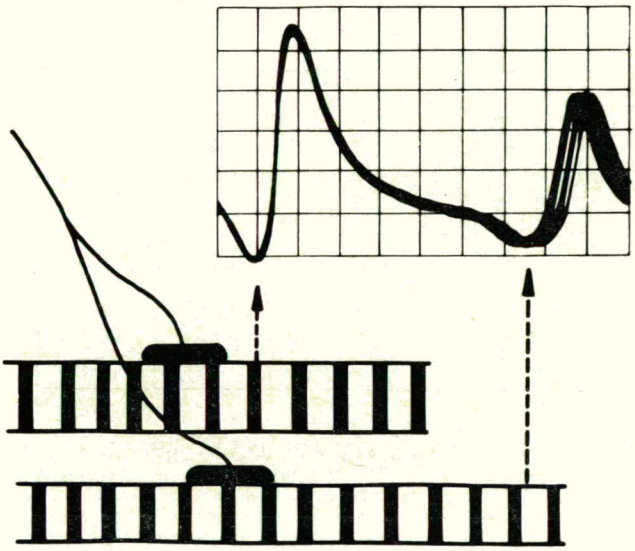


Fig. 3
Schematic representation of the recording of the neuromuscular jitter. The transmission of impulses across the motor end plates contributes the largest part to the total variation of the inter-potential intervals

Variation of the intervals between the individual spikes known as the neuromuscular jitter or the variation of the latency time of the spikes following a stimulus can provide an important information about the functions like the neuromuscular transmission, conduction of impulses along the nerve and muscle fibres, and synaptic transmission in the central nervous system (Fig. 3, 4).

The problem to be solved here was to identify the potentials and to compute statistical parameters of interspike time variation.

Fig. 5A) shows repetitive discharges of a pair of potentials generated by two muscle fibres. There is a small variation of the intervals between the two potentials, which is in the normal range, i.e. lower than 30 usec. Fig. 5B) shows a similar pair

of spikes but with a pathologically large variation of the intervals and intermittent blocking of the second spike.

The computer performs on-line the following statistics

1. histogram of intervals
2. floating average
3. distribution of intervals
4. dependence of the intervals on the discharge frequency and computes the respective numerical values (Fig. 6, 7, 8, 9).

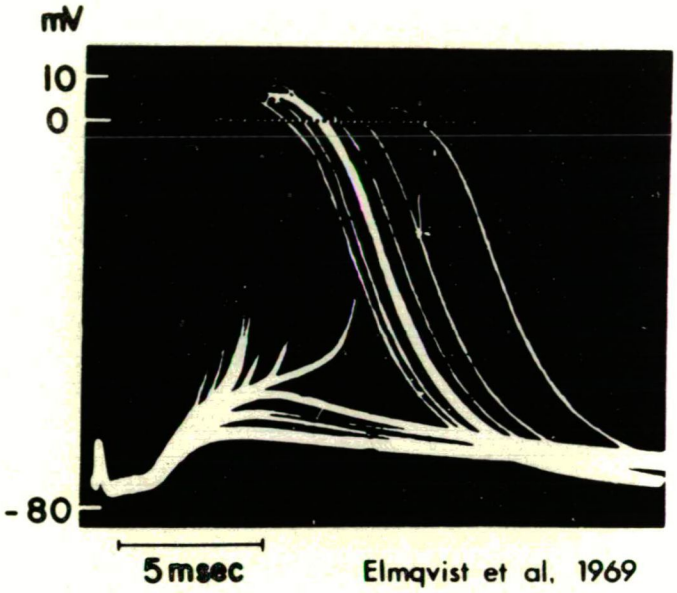


Fig. 4
Intracellular recording near the end plate showing the origin of the neuromuscular jitter. The action potentials of the muscle fibre are triggered when the end plate potential reaches the threshold level

The main difficulty to be solved was presented by the real-time operation which required fast algorithms for the recognition of the potentials and other computations.

We decided to use the distances between the imput pattern and a set of reference vectors as the classification criterion. Suppose that m reference vectors $R_1, R_2, \ldots, R_m$ are given with R_j associated with the pattern class ω_j. A minimum-distance classification scheme with respect to $R_1, R_2, \ldots, R_m$ is to classify the input X from ω_i, i.e. $X \in \omega_i$ if $|X-R_i|$ is the minimum, where $|X-R_i|$ is the Hamming's distance defined between X and R_i.

By the use of these computations we considerably increased the value of micro EMG for both diagnostic and research.

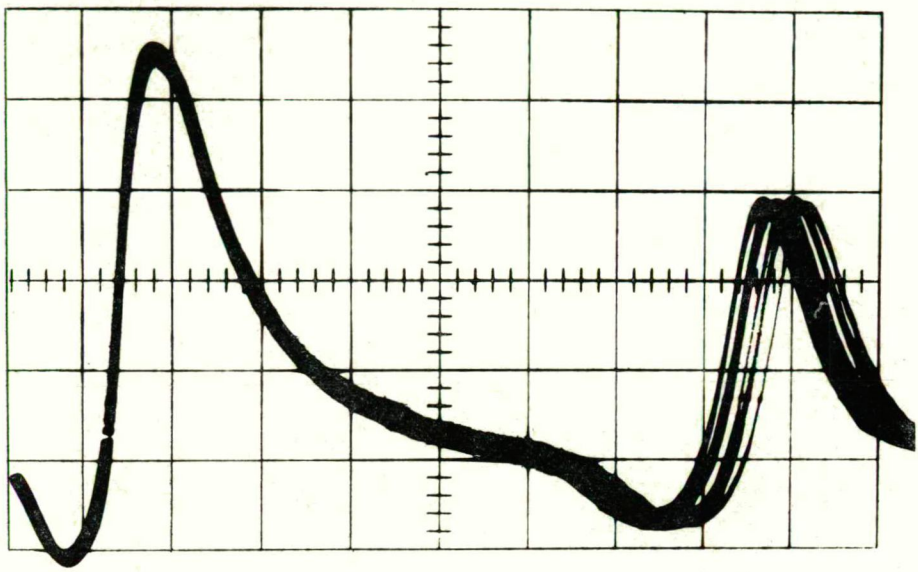


Fig. 5 A)
Action potentials of two muscle fibres of a motor unit;
a normal pair with small variation of interpotential intervals

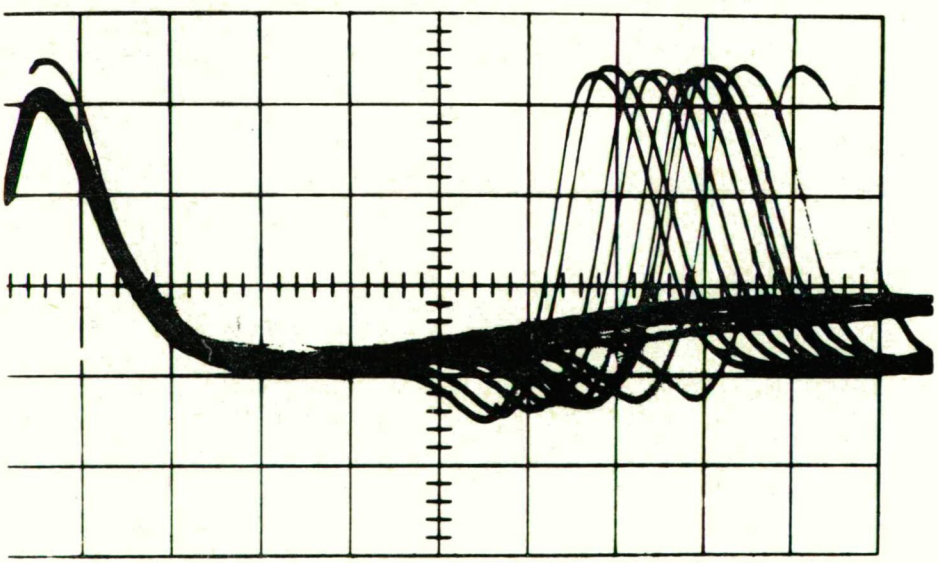


Fig. 5 B)
Action potentials of two muscle fibres of a motor unit;
a pathological pair with a large jitter and occasional blocking of the second potential

The problem of learning in simple polysynaptic reflexes

Initially, our study had a very simple aim: to investigate the properties of human flexion withdrawal reflex, particularly in the paraplegic patient with completely divided spinal cord. A classical definition says that it is a simple polysynaptic reflex subserving withdrawal of the limb from an offending stimulus. This reflex is organized at the level of the spinal cord (Fig. 10), but can be substantially influenced by the

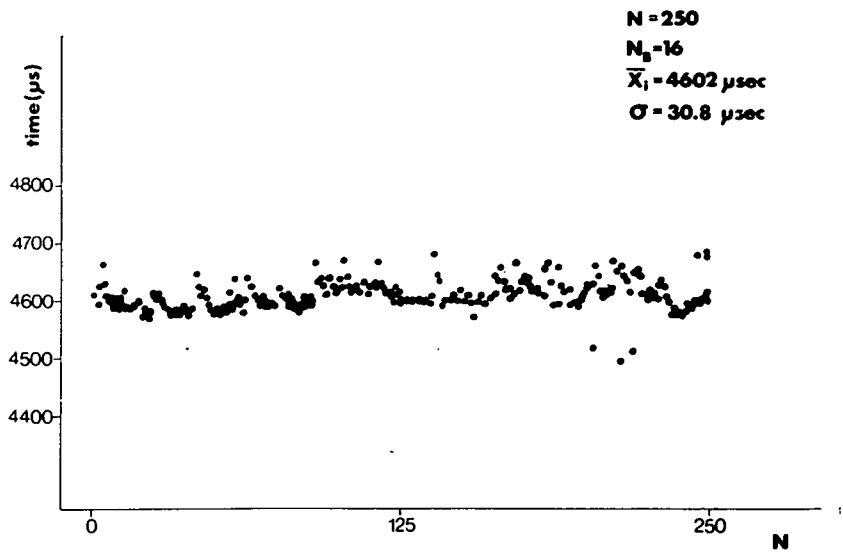


Fig. 6
Sequential histogram of interpotential intervals.
Each dot represents a single discharge of the potential pair

descending pathways from the brain. The paraplegic patient retains the flexion reflex, frequently in an exaggerated form, even though his spinal cord is divided from the rest of the central nervous system. The ambition of some research workers has been to use this reflex in the attempts to externally control the paralyzed lower limbs.

An immediate observation in our studies was that the flexion reflex, when repetitively elicited by electrical noxious stimuli in paraplegic patients, exhibits an unexpectedly large variation in size, even though stimulation parameters and other controllable experimental conditions are kept as constant as possible. Another regular finding was the tendency towards gradual decline of the size of the responses followed by their eventual cessation (Fig. 11). Closer study revealed all features of typical habituation[1], which, according to some authors[2], can be regarded as a primitive form of learning. Others, however, object against this view and rather tend to interpret habituation as a passive process of synaptic fatique[3].

Considering that the flexion reflex probably is one of the simplest systems capable of habituation we became aware of the opportunities which it offers in the

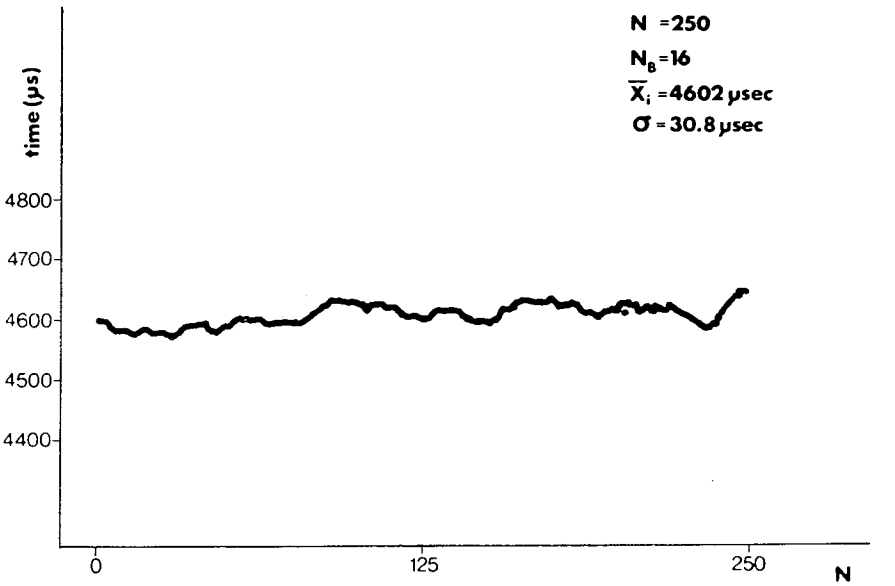


Fig. 7
Floating average of the intervals in Fig. 6

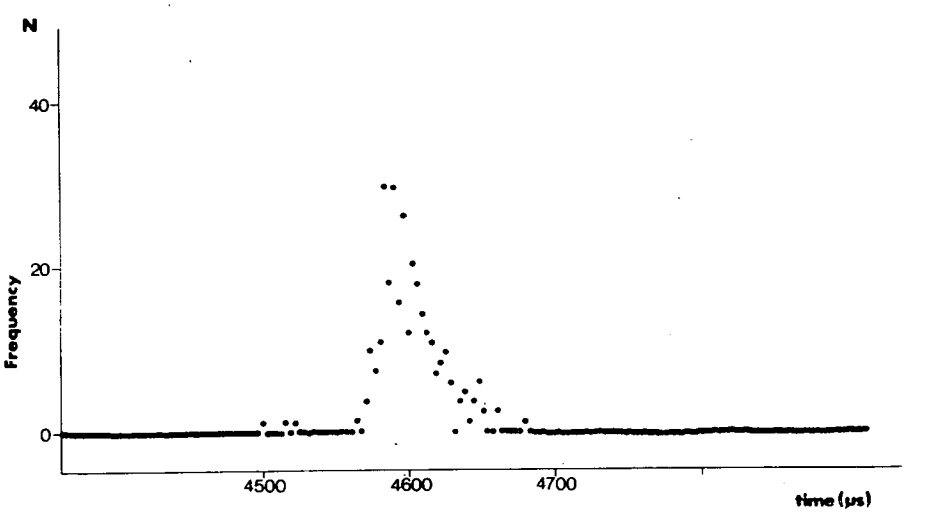


Fig. 8
Distribution histogram of the intervals in Fig. 6

studies of habituation, particularly in the basic neurophysiological problem: whether or not habituation is true learning.

Approaching this problem we first attempted to influence the process of habituation by stochastically changing parameters of stimulation[4] (Fig. 12). This method always resulted in slowing or precluding habituation. A most striking result is illustrated by Fig. 13, which shows responses to a series of equal stimuli, followed by a series of stimuli with stochastically changing amplitude, which however never exceeded that used in the first series. It can be easily seen that habituation was very effectively suppressed in the stochastical series.

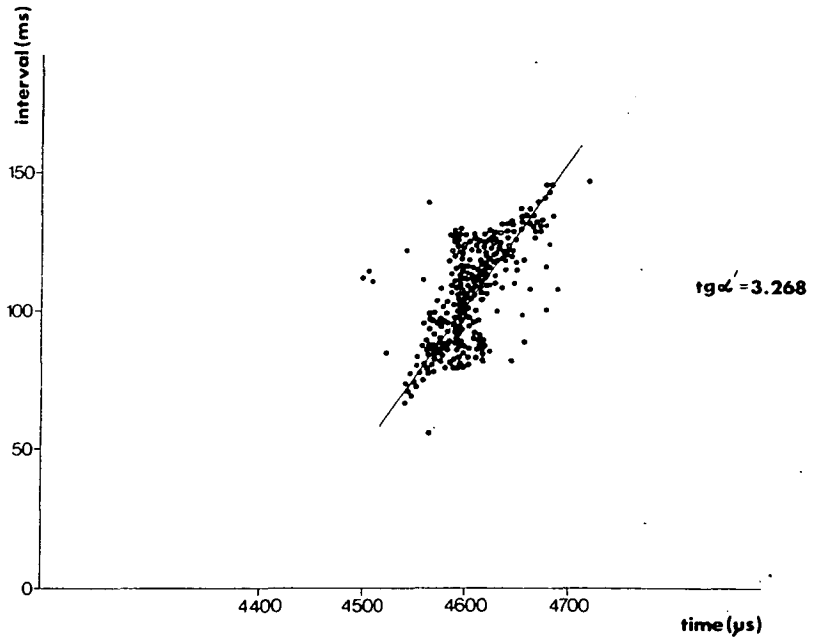


Fig. 9
Relationship between the interpotential intervals and the interdischarge intervals

This observation is a strong argument against the model of habituation based on passive synaptic fatigue. In this model, which incidentally has evolved from a similar experimental situation, i.e. flexion reflex in the spinal cat[5], dishabituation is only achieved with an extrastimulus, or a stronger stimulus, which activates additional interneurones with fresh synaptic endings (Fig. 14). Evidently this logic cannot be applied to our case; weaker stimuli should excite less afferents and less interneurones. Instead, we interpreted this finding to be due to some kind of recognition of stimulus strength, with a consequence that even a weaker stimulus acquires the meaning of a novelty necessary to interrupt the process of habituation. This concept implies the existence of another two functions in the system of the flexion reflex: classifying and memorizing the incoming information.

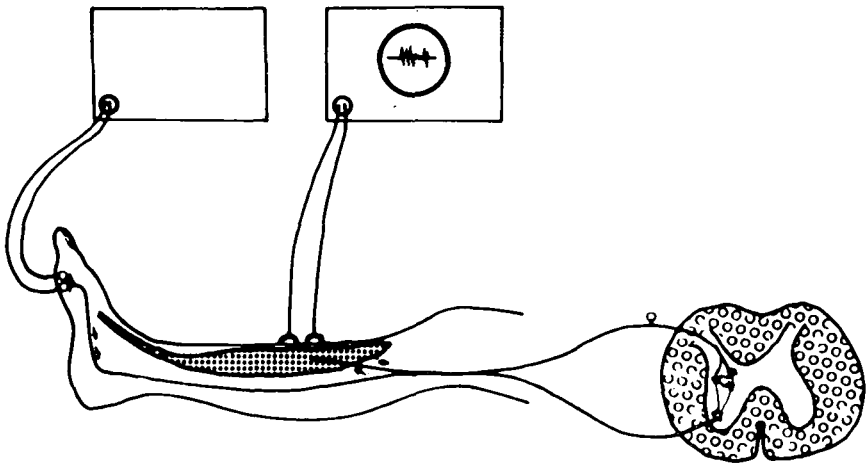


Fig. 10
Schematic representation of the reflex arc of the flexion reflex

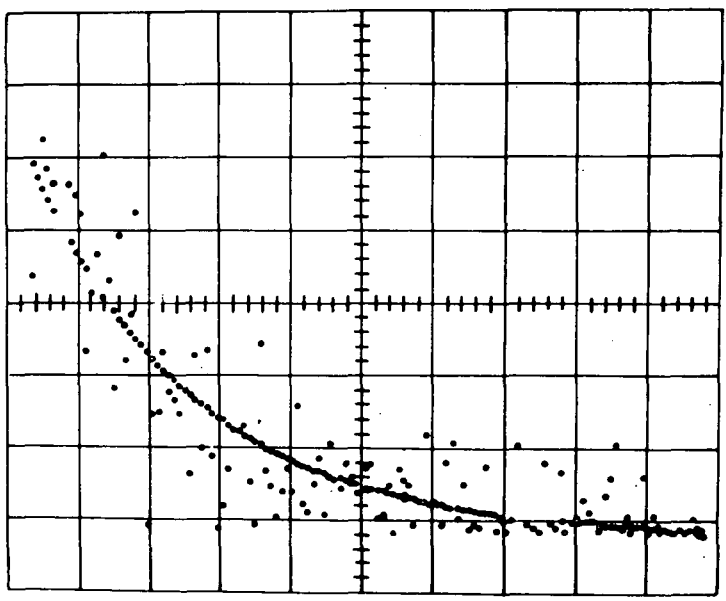


Fig. 11
Decline of the magnitude of the repetitively elicited flexion reflex responses. The line in the histogram is the minimum square error exponential curve

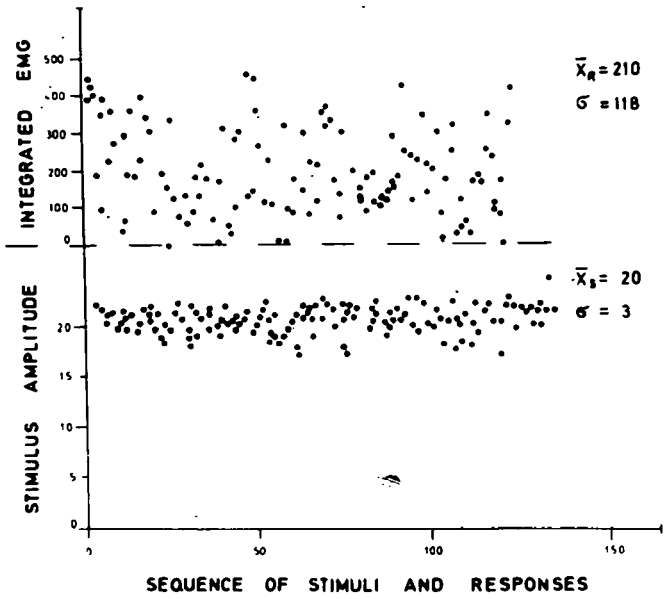


Fig. 12
Stochastically changing stimuli (upper histogram) and the respective responses (lowtoers higram)

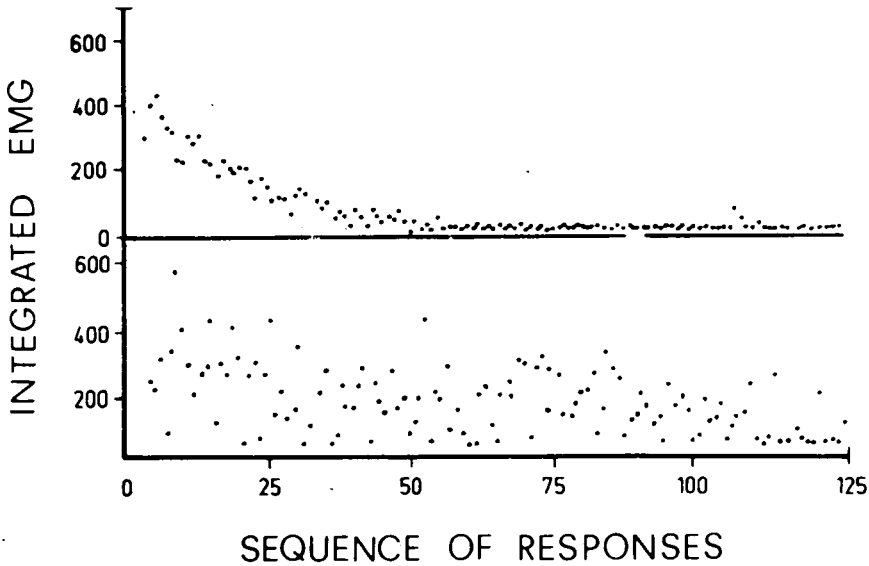


Fig. 13
Dishabituation produced by randomly changed stimulus amplitude (lower series of responses). In the upper series the stimulus amplitude was constant and equaled to the maximum amplitude in the lower series

In a following series of experiments we tried to demonstrate these two functions and estimate their capacities. A difficulty in these studies was the great variability of the responses as well as habituation itself. In order to investigate memory within the system of the flexion-reflex we had to use cross-correlation between stochastically changing stimulus parameters and the size of responses[6]. The cross-correlation coefficient between present responses and preceding stimuli, which ranged from

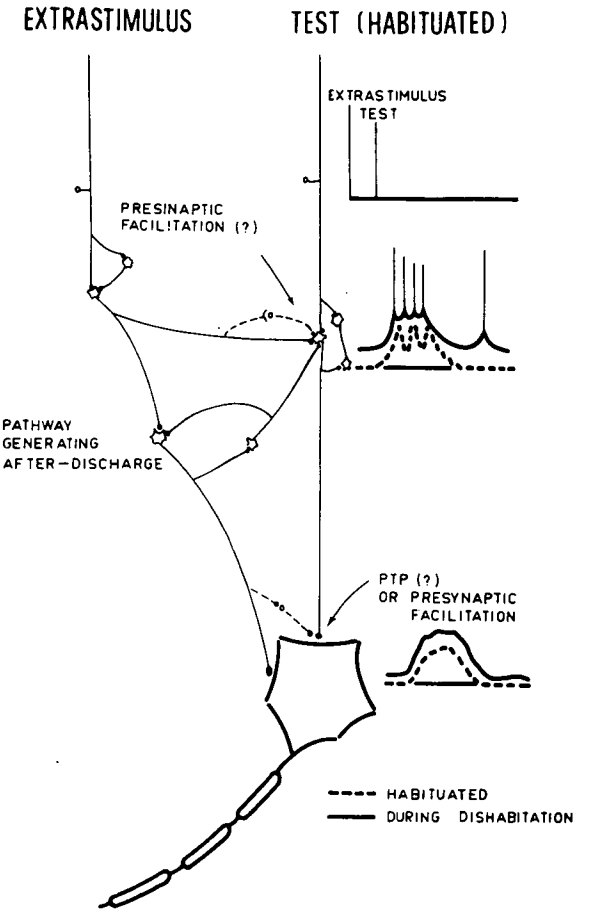


Fig. 14
Postulated mechanism of dishabituation[5]

0.1 to 0.4 (P 0.01), can be regarded as a good evidence of the existence of memory and can also be used as a quantitative estimate (Fig. 15). Because of the rich fluctuation of the size of responses, the testing for the ability of stimulus recognition was even more difficult. An essential requirement was that the comparing patterns of stimulation produced strictly identical afferent volleys. The simplest pair of

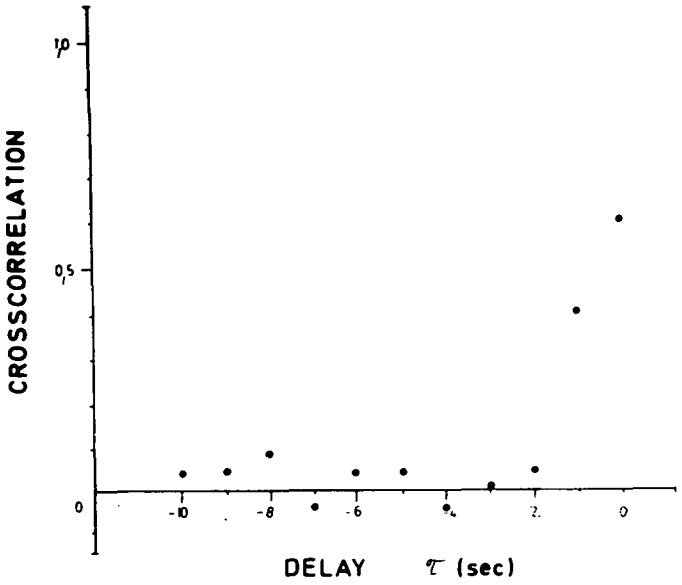


Fig. 15
Cross-correlation between stochastically changing stimuli and the responses

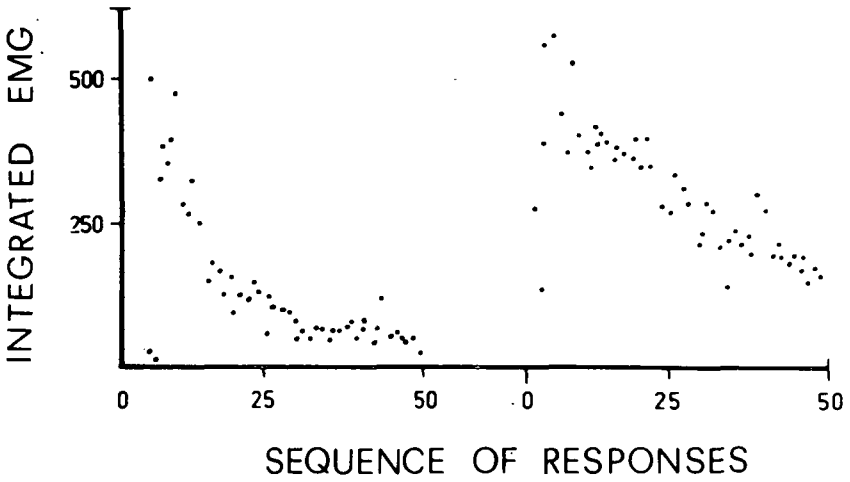


Fig. 16
Stimulation with stimuli of two strengths; shown are only the responses to the weak stimuli. Left: regular alternation of the weak and strong stimulus. Right: random sequence of the weak and strong stimulus

stimulation patterns satisfying this requirement consisted of two stimuli of different strengths, in one pattern alternating regularly while in the other pattern the sequence of the strong and the weak stimulus was random. Fig. 16 shows only the responses to the weak stimuli selected from both series in one of such experiments; habituation is clearly much more expressed in the left series with regular alternation of the strong and the weak stimulus. This finding can be interpreted as a result of recognition of temporal stimulation pattern besed on learning.

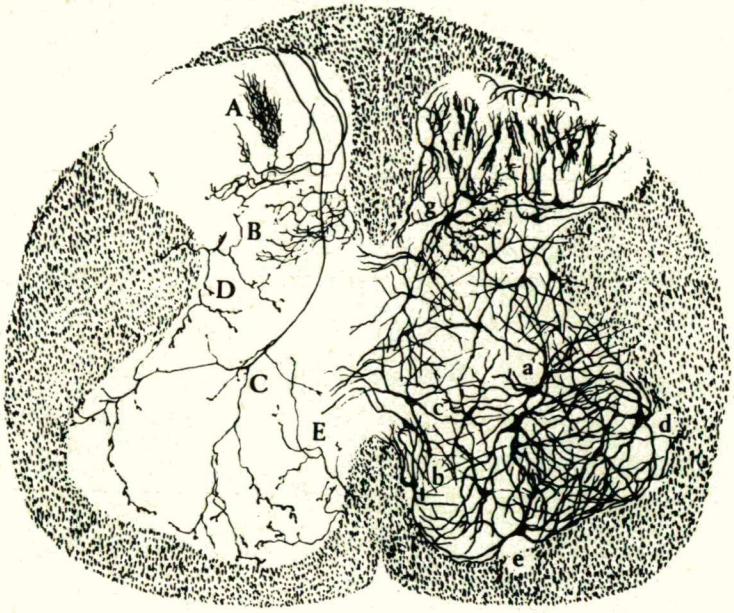


Fig. 17
Cross-section through lumbar spinal cord of the cat. 5% of the interneurones at that level are shown

A number of experiments were designed to see whether it is possible to demonstrate classical conditioning within the system of the flexion reflex in the paraplegic patients. However, statistically significant results indicative of such a possibility were not obtained.

In another series of experiments, the flexion reflex was recorded from single motoneurones with the aid of single fibre electromyography. The purpose of these experiments was to gain more insight in the structure and function of the reflex arc. One of the findings was that the latencies of consecutive responses varied several tense of milliseconds even within the first component, which is 10 to 20 times more than in the case of the monosynaptic *H*-reflex. This implies considerable possibilities for the processing of the afferent information in the way of the well-known neurophysiological processes of spatial and temporal summation as well as facilitation and inhibition. Thus, even though the system of the flexion reflex in terms of a neural network is rather specialized, it nevertheless seems to possess the

capacity for some adaptive decisions. On the other hand, it is simple enough to provide interesting possibilities for the basic neurophysiological and biocybernetic studies of habituation and learning.

One might claim that it is too simple to be capable of such functions. To convince you that it is not so, the last two pictures are shown.

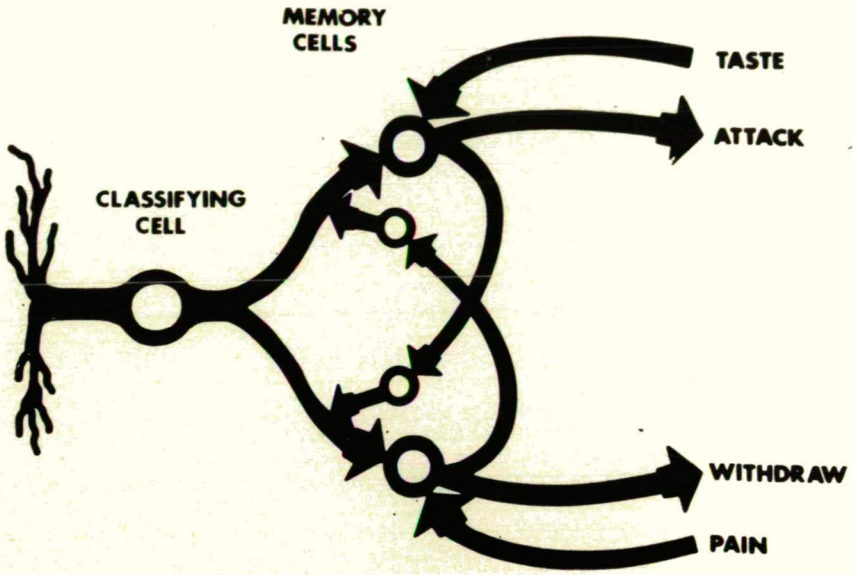


Fig. 18
The "mnemon" of Young. A minimum neural network of octopus capable of learning and adaptive decisions

Fig. 17 shows a cross-section through the spinal cord with the network of interneurones. Only 5% of the total number in that section is shown.

Fig. 18 on the other hand shows the simplest neural network capable of learning and adaptive decisions.

From this comparison it is evident that the structure of our model is rich enough.

Summary

This paper deals with some methods based on statistic theory and pattern recognition, which were used in studies of human reflexes, in the analysis of the electromyogram and in the studies of the single fibre electromyography.

A cybernetic study of the human flexion reflex was performed to see whether elements of learning can be demonstrated to occur in its pathway. Several methods based on stochastically changing input and statistical analysis of input-output relationship were designed. It has been shown that the system of the human flexion reflex is capable of memorizing parameters of stimulation. Experiments were also designed to investigate the ability of this reflex system for adaptive decisions based on learning.

FACULTY OF ELECTRICAL ENGINEERING
UNIVERSITY OF LJUBLJANA
LJUBLJANA, JUGOSLAVIA

References

[1] DIMITRIJEVIĆ, M. R., J. FAGANEL, M. GREGORIČ, P. W. NATHAN, & J. K. TRONTELJ, Habituation: effects of regular and stochastic stimulation, *J. Neurol. Neurosurg. Psychiat.*, v. 35, 1972, pp. 234—242.

[2] CONNEL, J. V., Comparative physiology: Learning in invertebrates, *Ann. Rev. Physiol.*, v. 28, 1966, pp. 107—136.

[3] THOMPSON, R. F. & W. A. SPENCER, Habituation: a model phenomenon for the study of neuronal substrates of behaviour, *Psychol. Rev.*, v. 75, 1966, pp. 16—45.

[4] DIMITRIJEVIĆ, M. R., L. GYERGYEK, & J. K. TRONTELJ, Stochastic stimulation used to study spinal interneuronal system in man, *Progress of Cybernetics*, Gordon & Breach Science Publ., London v. 1, 1970, pp. 353—357.

[5] SPENCER, W. A., R. F. THOMPSON, & D. R. NEILSON, Alternation in responsiveness of ascending and reflex pathways activated by iterated cutaneous afferent volleys, *J. Neurophysiol.*, v. 29, 1966, pp. 240—252.

[6] DIMITRIJEVIĆ, M. R., L. GYERGYEK, & J. K. TRONTELJ, A cybernetic approach to studies of human spinal reflexes. An attempt to demonstrate memory effects. *VIe Congres International de Cybernetique*, Namur 1970. Proceedings in press.

(Received Feb. 14, 1973)

The use of computers in nuclear medicine

By L. CSERNAY

The development of the isotope technique during the past ten to fifteen years has increased our diagnostic possibilities in many fields. Answers are obtained to completely new, previously unapproachable questions, and today the results of the older, more classical procedures can be attained more rapidly, more accurately, and with less inconvenience to the patient.

One of the large groups of routine isotope procedures is comprised of the morphological examination methods.

Liver scintigraphy is most frequently performed; in our laboratory alone there are more than 350 such examinations annually. Scintigraphic projection of the liver can be carried out with either a moving or a stationary detector (with a gamma camera) after the intravenous injection of radiogold, ^{99m}Tc or a ^{113m}In colloid preparation. Using a gamma camera we can also work with labelled Bengal red, which has rapid selection.

Electronic data processing in scintigraphic examinations provides not only quantitative, but also qualitative benefits. For us to be able to understand the refinements of the computer technique, it is first necessary to consider the theoretical and practical problems of the evaluation of scintigraphic examinations and the scintigrams produced.

What are the data which we wish to learn about the examined organ, for instance the liver, when we carry out a scintigraphic examination? The aim is to determine the size of the liver, its form, its position, and the uniformity or non-uniformity of the distribution of the activity. Our other task is to identify in the substance of the liver those circumscribed regions and nodules which differ from the hepatic tissue in their isotope concentration effects; they may enhance, but they usually decrease the normal concentration of the labelled active preparations used in the examination. It is decisive in the identification of regions behaving differently as regards isotope enrichment (tumours, cysts, abscesses), if the occurrence of such areas in the liver can be excluded by the examination with certainty in the case of suspicion.

The first part of the complex task is the simpler. The diagnosis of space-reducing processes in the substance of the liver is appreciably more difficult. Recognition of fist-sized tumours or cysts, or a completely uniform isotope distribution is relatively easy on technically adequate photographs. In perhaps one third of the exa-

minations the opinion of the diagnosing physician may be questionable. In these cases it is usually a matter of smaller nodules, 2—4 cm in diameter, or it may be that the non-uniformity of the isotope enrichment and the anatomical situation lead to the incorrect suggestion of a diagnosis of minor nodules. The expectations from scintigraphy are fulfilled if it provides suitably reliable accuracy in the questionable cases, in the diagnosis or exclusion of minor lesions. The minimum size of the cold region which can be detected in the ideal case is determined by the physical parameters of the instrument. The detection of nodules larger than this is possible in principle, but in practice remains uncertain up to a much larger limit, above which the diagnosis is no longer problematic. The aim of the examinations is to diagnose the theoretically identifiable nodules *in practice* too.

What are the factors which present obstacles to the attainment of this?

The statistical, non-uniform nature of the radiation, the distortion due to the collimator used, and the limitations of the resolution of the display system are all objective factors. The subjective evaluation of an obtained scintigram, however, means a further loss of information, or may even provide faulty information. The formulation of the findings influences their interpretation. It is not difficult to see that given acquired information can be interpreted in different ways by the clinician, merely because of the subjectivity of the formulation of the report. The potential of liver scintigraphy is deteriorated by the listed problems in just those cases where the lesions are small, and where a definite attitude affects the development of the future of the patient decisively. Electronic data processing can be used to combat all of the negative factors listed here.

The statistical nature of radioactive radiation is a physical regularity; its effect can not be eliminated, but can be substantially moderated by means of mathematical smoothing procedures.

On the application of a smoothing procedure, the statistical fluctuation regularly appearing on the scintigraphic picture is clearly reduced, the picture becomes more homogeneous, and the evaluability is improved. Attention must be drawn, however, to the fact that to a certain extent the smoothing procedures deteriorate the detection of areas differing in activities from those of their environments. (e.g. cold nodules), since the decreased activity values at the circumscribed position are changed in the direction of the higher level of the environment, and are smoothed. This undesirable effect occurs particularly on the use of a smoothing configuration with low centre-point weight, taking into account many adjacent elements. In spite of this, the use of smoothing procedures with appropriate discrimination has led to a significant advance in the evaluation of scintigrams. It is virtually inconceivable to carry out the method manually, since the processing of each scintigram would last 8—10 hours. For a medium-size computer the task, together with the new picture formation, requires 4—5 seconds machine time. The now basic step of electronic data processing of scintigrams is the smoothing of the data.

The second important problem of the scintigraphic procedure is that even the relatively most perfect, multichannel, focussed collimators are able to project the examined organ only with distortion.

If the two regions of different activities and also the activity-free detail between them, are projected by scintigraph, a picture differing from reality, or from the object-function, the distorted picture-function is obtained, which indicates activity even in the really activity-free regions, because the detector above these points also

observes radiation from the active regions on the two sides. Projection of the picture-function and object-function onto each other, confirms that the obtained information differs from reality. The distorting effect of the side scatter can not be neglected in the examination of the liver, particularly in the case of the smaller nodules, their identification is significantly affected, and hence the diagnostic potential of the procedure is deteriorated.

The common aim of the focussing procedures used in electronic data processing is that the distorting effect of the collimator be reduced on the picture, and if possible eliminated. Various mathematical procedures can be employed, but they have in common that they take into account the extent of distortion of the collimator actually used in the examinations, and its viewing to the side. The iteration procedure, the residual tabular procedure, and the Norbert Wiener filter, operating with concrete physical values of the collimators, approximate the distorted picture-function to the real object-function. Among others, the difference is for example that the iteration procedure, repeated one after the other, improves the picture-function stepwise, while the residual tabular method and the Wiener filter, or inverse filter, approximate in one step. At any event, the use of the any of the focussing procedures results in a substantially improved picture, which is much closer to the projected object. The improvement of the picture permits the identification of smaller nodules in daily practice, while more realistic and certain decisions can be reached using the data of a given photograph. The focussing procedures and other mathematical methods can naturally not change the size of nodule theoretically detectable with a given collimator, nor do they allow the obtaining of information for which the detector-collimator system performing the charting is incapable. The size of nodule identifiable in practice is approximated to the theoretical value, and the distorted information is improved.

The different focussing procedures are generally carried out after one of the smoothing procedures discussed earlier, because the focussing methods are very sensitive to statistical fluctuations. In direct daily practice, the concrete question arises of what type of focussing procedure is worth using in the processing of liver scintigrams. The answer is decided by the consideration of two factors. For practice, the most optimum procedure is that which results in the greatest improvement, while at the same time the machine time devoted to the data processing is the least. Clearly we are forced to compromise, taking into account that computer time is fairly expensive all over the world. Naturally, if a private computer is available for use, this latter factor affects the choice to a lesser extent.

The third problem of the scintigraphic method is the restricted resolution of the recording system. The recording on the customary industrial scintigraphs is performed in general with a line-striking or number indicating system and with a photo-technique. It would be optimum in theory if the recording system were to resolve the difference between the most active and the radiation-free regions into as many sections as possible, and, taking 1 % recording divisions, were to divide the data mass into 100 values for example. In this case the small changes too would be identifiable. In practice, however, it is almost inconceivable that a picture consisting of 2000—4000 elemental points would be easy to survey and evaluate after being divided into 100 parts. The classification of the values into fewer classes clearly deteriorates the resolution. The recording system of the commercial scintigraphs exhibits a compromise solution; information is lost by the use of fewer

value divisions, but at the same time diagnosis is made simpler by the increase of the clarity. The recording system of the scintigraph and the number of divisions can not be changed from case to case, and so only one or two concrete possibilities can be employed in the evaluation.

Here too electronic data processing results in an improvement in quality. With the use of 1—2 seconds machine time the data mass of the scintigraphic picture can be classified into very many different resolution groups; apart from one or two-digit numbers, various types of lettering, positive signs and empty spaces can be used for

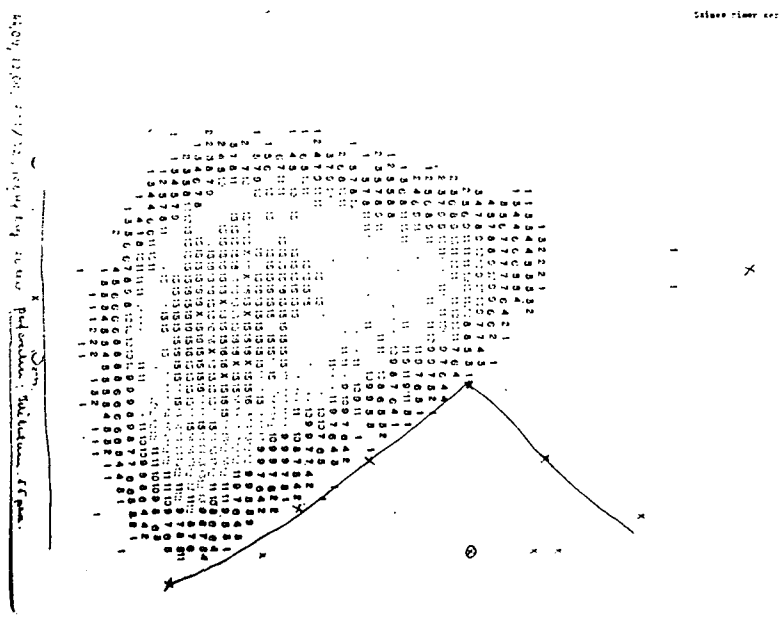


Fig. 1
Original picture—normal liver scan

the recording; isointensity lines can be calculated, and these can be displayed in a coloured picture, on a television screen. Despite the large variety of display possibilities, the conflict between the number of divisions and the clarity can not be solved, but merely decreased. Accordingly, there is a need for a completely different solution. A classification into groups ensuring better resolution can be achieved in the given region by accentuating the data of a part-region which appears suspect on consideration of the original photograph. Thus, the number of divisions for this region will be greater than the corresponding number when the complete picture is divided up. If 20% is subtracted from the values of the accentuated part-region in each of four steps, one after another, and a new classification is carried out, keeping the number of divisions after each subtraction, the resolution is continually increased without a deterioration in the clarity of the picture.

In order to demonstrate what has been said, Fig. 1 shows the original picture and also recording types achieved by means of computer.

The number of divisions on the original picture prepared with the scintigraph is 16. The identification is made more difficult by a logarithmic relation between the size of the numbers and the activity values.

The picture prepared with a fast print-out provides more realistic information, with a similar number of divisions, but with a linear relation (see Fig. 2).

The calculation and plotting of the isointensity lines provides a satisfactory pictorial representation even with the use of fewer divisions, and the number of levels may be increased or decreased. Without a computer, level recording can not be solved, of course.

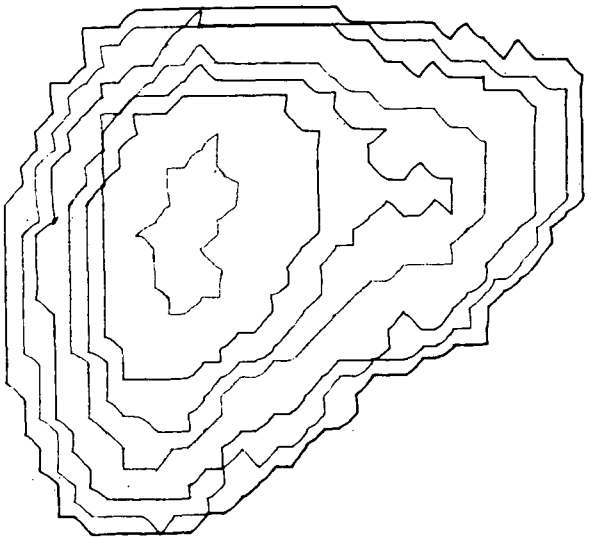


Fig. 2a
Isointensity lines picture—normal liver scan

The computer recording of the processed liver scintigram, of various forms and numbers of divisions, requires about 8—10 seconds per form. The analysis of problematic scintigrams with various recording forms needs 30—40 seconds, and finally, on the basis of the picture obtained with the most optimum recording as regards the task, an answer can be given to the question of the physician requesting the examination.

The fourth problem of the scintigraphic method is the subjective evaluation of the scintigrams. It can readily be seen that, depending on their specialist knowledge and their familiarity with the evaluation, different physicians may interpret a given scintigram in different ways, particularly if the picture is a limiting case or the changes are discrete. In addition, the interpretation of the picture is affected by other factors too, such as the time available for the evaluation, and the mood of the evaluating physician.

Let us look at this problem a little closer. The liver scintigram is a two-dimensional projection of the activity-concentrating hepatic tissue. Independently of the position and form of the liver, an opinion must be given in the evaluation as to the extent

of the liver projection too. The liver may have at least 8—10 types of normal form variants, and within the given form the estimation of the size is a subjective judgement based on experience of the pictures, without measurement. Planimetry of the picture is tiring, and in my experience is not done by anyone. In the evaluation it would be good to know the dimensions of the given liver projection is square centimetres, and to be able to classify it into a certain group according to this numerical value.

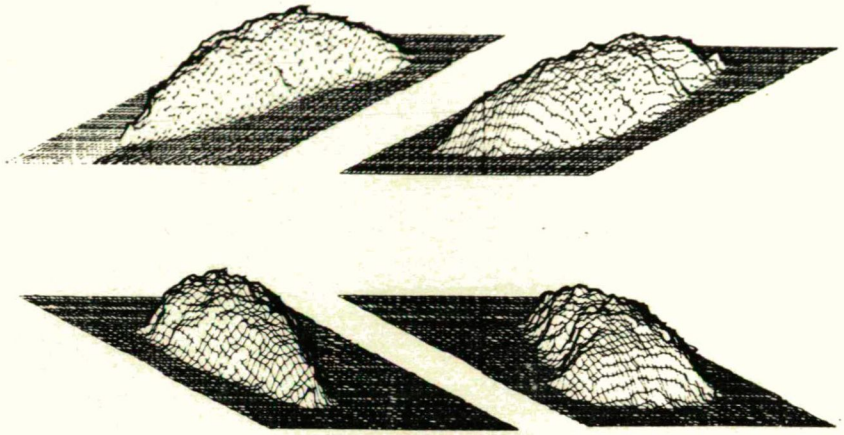


Fig. 2b
Three dimensional picture—normal liver scan

For the machine the calculation of the area of the liver projection in square centimetres is a task requiring only a second or so, and during this time it is also able to calculate the diameters of the liver projection in various directions. The knowledge of the area of the projection permits a more exact estimation of the size of the liver.

Identification of the form of the liver projection can also be performed by machine, by means of the form identification algorithm. Such a type of programme stores the possible normal variants in the memory of the computer, and compares the form of an examined liver scintigram with those of the stored normal variants in a previously determined order. If a difference from the stored forms of ± 5—10% is allowed, then the machine either finds a suitable, similar form during the programme, or there is no such form among those stored. In the former case the machine identifies the examined picture by naming the similar normal form, while in the latter case it records that the form of the examined liver can not be identified with those of the stored variants. This means that the programme has found a new form type, or the form of the examined liver is pathological.

It would be possible to list further partial problems in the objective machine evaluation, and to their possible solution forms. Instead of this, we shall outline in generality that the possibility of complete automation of the evaluation of liver scintigrams depends on whether the necessary tasks can be composed in the language of the mathematical methods and logic. In so far as the corresponding algorithm is found for the solution of every task in this case the automatic, objective evaluation

of the pictures can be achieved. Considerable efforts are being made in a number of laboratories all over the world to solve the problem of automatic picture evaluation. The partial results to date promise that it has proven possible to raise the subjective evaluation, which is one of the greatest problems in the scintigraphic procedure, to a mechanical level, so that it may be made *objective*. In contrast with the previous problems, however, this is still a task for the future, and further efforts are required for its solution.

The question can already be raised, however, of what level of evaluation can be expected from the mechanical evaluation programme. If the facts are analyzed realistically, it may be thought that the successfully solved picture-evaluation programme could give an opinion on a very good level for about 90% of the liver scintigrams. It will presumably not attain the level of the best evaluations, but it will certainly exceed those of anyone possessing insufficient practice and experience, and it will even provide assistance to the most expert too. The data calculated with the programme will be of help even to the most routine physician, and will draw attention to lesions which would possibly not have been identified at all, or only on a given day, depending on the level of the efforts or the mood of the evaluator. The mechanical evaluation programme means the peak of the computer technique in the solution of the problems of scintigraphy.

After this outline of the future perspectives, let us finally consider a fifth problem of scintigraphy, which is related to the others but differs from them in its essence and in its solution. Independently of the subjective picture evaluation, the formulation of the information content from the pictures, and the drawing of findings, are likewise subjective activities. With great probability the formulated opinions of two evaluating physicians derived from a given scintigram are essentially completely the same. To exaggerate the problem, it may be said that in a given case the written text may be at least as characteristic of the evaluator as of the lesion. Since the physician indicating the examination is presumably unfamiliar with the evaluation of scintigraphy, even when he is in possession of a copy of the picture he will primarily rely on the formulated opinion when he considers the result as an important or a less important contributing datum in his conscious activity to form a diagnosis.

Very considerable help is provided to overcome the labyrinths of subjective formulation by the application of the computer. In the possible changes in the size of the liver, its form, its position, its capacity to concentrate activity, and the description of the more active or colder regions outside or within the substance are formulated precisely and in an unambiguous way, and if coding symbols are ascribed to the formulated sentences or phrases, then we can construct a mathematical programme which leads, after the feeding in of the coding symbols characteristic of the examined liver, to a continuous text output and finally to a printed finding in the form of a medical finding.

What is the advantage of an opinion formulated with a coding symbol set?

1. The formulation of the finding is clear and unambiguous, and although its style is dry it is free of subjective comparison.

2. The physician performing the evaluation with the coding system is compelled to record every change.

3. It frees the staff of the laboratory from the typing work associated with the findings. The preparation with the fast printer of the computer of the complete

finding, on which the most important data regarding the patient and the employed procedure are given, requires 15—25 seconds.

4. The results of scintigraphy are stored on magnetic tape, which needs little space. There is a possibility at a later date, therefore, for a change characterized by a given coding symbol to be extracted from the complete store of liver scintigraphic results, or for the correlation of a lesion with a parameter defined by any other symbol to be analyzed statistically.

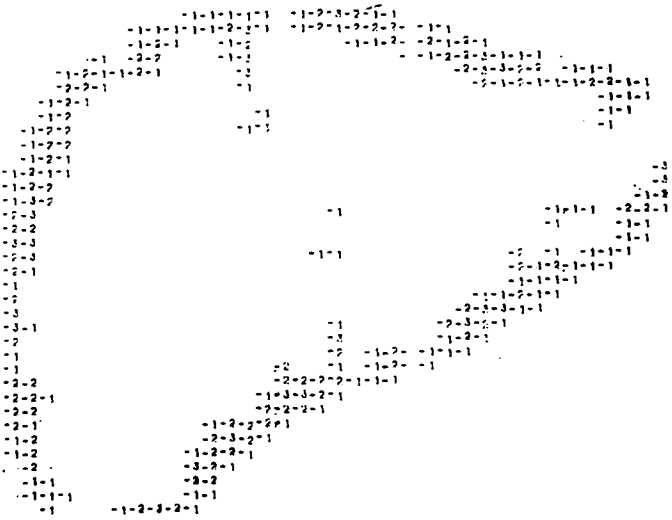


Fig. 3
Residual tabular picture—normal liver scan

After this theoretical and partially practical survey of the computer solution of liver scintigraphic problems and tasks, I should like to show very briefly how computer data processing has materialized in our laboratory.

The first important task in the solution of electronic data processing is the recording of the experimental data in a manner understandable for the computer.

In the outline of the principle of our present system it can be seen that at the same time as the measurement the experimental data of the scintigraph and the automatic sample changer are automatically recorded on paper tape with a perforator. The data measured at other sites and the data identifying and characterizing the patient and the examination procedure are recorded on paper tape on a Siemens telex machine installed in the laboratory, by female assistants specially trained for such perforation work. The data are recorded on the paper tape in the European telex code, and are processed daily in the off-line manner, i.e. separated in time from the measurements, by a Minsk-22 computer in the Cybernetics Laboratory of József Attila University. The computer possesses an 8 K word internal memory, fast data input and perforating unit, and a 6 magnetic tape storage unit. The computer uses a 128-position printer for the reproduction of the calculated results, findings and pictures. An incremental plotter is used to draw out the iso-

intensity lines. Our routine programmes are stored on magnetic tape, and by means of a controlling operational system developed in accordance with the requirements of nuclear medicine, they can be summoned and activated by keying in to the name of the programme from teletype connected to the machine. Magnetic tape is also used for the recording of the results and for the storage. In addition to the off-line processing method, separated in time, we shall be able to achieve simultaneous, on-line data processing too, by means of a direct data transfer line built between the laboratory and the computer.

Apart from the system controlling the data processing, 34 computer programmes have been developed since 1969 in collaboration with my mathematical colleague János Csirik. One third of the programmes ensure the read-in of the data and the display of the results in various forms, while the other, problem-oriented procedures perform the data processing of various concrete isotope diagnostic methods. The residual tabular procedure was developed in 1970 by János Csirik. The method takes into account the distorting effect of the collimator used, and very sensitively identifies regions of decreased or enhanced activity within the substance of the liver. The recording form is also of advantage for the demonstration of cold or hot nodules, because apart from the contour of the organ, symbol groups (in our case numbers with negative or positive signs) are printed out only where a nodule with lower or higher activity than that of the environment is situated. According to the results of the phantom examinations, the method is also suitable for the detection of nodules 2.5 cm in diameter under conditions for which a nodule 3.5 cm in diameter can only be uncertainly surmised on the original image.

I shall now give an account of the results of our previous work. In almost one third of the patients examined by liver scintigraph, neglecting differences in shape and size, clearly negative pictures are obtained as regards the demonstration of cold nodules. In 20% of our cases cold nodules can be identified even in the primary picture; with regard to the resolution of the scintigraph used, these are at least 5—6 cm in diameter. In these cases, therefore, the identification presents no problems. In almost half of the examinations, however, a definite stand can not be taken as to the possibility of occurrence of a space reducing process when the primary picture is inspected. After computer data processing, on the other hand, nodules could be identified on the substance of the liver, or this possibility could be excluded with certainty, in four fifths of these patients. In the remaining one fifth of the doubtful cases the doubt could not be resolved by means of the computer methods. Overall, therefore, it can be said that the accuracy of identification is substantially increased by the electronic data processing, and there is a corresponding decrease in the number of pictures for which a definite opinion can not be given after further careful study.

(Received Feb. 14, 1973)

Couverture irrédondante des fonctions booléennes définies par leurs monômes vrais et faux — fonctions simultanées

J. C. TORGUE,* K. PÁSZTOR,** P. AZEMA*

I-INTRODUCTION

Dans un article récent, E. MORREALE [4] a défini un algorithme pour la recherche d'une base irrédondante des fonctions logiques[1] définies par les points de leur borne supérieure. Dans un premier temps, cette méthode a été modifiée [6] pour traiter les fonctions très incomplètement spécifiées presque toujours définies par l'ensemble des points de leur borne supérieure et du complément de leur borne supérieure.

Dans les problèmes concrets il arrive très souvent que toutes les combinaisons des variables ne soient pas employées pour définir l'état de la sortie, c'est pourquoi la première-partie de cet article présente un algorithme déduit des deux précédents, s'appliquant au cas des fonctions données par un ensemble de monômes qui couvrent les points vrais et de monômes qui couvrent les points faux.

La deuxième partie aborde le problème de la couverture irrédondante des fonctions simultanées.[2] Ce problème, très complexe, a été étudié dans de nombreux articles. Certaines méthodes [1], Tag Method [2], Consensus [5] exigent le calcul de tous les implicants premiers et conduisent donc ou a la résolution de table de choix, ou à la recherche de couverture irrédondante par des moyens appropriés [3], [8]. L'approche présentée, basée sur les travaux de [7] et [4], permet de considérer seulement les implicants premiers utiles à la génération d'une base irrédondante.

II-FONCTIONS DEFINIES PAR LEURS MONOMES VRAIS ET FAUX

Soit $F(x_1, ..., x_i, ..., x_n)$ une fonction booléenne de n variables définies par l'ensemble P de ses monômes où elle prend la valeur 1 et par l'ensemble Q de ceux où elle prend la valeur 0. Nous allons chercher tour à tour si, pour chaque $i \in \{1, n\}$, il existe des implicants premiers appartenant à la base irrédondante et commençant par les variables x_i ou $\bar{x}_i$ qui seront notées x_i^α, $\alpha \in \{0, 1\}$ avec $x_i^0 = \bar{x}_i$ (valeur logique 0) et $x_i^1 = x_i$ (valeur logique 1). Pour cela, il est nécessaire d'introduire les trois opérateurs suivants.

[1] switching functions
[2] multiple output switching functions

Définitions

Définition 1. *Réduction.* La réduction notée $R_{x_i}\left(\frac{P}{Q}\right)$ efface dans P/Q la colonne correspondant à la variable x_i

$$R_{x_1}\left|\begin{array}{cccc|c} 0 & - & 1 & 0 & - & 1 & 0 \\ - & - & 0 & 0 & - & 0 & 0 \\ \hline 1 & 1 & 1 & 0 & 1 & 1 & 0 \\ 1 & - & 1 & 1 & - & 1 & 1 \\ 0 & 1 & 0 & - & 1 & 0 & - \end{array}\right| = \begin{array}{ccc} - & 1 & 0 \\ - & 0 & 0 \\ \hline 1 & 1 & 0 \\ - & 1 & 1 \\ 1 & 0 & - \end{array}.$$

Définition 2. *Intersection.* L'intersection notée $I_{x_i^\alpha}\left(\frac{P}{Q}\right)$ consiste à retenir dans le tableau P les lignes où $x_i=\alpha$ et à enlever de Q celles où $x_i=\bar{\alpha}$ puis à appliquer R_{x_i} à ce qu'il reste de P/Q

$$I_{x_3}\left|\begin{array}{cccc|c} 0 & 0 & - & 1 & 0 & - & 0 \\ 0 & - & 1 & 0 & & & \\ - & - & 0 & 0 & & & \\ \hline 1 & 1 & 1 & 0 & 1 & 1 & 0 \\ 1 & - & 1 & 1 & 1 & - & 1 \\ 0 & 1 & - & - & 0 & 1 & - \end{array}\right| = \begin{array}{ccc} & & \\ & & \\ & & \\ \hline 1 & 1 & 0 \\ 1 & - & 1 \\ 0 & 1 & - \end{array}.$$

Définition 3. *Couverture.* La couverture $C(P/Q)$ efface de P les lignes includes dans l'implicant premier trouvé et extraie des monômes les points restant à couvrir.

$$\frac{P}{Q} = \begin{array}{cccc} - & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ \hline 1 & 0 & - & 1 \\ 0 & 0 & 0 & - \end{array}$$

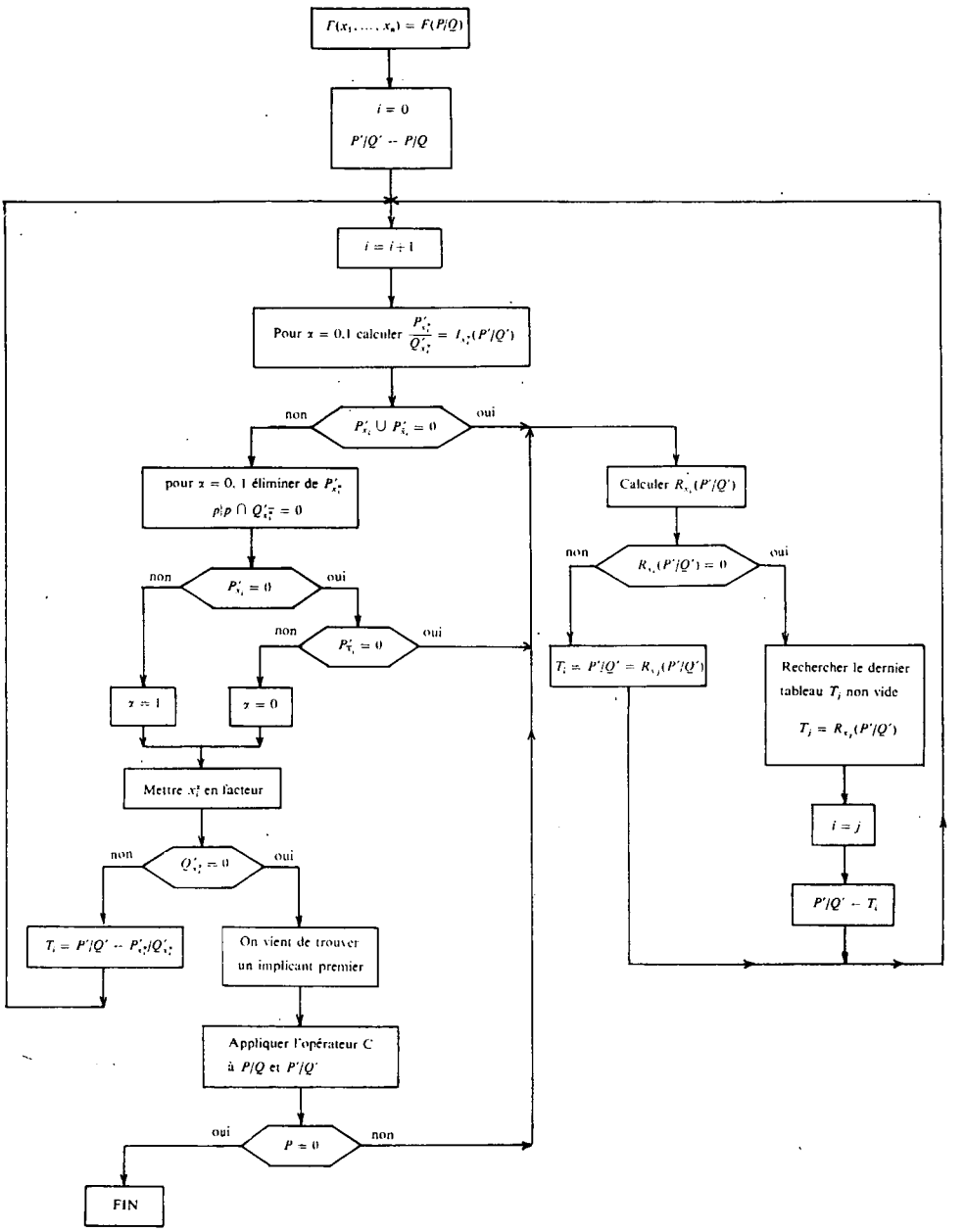
avec l'implicant premier $\bar{x}_1 x_3$ on a

$$C\left(\frac{P}{Q}\right) = \begin{array}{cccc} 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 \\ \hline 1 & 0 & - & 1 \\ 0 & 0 & 0 & - \end{array}.$$

Proposition 1. Considérons $I_{x_i^\alpha}\left(\frac{P}{Q}\right) = \frac{P_{x_i^\alpha}}{Q_{x_i^\alpha}}$ pour $\alpha\in\{0,1\}$; Si $P_{x_i^\alpha} \cap Q_{x_i^{\bar{\alpha}}} = \emptyset$ alors les points de $P_{x_i^\alpha}$ peuvent être couverts par un implicant premier où la variable x_i^α n'apparaît pas.

En effet, pour tout élément p de $P_{x_i^\alpha}$, $P_{x_i^\alpha} \cap Q_{x_i^{\bar{\alpha}}} = \emptyset$ signifie qu'aucun point de p n'appertient à $Q_{x_i^\alpha}$ donc que $x_i^{\bar{\alpha}}\cdot p + x_i^\alpha \cdot p$ appartient à la borne supérieure. Donc que p peut être couvert par un implicant premier où la variable x_i^α n'apparaît pas.

Organigramme



Proposition 2. A la fonction $F\left(\frac{P}{Q}\right)$ appliquons l'opérateur $I_{x_i^\alpha}$; Si $I_{x_i^\alpha}\left(\frac{P}{Q}\right) = \frac{P_{x_i^\alpha}}{Q_{x_i^\alpha}}$ est tel que $Q_{x_i^\alpha} = \emptyset$, alors x_i^α est implicant premier de F.

x_i^α est implicant premier car $Q_{x_i^\alpha} = \emptyset$ veut dire qu'aucun point de la fonction où x_i^α apparaît n'appartient pas à Q. Cet implicant est premier car Q étant non vide et la fonction F n'est pas la fonction identité.

Algorithme

1. Soit $F(x_1, \ldots, x_n) = P/Q$, $i = 0$; $T_i = P'/Q' = P/Q$, $i = 1$.

2. $i = i+1$;

a) Appliquer $I_{x_i^\alpha}$, $\alpha = \{0, 1\}$ à P'/Q';

b) Si $P'_{xi} \cup P'_{\bar{x}_i} = \emptyset$ aller en 7.

3. Pour $\alpha = \{0, 1\}$ vérifier s'il existe un implicant premier de la couverture devant contenir x_i^α. Enlever des tableaux $P'_{x_i^\alpha}$ les termes ayant une intersection vide avec $Q'_{x_i^\alpha}$.

4. *a)* Si $P'_{xi} \neq \emptyset$ poser $\alpha = 1$ et aller en 6;

b) Si $P'_{xi} \neq \emptyset$ poser $\alpha = 0$ et aller en 6.

5. Appliquer R_{x_i} à P'/Q'. Si le tableau obtenu est vide rechercher le dernier tableau non vide des étapes précédentes, soit T_j, faire $i=j$ et le mettre dans $R_{x_i}(P'/Q')$. Transférer $R_{x_i}(P'/Q')$ dans P'/Q', incrémenter i et aller en 2. Si non, mettre $R_{x_i}(P'/Q')$ en $P'/Q' = T_i$, aller en 2.

6. *a)* Mettre x_i^α en facteur;

b) Si $Q'_{x_i^\alpha} \neq \emptyset$ mettre $I_{x_i^\alpha}(P'/Q')$ dans $P'/Q' = T_i$, aller en 2.

Si $Q'_{x_i^\alpha} = \emptyset$ le terme facteur de $I_{x_i}(P'/Q')$ est un implicant premier de la base irrédondante. Appliquer l'opérateur Couverture à P/Q et P'/Q'. Si P n'est pas vide aller en 5.

7. FIN

III-FONCTIONS SIMULTANEES DEFINIES PAR L' ENSEMBLE DE LEURS POINTS VRAIS ET FAUX

La méthode exposée au paragraphe précédent peut être généralisée au cas des fonctions simultanées.

Notations

Soit $F=\{\varphi_1, \varphi_2, \ldots, \varphi_m\}$ m fonctions simultanées où chaque fonction φ_i $i=\{1,m\}$ dépend de n variables $x_1, x_2, \ldots, x_n$. Chaque φ_i est définie par l'ensemble de ses points vrais P_i et celui de ses points faux Q_i. L'ensemble des points vrais de F sera alors $P=\bigcup_{i=1}^{m} P_i$, celui de ses points faux $Q=\bigcup_{i=1}^{m} Q_i$. A chaque point de P (respectivement Q) on associe une étiquette constituée de m bits dont le $i^{\text{ième}}$ prendra pour valeur 1 ou 0 suivant que le point appartient ou n'appartient pas à P_i (respectivement Q_i).

Par exemple: $F = \{\varphi_1, \varphi_2\}$ où

$$\varphi_1 = x_1x_2x_3 + \bar{x}_1\bar{x}_2x_3 \multimap P_1 = \begin{matrix} 1 & 1 & 1 \\ 0 & 0 & 1 \end{matrix}$$

$$\varphi_2 = x_1x_2x_3 + x_1x_2\bar{x}_3 \multimap P_2 = \begin{matrix} 1 & 1 & 1 \\ 1 & 1 & 0 \end{matrix}$$

donnent

$$P = \begin{vmatrix} 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 \\ 1 & 1 & 0 & 0 & 1 \end{vmatrix}.$$

Définitions

Les opérateurs intersection $I_{x_i^{\alpha}}$ et réduction R_{x_i} gardent la même définition qu'au paragraphe II-1. L'opérateur couverture est modifié de la manière suivante.

Définition 4. *Couverture.* Dans l'ensemble des points couverts par l'implicant premier mettre à 0 les composantes des étiquettes correspondant aux fonctions effectivement couvertes. Si toutes les composantes d'une étiquette sont nulles, effacer de P la ligne correspondante.

Remarque. Ceci va nous conduire à l'utilisation de deux étiquettes différentes pour les points de P: l'étiquette initiale et une étiquette dite de travail, déduite de la première après application de l'opérateur couverture.

Nous noterons „Etic" et „Etict" les étiquettes originales et de travail correspondantes.

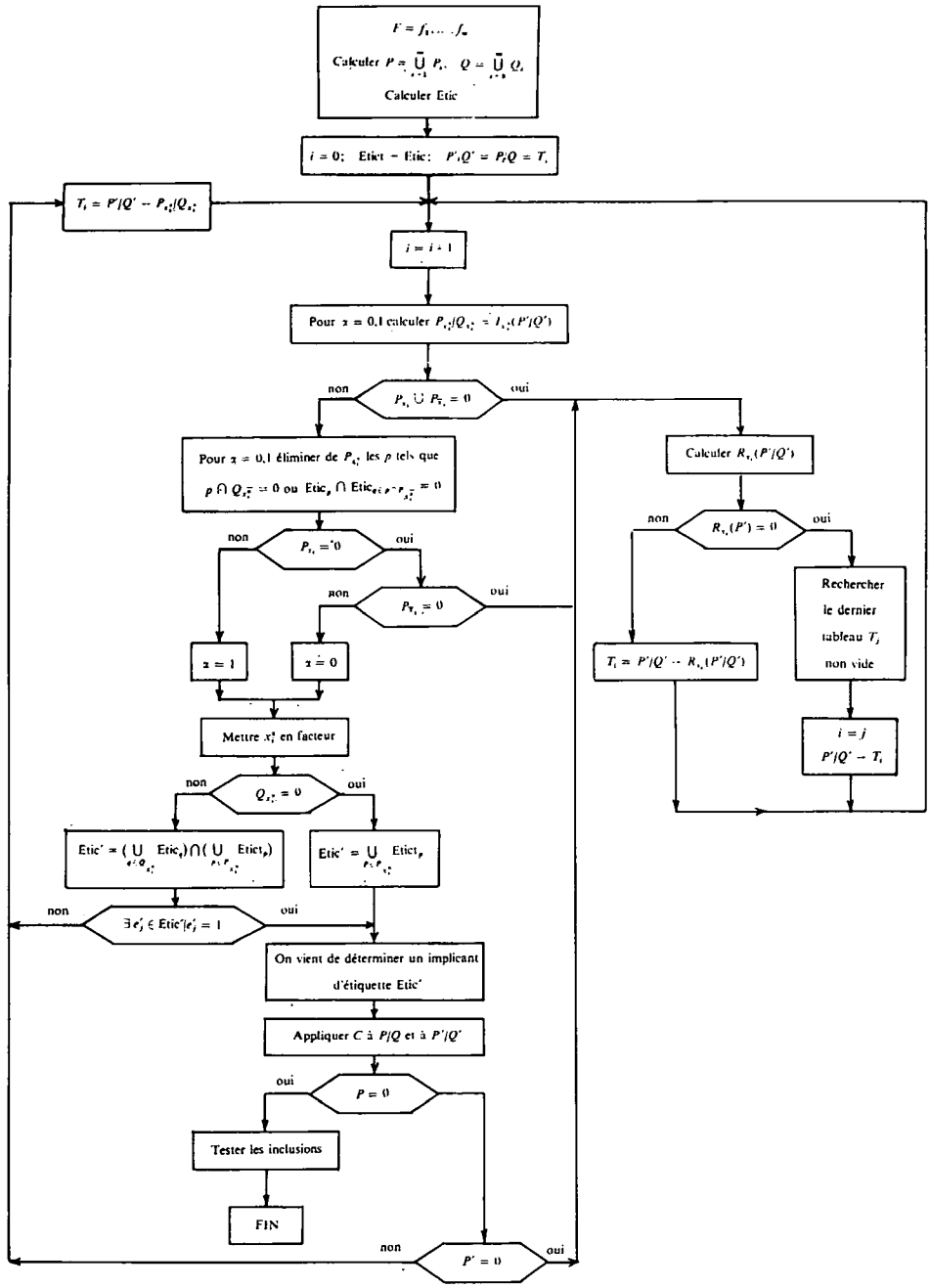
Nous allons définir sur ces étiquettes trois règles de calcul.

Définition 5. *Opération Union.* $\text{Etic} = \bigcup_{i=1}^{k} \text{Etic}_i$ où chaque composante de Etic est obtenue par l'union des composantes correspondantes de Etic_i.

Définition 6. *Opération ET.* $\text{Etic} = \bigcap_{i=1}^{k} \text{Etic}_i$ où chaque composante de Etic est obtenue en faisant l'intersection des composantes de même rang de Etic_i.

Définition 7. *Opérateur Négation.* $\overline{\text{Etic}}$ est obtenue par la complémentation de chaque composante de Etic.

Organigramme



Proposition 3. A la fonction $F(P/Q)$ appliquons l'opérateur $I_{x_i^\alpha}$. Si $I_{x_i^\alpha}(P/Q) = \frac{P_{x_i^\alpha}}{Q_{x_i^\alpha}}$ est tel que $Q_{x_i^\alpha} = \emptyset$ ou tel qu'il existe au moins une composante φ_j pour laquelle $Q_{x_i^\alpha} = \emptyset$, alors x_i^α est un implicant premier.

La démonstration est analogue à celle de la proposition 2.

Remarque. Pour reconnaitre les composantes φ_j pour lesquelles Q_{x_i} est vide, il faut calculer l'expression

$$\text{Etic} = (\overline{\bigcup_{q \in Q} \text{Etic}_q}) \bigcap (\bigcup_{p \in P} \text{Etict}_p).$$

Les composantes non nulles de Etic correspondent aux φ_j. Etic est également l'étiquette de l'implicant premier obtenu.

Proposition 4. Considérons $I_{x_i^\alpha}(P/Q) = \frac{P_{x_i^\alpha}}{Q_{x_i^\alpha}}$ pour $\alpha = \{0, 1\}$. Soit $A = P_{x_i^\alpha} \bigcap Q_{x_i^{\bar{\alpha}}}$; Si $A = \emptyset$ ou si pour chaque élément de A les étiquettes qui lui correspondent dans $P_{x_i^\alpha}$ et $Q_{x_i^{\bar{\alpha}}}$ sont disjointes, alors les points de $P_{x_i^\alpha}$ peuvent être couverts par un implicant premier où la variable x_i^α n'apparaît pas.

En effet pour un élément de A dire que les étiquettes qui lui correspondent dans $P_{x_i^\alpha}$ et $Q_{x_i^{\bar{\alpha}}}$ sont disjointes c'est dire que pour chaque fonction $\varphi_j, j = \{1, m\}$ la $j^{\text{ième}}$ composante de l'intersection des étiquettes est 0

— soit dans $P_{x_i^\alpha}$ ce en quoi le point A n'appartient pas à la fonction φ_j;

— soit dans $Q_{x_i^{\bar{\alpha}}}$ qui est alors vide pour φ_j. Et on est ainsi ramené à la démonstration de la propostion 1.

Algorithme

1. Pour $F = \{\varphi_1, \ldots, \varphi_m\}$ former les tableaux $P = \bigcup_{i=1}^{m} P_i$ et $Q = \bigcup_{i=1}^{m} Q_i$. Calculer les étiquettes Etic et Etict, $i = 0$, $T_i = P'/Q' = P/Q$.

2. $i = i + 1$; Pour $\alpha = \{0, 1\}$ appliquer $I_{x_i^\alpha}$. Si P'_{x_i} et $P'_{\bar{x}_i}$ sont vides aller en 5.

3. Pour $\alpha = \{0,1\}$ enlever de $P'_{x_i^\alpha}$ les termes n'appartenant pas à $Q'_{x_i^{\bar{\alpha}}}$ ou y apparaissant avec une étiquette disjointe.

4. *a*) Si $P'_{x_i} \neq \emptyset$ poser $\alpha = 1$ et aller en 6;

b) Si $P'_{\bar{x}_i} \neq \emptyset$ poser $\alpha = 0$ et aller en 6.

5. Appliquer R_{x_i} à P'/Q'. Si le tableau obtenu est vide mettre le dernier tableau T_j non vide dans R_{x_i} (P'/Q') et faire $i = j$. Transférer $R_{x_i}(P'/Q')$ dans P'/Q', incrémenter i et aller en 2. Si, non mettre $R_{x_i}(P'/Q')$ en $P'/Q' = T_i$ et aller en 2.

6. Mettre x_i^α en facteur.

7. *a)* Si un implicant premier a été généré le mettre dans la base irrédondante avec son étiquette. Appliquer C à P/Q et P'/Q'. Si P est vide aller en 8; Si P' est vide aller en 5.

b) Mettre $I_{x_i^\alpha}(P'/Q')$ dans $P'/Q'=T_i$, incrémenter i et aller en 2.

8. Tester pour chaque φ_j $(j=1, m)$ les redondances possibles par inclusion.

9. FIN

Exemple

Considérons la fonction $F=\{\varphi_1, \varphi_2, \varphi_3\}$ des variables $x_1 \ldots x_4$ définie par

$$\frac{P}{Q}=\begin{array}{cccc|ccc|ccc} & & & & & \text{Etic} & & & \text{Etict} & \\ 1&0&1&0 & 0&1&1 & 0&1&1 \\ 1&0&0&0 & 1&1&1 & 1&1&1 \\ 1&0&1&1 & 0&1&1 & 0&1&1 \\ \hline 0&0&1&0 & 0&0&1 & & & \\ 0&1&1&1 & 1&1&0 & & & \\ 1&0&1&0 & 1&0&0 & & & \\ 0&0&0&0 & 1&0&0 & & & \\ 1&1&1&0 & 1&0&0 & & & \end{array}.$$

L'application de $I_{x_1^\alpha}$ donne

$$\frac{P'_{x_1}}{Q'_{x_1}}=\begin{array}{ccc|ccc|ccc} 0&1&0 & 0&1&1 & 0&1&1 \\ 0&0&0 & 1&1&1 & 1&1&1 \\ 0&1&1 & 0&1&1 & 0&1&1 \\ \hline 0&1&0 & 1&0&0 & & & \\ 1&1&0 & 1&0&0 & & & \end{array} \quad \text{et} \quad \frac{P'_{\bar{x}_1}}{Q'_{\bar{x}_1}}=\begin{array}{c|ccc|ccc} & \multicolumn{6}{c}{\emptyset} \\ \hline 0&1&0 & 0&0&1 \\ 1&1&1 & 1&1&0 \\ 0&0&0 & 1&0&0 \end{array}.$$

A l'issuede l'étape 3 seul subsiste

$$\frac{P'_{x_1}}{Q'_{x_1}}=\begin{array}{ccc|ccc|ccc} 0&1&0 & 0&1&1 & 0&1&1 \\ \hline 0&1&0 & 1&0&0 & & & \\ 1&1&0 & 1&0&0 & & & \end{array}.$$

Le terme x_1 facteur de $\frac{P'_{x_1}}{Q'_{x_1}}$ est implicant pour φ_2 et φ_3 car pour ces deux composantes Q'_{x_1} est vide. L'application de la couverture donne

$$\frac{P}{Q}=\begin{array}{cccc|ccc|ccc} 1&0&0&0 & 1&1&1 & 1&0&0 \\ \hline 0&0&1&0 & 0&0&1 & & & \\ 0&1&1&1 & 1&1&0 & & & \\ 1&0&1&0 & 1&0&0 & & & \\ 1&1&1&0 & 1&0&0 & & & \\ 0&0&0&0 & 1&0&0 & & & \end{array}, \quad \frac{P'_{x_1}}{Q'_{x_1}}=\begin{array}{c|ccc|ccc} & \multicolumn{6}{c}{\emptyset} \\ \hline 0&1&0 & 1&0&0 \\ 1&1&0 & 1&0&0 \end{array}.$$

Le dernier tableau non vide est P/Q. On applique donc I_{x_1} ce qui conduit à

$$\frac{P'_{x_1}}{Q'_{x_1}} = \begin{array}{ccc|ccc|ccc} 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 & & & \\ 1 & 1 & 0 & 1 & 0 & 0 & & & \end{array}, \quad \frac{P'_{\bar{x}_1}}{Q'_{\bar{x}_1}} = \begin{array}{c|ccc|ccc} \emptyset & & & & & & \\ \hline 0 & 1 & 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \end{array}.$$

L'algorithme se poursuit sur $P'_{\bar{x}_1}/Q'_{\bar{x}_1}$, mise en facteur de x_1 et application de I_{x_2}

$$\frac{P'_{x_2}}{Q'_{x_2}} = x_1 \begin{array}{c|ccc} \emptyset & & & \\ \hline 1 & 0 & 1 & 0 & 0 \end{array}, \quad \frac{P'_{\bar{x}_2}}{Q'_{\bar{x}_2}} = \begin{array}{ccc|ccc|ccc} 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 \\ \hline 1 & 0 & 1 & 0 & 0 & & & \end{array}.$$

$P'_{\bar{x}_2}$ disparaît car $10 \cap 00 = \emptyset$. On calcule donc $R_{x_2}(P'_{x_1}/Q'_{x_1})$ ce qui donne

$$R_{x_2}(P'_{x_1}/Q'_{x_1}) = x_1 \begin{array}{cc|ccc|ccc} 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 \\ \hline 1 & 0 & 1 & 0 & 0 & & & \end{array}$$

sur lequel applique I_{x_3}

$$\frac{P'_{x_3}}{Q'_{x_3}} = x_1 \begin{array}{c|ccc} \emptyset & & & \\ \hline 0 & 1 & 0 & 0 \end{array}, \quad \frac{P'_{\bar{x}_3}}{Q'_{\bar{x}_3}} = x_1 \begin{array}{c|ccc|ccc} 0 & 1 & 1 & 1 & 1 & 0 & 0 \\ \hline \emptyset & & & & & & \end{array}$$

$x_1\bar{x}_3$ est implicant premier avec l'étiquette 1 0 0 donc pour la composante φ_1. On a donc pour couverture irrédondante

$$\begin{cases} \varphi_1 = x_1\bar{x}_3, \\ \varphi_2 = x_1, \\ \varphi_3 = x_1. \end{cases}$$

IV- CONCLUSION

Les deux algorithmes présentés ici ont pour atout de ne générer que les implicants prémiers nécessaires, à l'obtention d'une base irrédondante, les autres étant ignorés. De plus leur programmation ne nécessite au maximum que le double de l'emplacement mémoire nécessaire aux données, ce qui, pour un problème de taille importante, est un très puissant avantage.

Abstract

In this article we introduce a modification of the recursive operator [4]. We prepare an algorithm which with the help of the recursive operator determines a nonredundant covering of a partially defined Boolean function given in disjunctive normal form. This Boolean function is given by its implicants where the function takes the value 1 resp. 0.

By a transformation of the recursive operator we extend the algorithm to determine an irredundant covering of multiple-output functions. In this case the function $F = \{\varphi_1, \varphi_2, \ldots, \varphi_m\}$ is given by their true and false minterms.

The flow-chart of the algorithms can be found in the paper and the procedures are illustrated by several examples.

Резюме

В стате мы напишем модификацию рекурсивного оператора [4]. Разрабатываем алгорифм который с помощью рекурсивного оператора определяет тупыковое покрытие не польностю определенных функций данные в Дизюнктивном нормальном форме. Эти Булевые функции определены через импликатны где функция равно 1 или 0.

После переработки рекурсивного оператора расширяем алгорифм чтобы приспособить его для определения тупыкового покрытия многовыходных функций в таком случае когда $F=\{\varphi_1, \varphi_2, \ldots, \varphi_m\}$ данная через те точки где эти функции равны 1 или 0. В стате даются блочные схемы алгорифмов и разработано несколько задач с помощью этих алгорифмов.

* LABORATOIRE D'AUTOMATIQUE ET D'ANALYSE DES SYSTEMES DU C.N.R.S. TOULOUSE

** COMPUTER AND AUTOMATION INSTITUTE HUNGARIAN ACADEMY OF SCIENCES, BUDAPEST, HUNGARY

References

[1] Mc Cluskey, E. J. & H. Shorr, Essential multiple output prime implicants, *Proc. Symposium on Mathematical Theory of Automation*, v. 12, 1962, pp. 437—452.

[2] Sze Tsen Hu, *Mathematical theory of switching circuits and automata*, University of California Press, 1968.

[3] Miller, J. R., An extension of Ghazala's methods to incompletely specified multiple output switching functions, Rep. NRL (Naval Res. Lab.) 7028, Feb. 1970.

[4] Morreale, E., Recursive operators for prime implicants and irredundant normal form determination, *IEEE Trans. Computers*, v. 19, 1970, pp. 405—409.

[5] Marcus, M. P. & W. H. Michoff, Iterated consensus method for multiple output functions, *IBM J. Res. Develop.*, 1970, pp. 677—679.

[6] Azema, P., M. Diaz & J. C. Torgue, Minimisation des fonctions logiques incomplètement spécifiées, *Electronics Letters*, v. 8, 1972, pp. 393—394.

[7] Pasztor K.—Varga, On some minimizing algorithms of Boolean Functions, *Technica Academica Scientiarum Hungaricae*, v. 72, 1972, pp. 365—378.

[8] Majithia, J. C., A simple technique for determination of essential multiple output prime implicants, *IEEE Trans. Computers*, v. 21, 1972, pp. 1024—1026.

(Reçu le 8 juin 1973*)*

Turing — Berechenbarkeit in linearer Zeit

Von K. WAGNER

Einleitung

Eine interessante Aufgabe der Kompliziertheitstheorie ist es, die Klasse der Funktionen zu untersuchen, die durch Turingmaschinen berechnet werden, an die bestimmte Kompliziertheitsforderungen gestellt sind. Das gleiche gilt für die Klasse der Mengen, die durch solche Turingmaschinen entschieden werden. Oft wird angestrebt, solche Klassen von Funktionen bzw. Mengen ohne Benutzung kompliziertheitstheoretischer Begriffe zu charakterisieren. Dies ist z. B. für die in Realzeit berechenbaren Funktionen und entscheidbaren Mengen bereits gelungen. TRACHTENBROT stellt in [1] solche Untersuchungen für die in linearer Zeit berechenbaren überall definierten Funktionen und die in linearer Zeit entscheidbaren und aufzählbaren Mengen an. Das Ziel dieser Arbeit ist es, die dort erzielten Ergebnisse für die in linearer Zeit berechenbaren partiellen Funktionen zu verallgemeinern und noch einige neue Gesichtspunkte hinzuzufügen. So sind die Sätze 1, 2 und 6 sowie die Folgerungen 1 und 2 Verallgemeinerungen der TRACHTENBROTschen Ergebnisse im eben angegebenen Sinne.

Definitionen

Mit X^* bezeichnen wir die Wortmenge über dem endlichen Alphabet X, mit e das leere Wort. Für $w \in X^*$ sei $|w|$ die Länge des Wortes w, also

$$|e| = 0$$

und

$$|wa| = |w| + 1 \quad \text{für} \quad w \in X^* \quad \text{und} \quad a \in X.$$

J sei die Operation der Wortinversion, also

$$J(e) = e$$

und

$$J(wa) = aJ(w) \quad \text{für} \quad w \in X^* \quad \text{und} \quad a \in X.$$

Es seien X und Y endliche Alphabete. Eine partielle Wortfunktion f über $[X, Y]$ ist eine Funktion aus X^* in Y^*. Mit D_f und R_f bezeichnen wir den Definitionsbereich bzw. den Wertevorrat der Funktion f. Es sei f eine partielle Wortfunktion

über $[Y, Z]$ und g eine partielle Wortfunktion über $[X, Y]$. Dann ist $f \circ g$ eine partielle Wortfunktion über $[X, Z]$ mit

$$(f \circ g)(w) = {}_{Df} f(g(w)) \quad \text{für alle} \quad w \in X^*.$$

Eine Turingmaschine $\mathfrak{M} = [X, Z, f, g, h, z_0, z_1]$ sei in dieser Arbeit stets eine einbändrige einköpfige Turingmaschine mit dem Arbeitsalphabet X, der Zustandsmenge Z, der Überführungsfunktion $f: Z \times X \to Z$, der Ausgabefunktion $g: Z \times X \to X$, der Transportfunktion $h: Z \times X \to \{+1, -1\}$, dem Anfangszustand z_1 und dem Endzustand z_0. Die Arbeitsweise einer solchen Turingmaschine wird in [1] beschrieben. Hier sei nur erwähnt, daß die Turingmaschine $\mathfrak{M}$ ihre Arbeit auf dem Wort $w \in (X - \{\S\})^*$ in dem Feld beginnt, in dem der erste Buchstabe von w steht. Dabei sei $\S \in X$ das Leerzeichen von $\mathfrak{M}$.

Eine Wortfunktion φ über $[A, B]$ mit $A \cup B \subseteqq X$ wird durch die Turingmaschine $\mathfrak{M}$ berechnet $=_{Df}$.

1. Es ist $w \in D_\varphi$ genau dann, wenn $\mathfrak{M}$ die Arbeit auf dem Wort w nach endlich vielen Schritten beendet.

2. Ist $w \in D_\varphi$, so steht nach Beendigung der Arbeit von $\mathfrak{M}$ auf w auf dem Band von $\mathfrak{M}$ das nicht durch Zeichen $\S$ unterbrochene Wort $\varphi(w)$.

Es sei weiter $\mathfrak{M}$ eine Turingmaschine, die ihre Arbeit auf dem Wort w nach endlich vielen Schritten beendet. Wir bezeichnen wie in [1] mit

$t_{\mathfrak{M}}(w)$ die Anzahl der Takte, die $\mathfrak{M}$ zur Bearbeitung des Wortes w benötigt,

$r_{\mathfrak{M}}(w)$ die maximale Anzahl von Überschreitungen einer Feldgrenze bei der Arbeit von $\mathfrak{M}$ auf w und

$S_{\mathfrak{M}}(w)$ die Anzahl der Felder, die bei der Arbeit von $\mathfrak{M}$ auf w benötigt werden. Beendet $\mathfrak{M}$ die Arbeit auf w nicht nach endlich vielen Schritten, so sind $t_{\mathfrak{M}}(w)$, $r_{\mathfrak{M}}(w)$ und $S_{\mathfrak{M}}(w)$ nicht definiert.

Die Folge von Zuständen, mit denen die Turingmaschine $\mathfrak{M}$ bei ihrer Arbeit auf dem Wort w eine bestimmte Feldgrenze überschreitet, nennen wir die Pendelfolge dieser Feldgrenze bei der Bearbeitung von w durch $\mathfrak{M}$. Dabei werden die Zustände einer Pendelfolge einer Feldgrenze so markiert, daß die jeweilige Richtung der Feldgrenzenüberschreitung ersichtlich ist.

Die partielle Wortfunktion φ heißt durch die Turingmaschine $\mathfrak{M}$ in der Zeit $T(n)$ berechnet, wenn die Turingmaschine $\mathfrak{M}$ die Funktion φ berechnet und für jedes Wort $w \in D_\varphi$ die Beziehung

$$t_{\mathfrak{M}}(w) \leqq T(|w|)$$

gilt. Die partielle Wortfunktion φ heißt in linearer Zeit berechenbar, wenn es eine Turingmaschine gibt, die φ in einer Zeit $T(n) = O(n)$ berechnet.

Die partielle Wortfunktion φ heißt mit beschränktem Regime berechenbar, wenn es eine Konstante c und eine Turingmaschine $\mathfrak{M}$, die φ berechnet, gibt, so daß für jedes Wort $w \in D_\varphi$ die Beziehung $r_{\mathfrak{M}}(w) \leqq c$ gilt. Eine Menge $M \subseteqq X^*$ heißt in linearer Zeit (mit beschränktem Regime) entscheidbar, wenn für ein festes $a \in X$ die Funktion χ_M^a mit

$$\chi_M^a(w) =_{\mathrm{Df}} \begin{cases} a, \text{ falls } w \in M \\ e, \text{ falls } w \notin M \end{cases}$$

in linearer Zeit (mit beschränktem Regime) berechenbar ist.

1. In linearer Zeit berechenbare Funktionen

Wir geben zunächst ein Lemma an, das in [1] bewiesen ist und das wir beim Beweis von Satz 1 benötigen werden.

Lemma 1 (TRACHTENBROT). Ist L eine Menge von l paarweise verschiedenen Wörtern über dem Alphabet X mit k Buchstaben, so ist

$$\sum_{w \in L} |w| \geqq c \cdot l \cdot \log_k l,$$

wobei c eine von k abhängige positive Konstante ist.

Satz 1. Berechnet eine Turingmaschine $\mathfrak{M}$ die partielle Wortfunktion φ über $[X, Y]$ in einer Zeit $T(n)=o(n \cdot \log n)$, so berechnet $\mathfrak{M}$ die Funktion φ mit beschränktem Regime.

Beweis: Wir schließen indirekt und nehmen an, daß die Turingmaschine $\mathfrak{M}$ die Funktion φ zwar in einer Zeit $T(n)=o(n \cdot \log n)$ berechnet, jedoch nicht mit beschränktem Regime. Es gibt also eine Folge $\{w_\nu\}_{\nu=1,2,\ldots}$ von Wörtern über X mit

$$r_{\mathfrak{M}}(w_1) < r_{\mathfrak{M}}(w_2) < \ldots < r_{\mathfrak{M}}(w_\nu) < \ldots .$$

Wir konstruieren aus dieser Folge eine neue Folge $\{v_\nu\}_{\nu=1,2,\ldots}$ von Wörtern über X mit

$$r_{\mathfrak{M}}(v_1) < r_{\mathfrak{M}}(v_2) < \ldots < r_{\mathfrak{M}}(v_\nu) < \ldots \qquad (a)$$

und

$$t_{\mathfrak{M}}(v_\nu) \geqq c \cdot |v_\nu| \cdot \log |v_\nu| \qquad (b)$$

mit geeigneter Konstante $c>0$.

Da wegen (a) die Glieder der Folge $\{v_\nu\}_{\nu=1,2,\ldots}$ paarweise verschieden sind, wächst die Folge $\{|v_\nu|\}_{\nu=1,2,\ldots}$ über alle Schranken und somit ist $T(n) \neq o(n \cdot \log n)$. Nun zur Konstruktion der Folge $\{v_\nu\}_{\nu=1,2,\ldots}$.

Steht ein Wort w auf dem Band der Maschine $\mathfrak{M}$, so bezeichnen wir mit $\Gamma(w)$ die Menge der Grenzen derjenigen Felder, in denen ein Buchstabe von w steht, mit $\lambda(w)$ die linke Grenze des Feldes, in dem der erste Buchstabe von w steht und mit $\varrho(w)$ die rechte Grenze des Feldes, in dem der letzte Buchstabe von w steht.

Es sei nun $w_\nu = a_\nu \cdot w'_\nu$, wobei $a_\nu \in X$ ist.

Fall 1. Bei der Bearbeitung von w_ν durch $\mathfrak{M}$ wird das Maximum $r_{\mathfrak{M}}(w_\nu)$ von Grenzüberschreitungen auf den Feldgrenzen aus $\Gamma(w'_\nu)$ nicht angenommen. Gibt es ein Teilwort u von w'_ν mit $s(\lambda(u))=s(\varrho(u))$ ($s(i)$ ist dabei die an der Grenze i bei dieser Berechnung entstehende Pendelfolge), so wird dieses u in w'_ν durch das leere Wort e ersetzt. Mit dem so erhaltenen Wort verfahren wir wie mit w'_ν usw. Nach endlich vielen Schritten gelangt man zu einem Wort v'_ν, das kein Teilwort u besitzt mit $s(\lambda(u))=s(\varrho(u))$. Dann setzen wir $v_\nu =_{\mathrm{Df}} a_\nu \cdot v'_\nu$*).

Fall 2. Es gibt Wörter p_ν und q_ν mit $w'_\nu = p_\nu \cdot q_\nu$ und auf der Feldgrenze $\varrho(p_\nu) = \lambda(q_\nu)$ wird das Maximum $r_{\mathfrak{M}}(w_\nu)$ von Grenzüberschreitungen bei der Bearbeitung von w_ν durch $\mathfrak{M}$ angenommen. Mit p_ν und q_ν wird genauso verfahren wie im Fall 1 mit w'_ν, und wir erhalten die Wörter p'_ν und q'_ν. Nun setzen wir $v_\nu =_{\mathrm{Df}} a_\nu \cdot p'_\nu \cdot q'_\nu$.

* Dieses Verfahren ist im allgemeinen nicht eindeutig bestimmt, denn es kann zu einem Wort möglicherweise mehrere Teilwörter mit der oben angegebenen Bedingung geben. Es genügt, jeweils irgendeine der möglichen Ersetzungen durchzuführen.

Die Folge $\{v_\nu\}_{\nu=1,2,\ldots}$ besitzt nun die folgenden Eigenschaften.

$$\text{Für jedes } \nu = 1, 2, \ldots \text{ ist } \varphi(v_\nu) \text{ definiert.} \tag{1}$$

Da die Anfangsbuchstaben von w_ν und v_ν gleich sind, wird zu Beginn der Arbeit von $\mathfrak{M}$ auf v_ν der gleiche Befehl angewendet wie bei w_ν, also gleicher Zustand und gleiche Bewegungsrichtung. Die Arbeit von $\mathfrak{M}$ verläuft auf v_ν ebenso, wie sie bei der Bearbeitung von w_ν durch $\mathfrak{M}$ auf den Teilwörtern von w_ν verläuft, aus denen v_ν zusammengesetzt ist. Dies sichert die Bedingung $s(\lambda(u))=s(\varrho(u))$ für ein Teilwort u, das beim Übergang von w_ν zu v_ν eliminiert wird. Andererseits besitzen für ein Teilwort u von w_ν die Folgen $\lambda(u)$ und $\varrho(u)$ verschiedene Längen, falls bei der Bearbeitung von w_ν die Turingmaschine $\mathfrak{M}$ ihre Arbeit zwischen $\lambda(u)$ und $\varrho(u)$ beendet und ihre Arbeit dort nicht beginnt. Somit kann ein solches Teilwort u von w_ν beim Übergang von w_ν zu v_ν nicht ersetzt werden, und die Turingmaschine $\mathfrak{M}$, die φ berechnet, beendet ihre Arbeit auf v_ν ebenfalls nach endlich vielen Takten.

$$\text{Für jedes } \nu = 1, 2, \ldots \text{ gilt } r_{\mathfrak{M}}(v_\nu) = r_{\mathfrak{M}}(w_\nu). \tag{2}$$

Diese Folgerung ist evident, denn jede Pendelfolge, die bei der Bearbeitung von v_ν durch $\mathfrak{M}$ entsteht, entsteht auch bei der Bearbeitung von w_ν durch $\mathfrak{M}$. Mindestens eine Feldgrenze, auf der das Maximum von Grenzüberschreitungen bei der Bearbeitung von w_ν durch $\mathfrak{M}$ angenommen wird, bleibt jedoch beim Übergang von w_ν zu v_ν erhalten.

Trivialerweise gilt nun auch

$$r_{\mathfrak{M}}(v_1) < r_{\mathfrak{M}}(v_2) < \ldots < r_{\mathfrak{M}}(v_\nu) < \ldots. \tag{3}$$

Schließlich zeigen wir: Es gibt eine Konstante $c > 0$ mit

$$t_{\mathfrak{M}}(v_\nu) \geqq c \cdot |v_\nu| \cdot \log|v_\nu| \quad \text{für alle} \quad \nu = 1, 2, \ldots. \tag{4}$$

Im oben betrachteten Fall 1 ist $|v'_\nu| \geqq \frac{1}{2}|v_\nu|$. Die Anzahl der Takte, die $\mathfrak{M}$ in dem zwischen $\lambda(v'_\nu)$ und $\varrho(v'_\nu)$ liegenden Bereich arbeitet, ist gleich

$$\sum_{\gamma \in \Gamma(v'_\nu)} |s(\gamma)|.$$

Da $s(\Gamma(v'_\nu))$ aus $|v'_\nu|+1$ verschiedenen Pendelfolgen besteht, gilt nach Lemma 1

$$t_{\mathfrak{M}}(v_\nu) \geqq \sum_{\gamma \in \Gamma(v'_\nu)} |s(\gamma)| \geqq c'|v'_\nu| \cdot \log|v'_\nu| \quad \text{mit} \quad c' > 0.$$

Wegen $|v'_\nu| \geqq \frac{1}{2}|v_\nu|$ existiert eine Konstante $c > 0$ mit

$$t_{\mathfrak{M}}(v_\nu) \geqq c \cdot |v_\nu| \cdot \log|v_\nu|.$$

Im oben betrachten Fall 2 gilt entweder $|p'_\nu| \geqq \frac{1}{3}|v_\nu|$ oder $|q'_\nu| \geqq \frac{1}{3}|v_\nu|$. Hier ergibt sich die gleiche Rechnung wie im Fall 1. Damit ist Satz 1 bewiesen.

Satz 2. Berechnet die Turingmaschine $\mathfrak{M}$ die partielle Wortfunktion φ über $[X, Y]$ mit beschränktem Regime, so berechnet $\mathfrak{M}$ die Funktion φ in linearer Zeit.

Beweis: Wir wählen eine Konstante R, so daß für jedes $w \in D_\varphi$ die Ungleichung $r_{\mathfrak{M}}(w) \leqq R$ gilt. Ist $w \in D_\varphi$, so erhält man

$$t_{\mathfrak{M}}(w) \leqq R \cdot (S(w)+1).$$

Da es nur endlich viele Pendelfolgen mit der maximalen Länge R gibt und wegen $w \in D_\varphi$ sowie links von $\lambda(w)$ als auch rechts von $\varrho(w)$ jede dieser Pendelfolgen nur einmal auftreten kann, gibt es eine Konstante c' mit $S(w) \leqq |w| + c'$. Mithin gibt es eine Konstante c mit $t_{\mathfrak{M}}(w) \leqq c \cdot |w|$. Damit ist Satz 2 bewiesen.

Folgerung 1. Berechnet eine Turingmaschine $\mathfrak{M}$ die partielle Wortfunktion φ in einer Zeit $T(n) = o(n \cdot \log n)$, so berechnet $\mathfrak{M}$ die Funktion φ bereits in linearer Zeit.

Für eine zweistellige partielle Wortfunktion ψ über $[X, Y]$ definieren wir für alle $w \in X^*$ die Funktionen $\psi_{1,w}$ und $\psi_{2,w}$ mit

$$\psi_{1,w}(w') = {}_{\mathrm{Df}}\psi(w, w')$$

und

$$\psi_{2,w}(w') = {}_{\mathrm{Df}}\psi(w', w).$$

Weiter sei $G_1(\psi) = {}_{\mathrm{Df}} kz\{\psi_{1,w},\ w \in X^*\}$ das Gewicht von ψ in der 1. Stelle und $G_2(\psi) = {}_{\mathrm{Df}} kz\{\psi_{2,w},\ w \in X^*\}$ das Gewicht von ψ in der 2. Stelle.

Satz 3. Eine partielle Wortfunktion φ über $[X, Y]$ ist genau dann mit beschränktem Regime berechenbar, wenn es zwei zweistellige partielle Wortfunktionen φ_l und φ_r gibt mit

1) Die Werte $\varphi(w, w')$, $\varphi_l(w, w')$ und $\varphi_r(w, w')$ sind alle gleichzeitig definiert oder nicht definiert. Im Falle ihrer Existenz gilt

$$\varphi(w \cdot w') = \varphi_l(w, w') \cdot \varphi_r(w, w').$$

2) Ist $\varphi(waw')$ mit $a \in X$ definiert, so gilt $|\varphi_l(wa, w')| = |\varphi_l(w, aw')| + 1$, falls $\varphi_l(wa, w') \neq e$ und $\varphi_r(w, aw') \neq e$ und es gilt $|\varphi_l(wa, w')| = |\varphi_l(w, aw')|$, sonst.

3) Es sind $G_2(\varphi_l)$ und $G_1(\varphi_r)$ endlich und aus $(\varphi_l)_{2,w} = (\varphi_l)_{2,w'}$ $((\varphi_r)_{1,w} = (\varphi_r)_{1,w'})$ folgt $(\varphi_l)_{2,aw} = (\varphi_l)_{2,aw'}$ $((\varphi_r)_{1,wa} = (\varphi_r)_{1,w'a})$ für alle $a \in X$.

Bemerkung: Man sieht wegen 1) leicht ein, daß 2) durch folgende Bedingung äquivalent ersetzt werden kann.

2') Ist $\varphi(waw')$ mit $a \in X$ definiert, so gilt $|\varphi_r(w, aw')| = |\varphi_r(wa, w')| + 1$, falls $\varphi_l(wa, w') \neq e$ und $\varphi_r(w, aw') \neq e$ und es gilt $|\varphi_r(w, aw')| = |\varphi_r(wa, w')|$, sonst.

Beweis von Satz 3: I. Die Turingmaschine $\mathfrak{M}$ berechne φ mit beschränktem Regime. Ist $\varphi(w \cdot w')$ für $w, w' \in X^*$ definiert, so setzen wir $\varphi_l(w, w')$ gleich dem Wort, welches nach der Bearbeitung von $w \cdot w'$ durch $\mathfrak{M}$ links von $\lambda(w') = \varrho(w)$ steht und $\varphi_r(w, w')$ gleich dem Wort, welches nach der Bearbeitung von $w \cdot w'$ durch $\mathfrak{M}$ rechts von $\lambda(w') = \varrho(w)$ steht. Ist $\varphi(w \cdot w')$ nicht definiert, so seien $\varphi_l(w, w')$ und $\varphi_r(w, w')$ ebenfalls nicht definiert.

Die Bedingungen 1) und 2) sind trivialerweise erfüllt. Es ist die Bedingung 3) nachzuweisen. Wir zeigen zunächst, daß $G_2(\varphi_l)$ endlich ist. Da die Turingmaschine $\mathfrak{M}$ mit beschränktem Regime arbeitet, gibt es eine natürliche Zahl R, so daß $r_{\mathfrak{M}}(v) \leqq 2R + 1$ für alle $v \in D_\varphi$ gilt. Außerdem besitze $\mathfrak{M}$ genau k Zustände.

Die Frage nach $G_2(\varphi_l)$ ist die Frage danach, auf wieviel verschiedene Arten das am Beginn einer Berechnung durch $\mathfrak{M}$ rechts von einer gewissen Feldgrenze stehende Wort auf den Berechnungsvorgang links von dieser Feldgrenze Einfluß nehmen kann. Diese Einflußnahme ist jedoch nur möglich durch die Zustände, in denen der Lese- und Schreibekopf von $\mathfrak{M}$ diese Feldgrenze von rechts nach links überschreitet.

Da die erste Überschreitung dieser Grenze von links nach rechts geschieht, interessieren nur die ersten R Überschreitungen dieser Grenze von rechts nach links. Gibt es mehr als R solche Überschreitungen, so ist φ_l auf den zugrundeliegenden Argumenten nicht definiert. Von links nach rechts kann diese Grenze $(R+1)$-mal überschritten werden. Wird diese Feldgrenze in einem bestimmten Zustand von links nach rechts überschritten, so sind $k+2$ „Reaktionen" möglich:

1. Rückkehr über die Grenze mit einem der k möglichen Zustände,
2. $\mathfrak{M}$ beendet die Arbeit rechts von dieser Grenze oder
3. $\mathfrak{M}$ arbeitet rechts von dieser Grenze ad infinitum weiter (also φ_l ist auf diesen Argumenten nicht definiert).

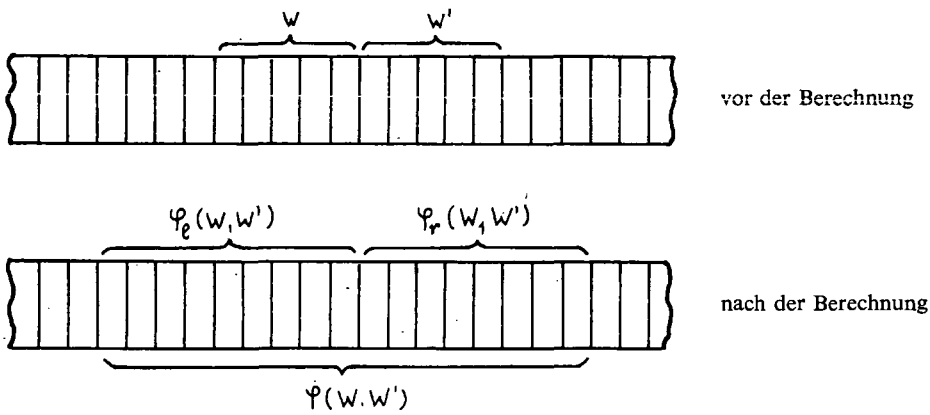


Abb. 1

Insgesamt sind dies $(k+2)^k$ mögliche „Reaktionen" bei jeder Überschreitung der Grenze von links nach rechts. Ausgenommen ist dabei die $(R+1)$-te Überschreitung dieser Grenze von links nach rechts, denn dort folgt aus der Rückkehr über die Grenze, daß φ_l auf diesen Argumenten nicht definiert ist. In diesem Fall gibt es also als „Reaktion" auf einen bestimmten Zustand nur zwei Möglichkeiten, also insgesamt 2^k Möglichkeiten. Damit ergibt sich schließlich $G_2(\varphi_l) \leqq ((k+2)^k)^R \cdot 2^k$. Außerdem ist klar, daß die „Reaktionen" bei einer Überschreitung einer Feldgrenze von links nach rechts nur von den „Reaktionen" beim Überschreiten der nächsten rechts gelegenen Feldgrenze und dem Inhalt der zwischen diesen beiden Feldgrenzen liegenden Feldes abhängt. Also haben wir mit $(\varphi_l)_{2,w} = (\varphi_l)_{2,w'}$ auch $(\varphi_l)_{2,aw} = (\varphi_l)_{2,aw'}$ für alle $a \in X$.

Analog zeigt man, daß $G_1(\varphi_r)$ endlich ist und daß mit $(\varphi_r)_{1,w} = (\varphi_r)_{1,w'}$ auch $(\varphi_r)_{1,wa} = (\varphi_r)_{1,w'a}$ gilt für alle $a \in X$.

II. Es seien φ, φ_l und φ_r Wortfunktionen, die den Bedingungen 1)—3) genügen.

Die Wörter $w, w' \in X^*$ stehen genau dann in der Relation $\sim_l$ ($\sim_r$), wenn $(\varphi_l)_{2,w} = (\varphi_l)_{2,w'}$ $((\varphi_r)_{1,w} = (\varphi_r)_{1,w'})$ ist. Es ist klar, daß $\sim_l$ und $\sim_r$ Äquivalenzrelationen auf X^* sind. Mit $w^{(l)}$ bzw. $w^{(r)}$ bezeichnen wir diejenige Äquivalenzklasse bzgl. $\sim_l$ bzw. $\sim_r$, in der w liegt. Aus 3) folgt nun

a) Die Äquivalenzrelation $\sim_l$ ist linksstabil bzgl. der Operation der Wortaneinandersetzung.
b) Die Äquivalenzrelation $\sim_r$ ist rechtsstabil bzgl. der Operation der Wortaneinandersetzung.

Ist $\varphi(u \cdot v \cdot w)$ für $u, v, w \in X^*$ definiert, so bezeichnen wir mit $\varphi_m(u, v, w)$ dasjenige Wort, für das gilt

$$\varphi_l(uv, w) = \varphi_l(u, vw) \cdot \varphi_m(u, v, w). \tag{α}$$

Ist $\varphi(u \cdot v \cdot w)$ nicht definiert, so ist $\varphi_m(u, v, w)$ ebenfalls nicht definiert.

Im Falle der Existenz von $\varphi(u \cdot v \cdot w)$ haben wir sofort auch

$$\varphi_r(u, vw) = \varphi_m(u, v, w) \cdot \varphi_l(uv, w). \tag{β}$$

Die Paare $(v_1, w_1), (v_2, w_2) \in X^* \times X^*$ stehen genau dann in der Relation $\sim_m$, wenn $\varphi_m(v_1, u, w_1) = \varphi_m(v_2, u, w_2)$ für jedes $u \in X^*$ gilt. Die Relation $\sim_m$ ist eine Äquivalenzrelation und wir zeigen

c) Aus $u_1 \sim_r u_2$ und $w_1 \sim_l w_2$ folgt $(u_1, w_1) \sim_m (u_2, w_2)$.

Ist $\varphi(u_1 v w_1)$ definiert, so gilt

$$\begin{aligned}
\varphi_m(u_1, v, w_1) \cdot \varphi_r(u_2 v, w_1) &= \varphi_m(u_1, v, w_1) \cdot \varphi_r(u_1 v, w_1) \quad \text{wegen } u_1 \sim_r u_2 \text{ und } b) \\
&= \varphi_r(u_1, v w_1) \text{ wegen } (\beta) \\
&= \varphi_r(u_2, v w_1) \text{ wegen } u_1 \sim_r u_2 \\
&= \varphi_m(u_2, v, w_1) \cdot \varphi_r(u_2 v, w_1).
\end{aligned}$$

Mithin gilt $\varphi_m(u_1, v, w_1) = \varphi_m(u_2, v, w_1)$. Analog folgt aus $w_1 \sim_l w_2$ die Gleichung $\varphi_m(u_2, v, w_1) = \varphi_m(u_2, v, w_2)$.

Ist $\varphi(u_1 v w_1)$ nicht definiert, so sind auch $\varphi_m(u_1, v, w_1)$ und $\varphi_m(u_2, v, w_2)$ nicht definiert. Damit ist *c)* bewiesen.

Mit *c)* wurde gezeigt, daß $\varphi_m(u, v, w)$ lediglich von v und der $\sim_r$ — bzw. $\sim_l$ — Klasse abhängt, in der u bzw. w liegt. Mit *c)* wurde also die Reräsentantenunabhängigkeit der Definition

$$\varphi'_m(u^{(r)}, v, w^{(l)}) =_{\mathrm{DI}} \varphi_m(u, v, w)$$

gezeigt. Damit haben wir auch

d) Ist $\varphi(a_1 \cdot \ldots \cdot a_n)$ für $a_1, \ldots, a_n \in X$ definiert, so gilt

$$\begin{aligned}
\varphi(a_1 \cdot \ldots \cdot a_n) = \varphi_l(e, a_1 \cdot \ldots \cdot a_n) \cdot \varphi'_m(e^{(r)}, a_1, (a_2 \cdot \ldots \cdot a_n)^{(r)}) \cdot \ldots \cdot \\
\ldots \cdot \varphi'_m((a_1 \cdot \ldots \cdot a_{i-1})^{(r)}, a_i, (a_{i+1} \cdot \ldots \cdot a_n)^{(l)}) \cdot \ldots \cdot \\
\ldots \cdot \varphi'_m((a_1 \cdot \ldots \cdot a_{n-1})^{(r)}, a_n, e^{(l)}) \cdot \varphi_r(a_1 \cdot \ldots \cdot a_n, e),
\end{aligned}$$

wobei $\varphi'_m((a_1 \cdot \ldots \cdot a_{i-1})^{(r)}, a_i, (a_{i+1} \cdot \ldots \cdot a_n)^{(l)}) \in Y \cup \{e\}$ für $i = 1, \ldots, n$.

Nun können wir eine Turingmaschine $\mathfrak{M}$ konstruieren, die die Funktion φ mit beschränktem Regime berechnet. Diese Turingmaschine $\mathfrak{M}$ arbeitet wie folgt: Zunächst ersetzt $\mathfrak{M}$ in dem auf dem Band stehenden Wort $a_1 \cdot \ldots \cdot a_n$ $(a_1, \ldots, a_n \in X)$ von links nach rechts jeden Buchstaben a_i durch das Zeichen $((a_1 \cdot \ldots \cdot a_{i-1})^{(r)}, a_i)$. Dabei hat $\mathfrak{M}$ nach Abarbeitung des Anfangswortes $a_1 \cdot \ldots \cdot a_n$ den Zustand $(a_1 \cdot \ldots \cdot a_i)^{(r)}$ inne.

Nach *a)* ist das ausreichend, um im nächsten Schritt den Zustand $(a_1 \cdot \ldots \cdot a_{i+1})^{(r)}$ annehmen zu können. Am Wortende angelangt, wird $\varphi_r(a_1 \cdot \ldots \cdot a_n, e)$ auf das Band geschrieben, falls $\varphi_r(a_1 \cdot \ldots \cdot a_n, e)$ definiert ist. Nun wird das Wort von rechts nach links durchlaufen, wobei das Feld mit dem Inhalt $((a_1 \cdot \ldots \cdot a_{i-1})^{(r)}, a_i)$ mit dem Zustand $(a_{i+1} \cdot \ldots \cdot a_n)^{(l)}$ erreicht (dies ist nach *b)* möglich) und mit dem Buchstaben $\varphi'_m((a_1 \cdot \ldots \cdot a_{i-1})^{(r)}, a_i, (a_{i+1} \cdot \ldots \cdot a_n)^{(l)})$ beschrieben wird, falls $\varphi'_m((a_1 \cdot \ldots \cdot a_{i-1})^{(r)}, a_i, (a_{i+1} \cdot \ldots \cdot a_n)^{(l)})$ ein Buchstabe ist bzw. mit ? beschrieben wird, falls $\varphi'_m((a_1 \cdot \ldots \cdot a_{i-1})^{(r)}, a_i, (a_{i+1} \cdot \ldots \cdot a_n)^{(l)}) = e$. Ist der Wortanfang erreicht, so wird, falls definiert, $\varphi_l(e, a_1 \cdot \ldots \cdot a_n)$ auf das Band geschrieben. Ist irgendeiner der benötigten Funktionswerte nicht definiert, so läuft $\mathfrak{M}$ ad infinitum nach links.

Es ist klar, daß $\mathfrak{M}$ die Funktion φ mit beschränktem Regime berechnet. Damit ist Satz 3 bewiesen.

Aus dem Beweis des Satzes 3 ergibt sich die

Folgerung 2. Ist die partielle Wortfunktion φ über $[X, Y]$ mit beschränktem Regime berechenbar, so gibt es eine Turingmaschine, die φ berechnet und deren Lese- und Schreibekopf bei der Bearbeitung eines jeden Wortes aus X^* nur einmal die Bewegungsrichtung ändert (Bewegung des Kopfes zuerst von links nach rechts).

Aus Folgerung 2 ergibt sich eine andere Charakterisierung der in linearer Zeit berechenbaren partiellen Wortfunktionen. Zunächst eine Definition.

Eine partielle Wortfunktion φ heißt genau dann einseitig berechenbar, wenn es eine Turingmaschine gibt, die φ berechnet und deren Lese- und Schreibekopf sich in jedem Arbeitstakt um ein Feld nach rechts bewegt. Solche einseitig berechenbaren Funktionen wurden von MODROW in [2] untersucht. Setzt man noch voraus, daß diese Funktionen überall definiert sind, so sind sie fastsequentiell im Sinne von WECHSUNG ([3]).

Satz 4. Eine partielle Wortfunktion φ ist genau dann mit beschränktem Regime berechenbar, wenn es zwei einseitig berechenbare Wortfunktionen φ_1 und φ_2 gibt mit $\varphi = J \circ \varphi_2 \circ J \circ \varphi_1$.

Dieser Satz ergibt sich aus Folgerung 2.

Den Inhalt der Sätze 1—4 fassen wir zusammen.

Satz 5. Für eine partielle Wortfunktion φ über $[X, Y]$ sind folgende Aussagen äquivalent:

a) φ ist in einer Zeit $T(n) = o(n \cdot \log n)$ berechenbar.
b) φ ist in linearer Zeit berechenbar.
c) φ ist mit beschränktem Regime berechenbar.
d) Es existieren zwei einseitig berechenbare Funktionen φ_1 und φ_2 mit $\varphi = J \circ \varphi_2 \circ J \circ \varphi_1$.
e) Es existieren zwei zweistellige partielle Wortfunktionen φ_l und φ_r mit

1) Die Werte $\varphi(w \cdot w')$, $\varphi_l(w, w')$ und $\varphi_r(w, w')$ sind alle gleichzeitig oder nicht definiert. Im Falle ihrer Existenz gilt

$$\varphi(w \cdot w') = \varphi_l(w, w') \cdot \varphi_r(w, w').$$

2) Ist $\varphi(waw')$ mit $a \in X$ definiert, so gilt $|\varphi_l(wa, w')| = |\varphi_l(w, aw')| + 1$, falls $\varphi_l(wa, w') \neq e$ und $\varphi_r(w, aw') \neq e$ und es gilt $|\varphi_l(wa, w')| = |\varphi_l(w, aw')|$, sonst.

3) Es sind $G_2(\varphi_l)$ und $G_1(\varphi_r)$ endlich und aus $(\varphi_l)_{2,w} = (\varphi_l)_{2,w'}((\varphi_r)_{1,w} = (\varphi_r)_{1,w'})$ folgt $(\varphi_l)_{2,aw} = (\varphi_l)_{2,aw'}((\varphi_r)_{1,wa} = (\varphi_r)_{1,w'a})$ für alle $a \in X$.

2. In linearer Zeit entscheidbare und aufzählbare Mengen

Wir wollen hier einen Satz von TRACHTENBROT ([1]) verallgemeinern, der besagt, daß die Klasse der mit beschränktem Regime (d.h. in linearer Zeit) entscheidbaren Wortmengen zusammenfällt mit der Klasse der durch überall definierte, in linearer Zeit berechenbare Funktionen aufgezählte Mengen und der Klasse der regulären Mengen.

Satz 6. Folgende Aussagen sind für eine Wortmenge M äquivalent:

a) M ist in linearer Zeit entscheidbar.
b) M ist Wertebereich einer in linearer Zeit berechenbaren partiellen Wortfunktion.
c) M ist Definitionsbereich einer in linearer Zeit berechenbaren partiellen Wortfunktion.
d) M ist regulär.

Beweis: I. Aus *a)* folgt *c)*. Es sei $\mathfrak{M}$ eine Turingmaschine, die M mit beschränktem Regime entscheidet. Nach Beendigung der Arbeit von $\mathfrak{M}$ auf einem Wort $w \in M$ steht auf dem Band ein ausgezeichneter Buchstabe a, in dem Fall $w \notin M$ ist das Band nach der Beendigung der Arbeit von $\mathfrak{M}$ leer. Wir setzen o.B.d.A. voraus, daß im ersten Fall $\mathfrak{M}$ links von a anhält. Wir betrachten eine Turingmaschine $\mathfrak{M}'$, die wie $\mathfrak{M}$ arbeitet, aber im Endzustand von $\mathfrak{M}$ nach rechts läuft, bis ein von $\overset{\circ}{|}$ verschiedener Buchstabe erreicht wird. Wird ein solcher Buchstabe erreicht, so hält $\mathfrak{M}'$ an, im anderen Falle läuft $\mathfrak{M}'$ ad infinitum nach rechts. Somit ist ein Wort w genau dann in M, wenn die durch $\mathfrak{M}$ (mit beschränktem Regime) berechnete Funktion auf w definiert ist.

II. Aus *c)* folgt *b)*. Es sei $\mathfrak{M}$ eine Turingmaschine, die die Funktion φ mit beschränktem Regime berechnet. Wir konstruieren eine Turingmaschine $\mathfrak{M}'$, die eine Funktion φ' mit beschränktem Regime berechnet, so daß $D_\varphi = R_{\varphi'}$ gilt.

Die Turingmaschine $\mathfrak{M}'$ arbeitet wie $\mathfrak{M}$, nur daß sie anstelle des Buchstaben y das Zeichen (x, y) druckt, wobei x der urspüngliche Inhalt des betreffenden Feldes ist. Wird der Endzustand von $\mathfrak{M}$ erreicht, so wird überall das Zeichen (x, y) durch x ersetzt. Hier die Befehlsliste von $\mathfrak{M}'$:

Ist $zx \to z'y\sigma$ $(\sigma \in \{+1, -1\})$ ein Befehl von $\mathfrak{M}$, so sind

$zx \to z'(x, y)\sigma$ und

$z(x', x) \to z'(x', y)\sigma$ (x' ist aus dem Arbeitsalphabet von $\mathfrak{M}$).

Befehle von $\mathfrak{M}'$.

Ist z_0 der Endzustand von $\mathfrak{M}$, so sind

$z_0(x, y) \to z_0(x, y) - 1$ (x und y sind aus dem Arbeitsalphabet von $\mathfrak{M}$),

$z_0 x \to z_0' x + 1$,

$z_0'(x, y) \to z_0' x + 1$ und

$z_0' x \to s_0 x + 1$

Befehle von $\mathfrak{M}'$. Dabei sind z_0' und s_0 neue Zustände, s_0 ist der Endzustand von $\mathfrak{M}'$. Es ist evident, das $\mathfrak{M}'$ das Gewünschte leistet.

III. Aus *b)* folgt *d)*. Es sei M der Definitionsbereich einer in linearer Zeit berechenbaren partiellen Wortfunktion φ über $[X, Y]$. Nach Satz 4 gibt es zwei zweistellige einseitig berechenbare partielle Wortfunktionen φ_1 und φ_2 mit $\varphi = J \circ \varphi_2 \circ J \circ \varphi_1$. Dann gibt es zwei partielle sequentielle Funktionen ψ_1 und ψ_2 mit endlichem Gewicht und $\varphi_1(w) = \psi_1(w*)$ sowie $\varphi_2(w) = \psi_2(w*)$ für alle $w \in X^*$ und einem geeigneten Buchstaben $* \notin X$. Die Menge $X^* \cdot \{*\} =_{\mathrm{Df}} =_{\mathrm{Df}} \{w* ; w \in X^*\}$ ist regulär. Dann ist auch die Menge $\psi_1(X^* \cdot \{*\}) = \varphi_1(X^*)$ regulär. Die Menge $M_1 =_{\mathrm{Df}} J(\varphi_1(X^*))$ ist regulär und somit auch die Menge $M_1 \cdot \{*\} =_{\mathrm{Df}} \{w* ; w \in M_1\}$. Dann ist aber die Menge $\psi_2(M_1 \cdot \{*\}) = \varphi_2(M_1)$ ebenfalls regulär. Mithin ist $M = J(\varphi_2(J(\varphi_1(X^*)))) = J(\varphi_2(M_1))$ regulär.

IV. Die Aussage *a)* folgt trivialerweise aus *d)*. Damit ist Satz 6 bewiesen.

Zum Schluß sei Herrn Doz. Dr. WECHSUNG, Jena, für die wertvollen Gespräche und Hinweise zu diesen Problemen gedankt.

Kurzfassung

Das TRACHTENBROTsche Ergebnis, daß jede Turingmaschine, die eine überall definierte Wortfunktion in einer Zeit $T(n) = o(n \cdot \log n)$ berechnet, diese Funktion bereits in linearer Zeit und mit beschränktem Regime berechnet, wird auf den Fall partieller Wortfunktionen verallgemeinert, und es wird eine „maschinenfreie" Charakterisierung der in linearer Zeit berechenbaren partiellen Wortfunktionen angegeben. Wie im Fall der überall definierten Wortfunktionen gilt auch für die in linearer Zeit berechenbaren partiellen Wortfunktionen, daß die Klasse der Wertebereiche dieser Funktionen mit der Klasse der regulären Mengen zusammenfällt, und außerdem ist diese identisch mit der Klasse der Definitionsbereiche dieser Funktionen.

SEKTION MATHEMATIK DER
FRIEDRICH-SCHILLER-UNIVERSITÄT
69 JENA — DDR
UNIVERSITÄTSHOCHHAUS

Literatur

[1] Трахтенброт, Б. А., Сложность алгоритмов и вычислений, Новосибирск 1967.
[2] MODROW, H. D., Eine Erweiterung der Theorie der determinierten abstrakten Automaten auf nicht-sequentielle Wortfunktionen., Diss. A, Humboldt-Universität Berlin, 1969.
[3] WECHSUNG, G., Quasisequentielle Funktionen, *Acta Cybernet.*, Bd. 2, Heft 1, 1973, S. 24—33.

(Eingegangen am 30. Juli 1973)

Tessellation transformations

By L. SZABÓ

I. Introduction. Notations

In this paper we present some results on mappings induced by cellular automata. These mappings will be called here tessellation transformations. The notions and notations may be found partly in [2] and [7], but for the sake of the convenience of the reader, all necessary definitions will be given in this introductory part and at the beginning of the third section.

In the second section we deal with semigroups consisting of all tessellation transformations in a given tessellation array, under a fixed number of internal states. They will be characterized up to isomorphism by two parameters. Some inclusion theorems for these semigroups are also proved. The third section of our paper concerns cellular automata with a quiescent state. The investigations are related to and may be considered as a continuation of Moore's and Myhill's results in this area. Among others, it will be shown that the density of the tessellation transformations which are one-to-one on the finite configurations is equal to zero in the set of all such transformations. This solves a problem raised by Moore in [5].

Now we shall list the basic concepts.

A d-dimensional cellular automaton (shortly: CA) is a quadruple $\mathfrak{A}=(A, E^d, X, f)$, where

1. A is a finite set called the state alphabet. Its cardinality is supposed to be at least two.

2. E^d, called the d-dimensional tessellation array, is the set of all d-tuples of integers called cells. E^d is an Abelian group with respect to the componentwise sum of the d-tuples. E^d can be visualized as a Euclidean d-space subdivided into cells which are d-cubes of unit dimensions and whose centers have integer coordinates.

3. X, called the neighbourhood template, is an n-tuple of distinct elements of E^d (n is a positive integer), i.e.,

$$X = (\xi_1, \ldots, \xi_n), \quad \xi_i \in E^d, \qquad i = 1, \ldots, n.$$

For any $\alpha \in E^d$, $N(X, \alpha)=(\alpha+\xi_1, \ldots, \alpha+\xi_n)$ is said to be the neighbourhood of the cell α.

4. f is an arbitrary function from A^n into A called the local transition function.

We shall refer to a mapping $c: E^d \to A$ as configuration (more precisely: d-dimensional configuration over the alphabet A). The set of all configurations is denoted

by $C_{A,d}$. The image $c(\alpha)$ of $\alpha \in E^d$ will be called the contents of the cell α under the configuration c. The restriction of c to $N(X, \alpha)$ is denoted by $c(N(X, \alpha))$, i.e.,

$$c(N(X, \alpha)) = (c(\alpha+\xi_1), \ldots, c(\alpha+\xi_n)).$$

The global transition function $\Phi_{\mathfrak{A}}: C_{A,d} \to C_{A,d}$ of the CA is defined by

$$(c\Phi_{\mathfrak{A}})(\alpha) = f(c(N(X, \alpha))) \quad \text{for all} \quad \alpha \in E^d.$$

II. Full semigroups of tessellation transformations

Let A be a finite nonempty set ($|A| \geqq 2$) and let d be a positive integer. A mapping $\Phi: C_{A,d} \to C_{A,d}$ is said to be a tessellation transformation if there exists a CA $\mathfrak{A}$ such that its global transition function $\Phi_{\mathfrak{A}}$ is equal to Φ. The set of all such mappings will be denoted by $M_{A,d}$.

Theorem 1. If $\Phi, \Psi \in M_{A,d}$, then $\Phi\Psi \in M_{A,d}$, i.e. $M_{A,d}$ is a semigroup.

Proof. According to the assumption of the theorem there are two CA $\mathfrak{A}^{(1)} = (A, E^d, X^{(1)}, f^{(1)})\,(X^{(1)} = (\xi_1^{(1)}, \ldots, \xi_n^{(1)}))$ and $\mathfrak{A}^{(2)} = (A, E^d, X^{(2)}, f^{(2)})\,(X^{(2)} = (\xi_1^{(2)}, \ldots, \xi_m^{(2)}))$ such that $\Phi_{\mathfrak{A}^{(1)}} = \Phi$ and $\Phi_{\mathfrak{A}^{(2)}} = \Psi$. Let us consider a CA $\mathfrak{A} = (A, E^d, X, f)$, where

1. The set of the components of X is $\langle \xi_i^{(1)} + \xi_j^{(2)} | 1 \leqq i \leqq n, 1 \leqq j \leqq m \rangle$.

2. We obtain f in the following way: Consider the function $f' = f^{(2)}(f^{(1)}(x_{11}, \ldots, x_{1n}), \ldots, f^{(1)}(x_{m1}, \ldots, x_{mn}))$. Identify x_{ij} with $x_{i'j'}$ if $\xi_i^{(1)} + \xi_j^{(2)} = \xi_{i'}^{(1)} + \xi_{j'}^{(2)}$, $1 \leqq i, i' \leqq n$, $1 \leqq j, j' \leqq m$. Then we obtain from the function f' a new function f''. Finally, writing the variables of f'' in the order which corresponds to X we get f. It is easy to see that $\Phi_{\mathfrak{A}} = \Phi\Psi$.

We call $M_{A,d}$ the full semigroup of tessellation transformations.

The transformations θ_δ ($\delta \in E^d$) defined by $(c\theta_\delta)(\alpha) = c(\alpha - \delta)$ for all $\alpha \in E^d$ are called translations. They are obviously tessellation transformations. A $c' \in C_{A,d}$ is said to be a copy of $c \in C_{A,d}$ if there is a translation θ_δ such that $c' = c\theta_\delta$. Clearly, if $c = c\theta_\delta$ for all $\delta \in E^d$, then there exists an $a \in A$ such that $c(\alpha) = a$ for all $\alpha \in E_d$. It is evident that $\theta_{\delta_1}\theta_{\delta_2} = \theta_{\delta_1+\delta_2}$ for any $\delta_1, \delta_2 \in E^d$, and if T_d denotes the set of all translations we get $T_d \cong E^d$.

The transformations Ω_a ($a \in A$) defined by $(c\Omega_a)(\alpha) = a$ for all $\alpha \in E^d$ are called constant transformations (briefly: constants). They are tessellation transformations. It is trivial that the number of all constants is $|A|$.

Lemma 1. A tessellation transformation Φ is a central element of $M_{A,d}$ if and only if Φ is a translation.

Proof. Let $Z(M_{A,d})$ denote the center of $M_{A,d}$. The set of all translations $T_d \subset Z(M_{A,d})$ is trivial. Now suppose that $\Phi \in Z(M_{A,d})$ and suppose $\mathfrak{A} = (A, E^d, X, f)\,(X = (\xi_1, \ldots, \xi_n))$ is a CA such that $\Phi_{\mathfrak{A}} = \Phi$. Consider a CA $\mathfrak{B} = (A, E^d, Y, g)\,(Y = (\eta_1, \ldots, \eta_m))$ such that every element of the set $\langle \xi_i + \eta_j | 1 \leqq i \leqq n, 1 \leqq j \leqq m \rangle$ has a unique representation as a sum of components of X and Y. (For any X there exists such a

Y with an arbitrary number of componens.) Under such choice of Y, $\Phi_{\mathfrak{A}}\Phi_{\mathfrak{B}}=\Phi_{\mathfrak{B}}\Phi_{\mathfrak{A}}$ implies

$$f(g(x_{11}, \ldots, x_{1m}), \ldots, g(x_{n1}, \ldots, x_{nm})) = g(f(x_{11}, \ldots, x_{n1}), \ldots, f(x_{1m}, \ldots, x_{nm}))$$

for all $x_{11}, \ldots, x_{nm} \in A$ (see 2. in the proof of Theorem 1). Accordingly we obtain that f commutes with all functions defined on A. Hence it follows that f is a projection ([3] pp. 128, Prop. 3.2.), consequently $\Phi_{\mathfrak{A}}=\Phi$ is a translation.

Lemma 2. A tessellation transformation Φ is a right zero if and only if Φ is a constant.

Proof. The sufficiency is trivial. Assume that Φ is a right zero and let $\mathfrak{A}=(A, E^d, X, f)$ (with n-ary f) be a CA such that $\Phi_{\mathfrak{A}}=\Phi$. Let $c \in C_{A,d}$ be a configuration for which

$$A^n = \langle c(N(X, \alpha)) | \alpha \in E^d \rangle \qquad (*)$$

holds. Since Φ is a right zero, we obtain $(c\Phi)\theta_\delta = c(\Phi\theta_\delta) = c(\theta_\delta\Phi) = c\Phi$ for all translations θ_δ, whence there exists an $a \in A$ such that $(c\Phi)(\alpha)=a$ for all $\alpha \in E^d$. According to $(*)$ it follows that $f(x_1, \ldots, x_n)=a$ for any $x_1, \ldots, x_n \in A$ showing that Φ is a constant.

Theorem 2. $M_{A_1,d_1} \cong M_{A_2,d_2}$ if and only if $|A_1|=|A_2|$ and $d_1=d_2$.

Proof. The sufficiency is trivial. $M_{A_1,d_1} \cong M_{A_2,d_2}$ implies that the numbers of the right zeros of M_{A_1,d_1} and M_{A_2,d_2} are equal, i.e., in view of Lemma 2 we have $|A_1|=|A_2|$. Furthermore $Z(M_{A_1,d_1}) \cong Z(M_{A_2,d_2})$, whence by Lemma 1 we get $T_{d_1} \cong T_{d_2}$, which implies $E^{d_1} \cong E^{d_2}$. Hence $d_1=d_2$.

According to Theorem 2 a full semigroup of tessellation transformations is determined up to isomorphism by two positive integers $l(=|A|)$ and d. Therefore we shall denote this semigroup also by $M_{l,d}$.

Theorem 3. For any positive integer l_1, l_2 ($\geqq 2$) and d_1, d_2 such that $d_1 \leqq d_2$, the semigroup M_{l_1,d_1} is the homomorphic image of a subsemigroup of M_{l_1,d_1}.

Proof. The reader can easily verify that, if $d_1 \leqq d_2$ then M_{l,d_1} may be embedded in M_{l,d_2} for any l. Therefore it is sufficient to prove the statement for $d_1=d_2=d$. Let A_1 and A_2 be two sets with cardinality l_1 and l_2. We have to prove that $M_{A_1,d}$ is the homomorphic image of a subsemigroup of $M_{A_2,d}$.

1. First suppose that $|A_1| \leqq |A_2|$. We may assume without loss of generality that $A_1 \subset A_2$. Thus we get $C_{A_1,d} \subset C_{A_2,d}$. Let us consider the subsemigroup M of $M_{A_2,d}$ defined by

$$M = \langle \Phi | \Phi \in M_{A_2,d} \quad \text{and} \quad c\Phi \in C_{A_1,d} \quad \text{for all} \quad c \in C_{A_1,d} \rangle,$$

and let ϱ be a congruence on M defined by

$$\Phi_1 \varrho \Phi_2 \quad \text{if and only if} \quad c\Phi_1 = c\Phi_2 \quad \text{for any} \quad c \in C_{A_1,d}.$$

It is easy to see that

$$M/\varrho \cong M_{A_1,d}.$$

2. Now assume that $|A_1|>|A_2|$. For the sake of easier perspicuity we prove only in the case $d=2$. The proof for an arbitrary d is similar. On the base of the first part of the proof we may assume that $|A_2|=2$, and we may also assume that $A_2=\langle 0, 1\rangle$. Let n be a positive integer such that $2^{(n-4)^2}>l_1$. Let us subdivide the tessellation array E^2 into square blocks of size $n\times n$. Every block is designated by an element of E^2, as shown in Fig. 1. For any $\alpha\in E^2$, B_α denotes the block designated by α. The subdivided tessellation array will be referred to as block structure of E^2 denoted by E^2_B. It may be considered a tessellation array whose cells are blocks.

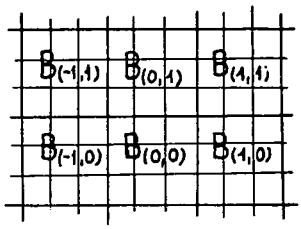


Fig. 1

Let S_{n-4} be a square block of size $(n-4)\times(n-4)$ and let A'_2 be the set of all mappings from S_{n-4} into A_2. (E.g., Fig. 2 shows an element of A'_2). Since $|A'_2|=2^{(n-4)^2}>l_1$, there is a one-to-one mapping $\tau: A_1\to A'_2$. Now we define a one-to-one mapping ϑ: $C_{A_1,2}\to C_{A_2,2}$. For any $c\in C_{A_1,2}$, $c\vartheta\in C_{A_2,2}$ is a configuration whose restriction to an arbitrary block $B_\alpha(\in E^2_B)$, denoted by $c\vartheta/B_\alpha$, is defined by the following way:

1. The restriction of $c\vartheta$ to the inner array of size $(n-4)\times(n-4)$ of the block B_α equals $(c(\alpha))\tau$.

2. Each cell belonging to the outside layer of size 1 of the block B_α contains state 1.

1	0	1
0	0	1
1	0	0

$n-4$

Fig. 2

$n-4$

1	1	1 ... 1	1	1
1	0	0 ... 0	0	1
1	0		0	1
⋮	⋮	$(c(\alpha))\tau$	⋮	⋮
1	0		0	1
1	0	0 ... 0	0	1
1	1	1 ... 1	1	1

n

Fig. 3

3. Each cell belonging to the layer of size 1 around the inner array of size $(n-4)\times(n-4)$ of the block B_α contains state 0 (see Fig. 3).

Let C be the subset of $C_{A_2,2}$ defined by

$$C=\langle c|c=c'\vartheta,\quad c'\in C_{A_1,2}\rangle.$$

(It may be seen from the definition of the elements of C that the block structure of c can be uniquely recognized for all $c\in C$.) Now we associate a mapping $\Psi: C\to C$ with every $\Phi\in M_{A_1,2}$ defined by $c\Psi=((c\vartheta^{-1})\Phi)\vartheta$ for all $c\in C$. Let M be denote the set of such mappings, i.e.,

$$M=\langle\Psi|\Psi=\vartheta^{-1}\Phi\vartheta,\quad\Phi\in M_{A_1,2}\rangle.$$

It is obvious from the definitions that $M \cong M_{A_1,2}$. Thus it is enough to prove that M is a homomorphic image of a subsemigroup of $M_{A_2,2}$. For this it is sufficient to show that for any $\Psi \in M$ there is a $\Psi' \in M_{A_2,2}$ such that the restriction of Ψ' to C is equal to Ψ.

Let $\Psi = \vartheta^{-1}\Phi\vartheta \in M(\Phi \in M_{A_1,2})$ and let $\mathfrak{A} = (A^1, E^2, X, f)\,(X = (\xi_1, \ldots, \xi_n))$ be a CA such that $\Phi_{\mathfrak{A}} = \Phi$. We associate with X a neighbourhood template X_B in E_B^2 defined by $X_B = (B_{\xi_1}, \ldots, B_{\xi_n})$. The sequence $(B_{\alpha+\xi_1}, \ldots, B_{\alpha+\xi_n})\,(\alpha \in E^2)$ is denoted by $N(X_B, B_\alpha)$.

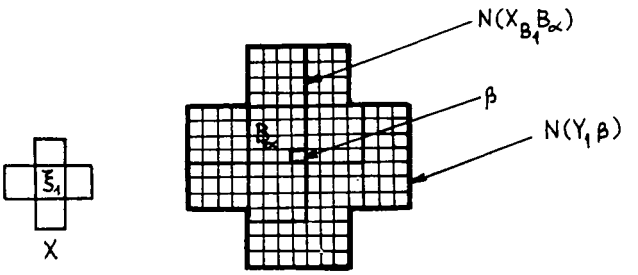


Fig. 4

Now we define a CA $\mathfrak{B} = (A_2, E^2, Y, g)$ such that the restriction of $\Phi_{\mathfrak{B}}$ to C is equal to Ψ.

1. Y is a neighbourhood template for which $N(Y, \beta)$ contains all blocks which belong to $N(X_B, B_\alpha)$ $(B_\alpha \in E_B^2)$ for all $\beta \in B_\alpha$ (see Fig. 4).

2. Let c be an arbitrary element of C. We now show how we may determine $g(c(N(Y, \beta)))$ for any $\beta \in E^2$. Let $\beta \in B_\alpha (\in E_B^2)$. Since $N(Y, \beta)$ contains all blocks belonging to $N(X_B, B_\alpha)$ and the block structure of c can be uniquely recognized we know $c|B_{\alpha+\xi_1}, \ldots, c|B_{\alpha+\xi_n}$ and so we also know $c\vartheta^{-1}(\alpha+\xi_1), \ldots, c\vartheta^{-1}(\alpha+\xi_n)$, i.e., $c\vartheta^{-1}(N(X, \alpha))$. From this we can determine $f(c\vartheta^{-1}(N(X, \alpha))) = ((c\vartheta^{-1})\Phi)(\alpha)$ and $((c\vartheta^{-1})\Phi)\vartheta|B_\alpha = c\Psi|B_\alpha$ as well. But we can uniquely determine the position of β in the block B_α, thus we can also determine $g(c(N(Y, \beta))) = (c\Psi)(\beta)$ as the state contained in the cell β under $c\Psi|B_\alpha$.

Theorem 4. If $d_1 \leqq d_2$, then $M_{l_1,d_1} \times M_{l_2,d_2}$ can be embedded in $M_{l_1 l_2, d_2}$.

Proof. In wiew of the remark at the beginning of the preceding proof we prove the statement for $d_1 = d_2 = d$ only. Let A_1 and A_2 be two sets with cardinalities l_1 and l_2. We have to show that $M_{A_1,d} \times M_{A_2,d}$ can be embedded in $M_{A_1 \times A_2, d}$. Let $c \in C_{A_1 \times A_2, d}$. Since $c(\alpha) = (c_1(\alpha), c_2(\alpha))\,(\alpha \in E^d)$ we may write (c_1, c_2) instead of c. Let $\vartheta: M_{A_1,d} \times M_{A_2,d} \to M_{A_1 \times A_2, d}$ a mapping defined by

$$c(\Phi\vartheta) = (c_1, c_2)\,(\Phi\vartheta) = (c_1\Phi_1, c_2\Phi_2) \quad (c = (c_1, c_2) \in C_{A_1 \times A_2\ d})$$

for all $\Phi = (\Phi_1, \Phi_2) \in M_{A_1,d} \times M_{A_2,d}$. One can easy verify that $\Phi\vartheta$ is a tessellation transformation and ϑ is an isomorphism.

Remark. $M_{l_1,d_1} \times M_{l_2,d_2}$ is not isomorphic to $M_{l_1 l_2, d_2}$ because $Z(M_{l_1 l_2, d_2}) \cong E^{d^2}$, $Z(M_{l_1,d_1} \times M_{l_2,d_2}) \cong E^{d_1+d_2}$ and $E^{d_2} \cong E^{d_1+d_2}$.

For a CA $\mathfrak{A}=(A, E, X, f)$ we shall say that the global transition function $\Phi_{\mathfrak{A}}$ has speed p, if the maximum of the absolute values of the coordinates of the components belonging to X is p, i.e.,

$$p = \max_{\substack{1 \leqq i \leqq n \\ 1 \leqq j \leqq d}} |i_j|$$

where $X=(\xi_1, \ldots, \xi_n)$ and $\xi_i=(i_1, \ldots, i_d)$, $i=1, \ldots, n$.

Lemma 3. Let $C=\langle c_1, \ldots, c_k\rangle$ be a finite set of distinct configurations for which $R_{ij}=\langle\alpha|\alpha\in E^d$ and $c_i(\alpha)\neq c_j(\alpha)\rangle$, $1\leqq i, j\leqq k$, is a finite set and no element of C is a copy of another element of C. For any transformation $\Phi: C\to C$ there exists a tessellation transformation Ψ such that $c_i\Phi=c_i\Psi$, $i=1, \ldots, k$.

Proof. By the assumption $R=\bigcup_{i,j=1}^{k} R_{ij}$ is a finite set. Therefore there is a positive integer p such that R can be included in a d-cube of size $p\times p$.

Let Φ be a transformation of C and let $\mathfrak{A}=(A, E^d, X, f)$ be a CA, where the set of all components of X is equal to the square of size $(2p+1)\times(2p+1)$ with center $(0, \ldots, 0)\in E^d$. Since $N(X, \alpha)$ contains R for all $\alpha\in R$, one can easily define the local transition function f such that the restriction of $\Phi_{\mathfrak{A}}$ to C equals Φ.

Theorem 5. Every finite semigroup is a homomorphic image of a subsemigroup of $M_{1,d}$.

Proof. It is enough to prove the statement for the full transformation semigroup on a finite set with arbitrary cardinality. This is trivial from Lemma 3.

Corollary. Any $M_{1,d}$ generates the variety of all semigroups.

Proof. Indeed, from Theorem 5 it follows that non-trivial identities do not hold on $M_{1,d}$ (for this Corollary consult [3]).

III. Tessellation transformations with a distinquished state

A CA (A, E^d, X, f) is said to be an initial cellular automaton (shortly: ICA), if there is a state $a_0\in A$ called the quiescent state, such that $f(a_0, \ldots, a_0)=a_0$. In this case we shall use the notation (A, a_0, E^d, X, f).

For a set A ($|A|\geqq 2$) and $a_0\in A$, the symbol M_{A,d,a_0} denotes the set of all tessellation transformations in $M_{A,d}$ induced by ICA with quiescent state a_0. It is evident that M_{A,d,a_0} is a subsemigroup of $M_{A,d}$.

A $c\in C_{A,d}$ is said to be a finite configuration if

$$\sup(c) = \langle\alpha|\alpha\in E^d \quad \text{and} \quad c(\alpha)\neq a_0\rangle$$

is a finite set. C_F denotes the set of such configurations. In the sequel, if estimates are given for the number of configurations with some properties we shall not distinquish between two configurations if one is a copy of the other. Clearly, if $\Phi\in M_{A,d,a_0}$ and $c\in C_F$ than $c\Phi\in C_F$.

A $c\in C_F$ is said to be an n-configuration if $\sup(c)$ may be included in a d-cube of size $n\times n$, i.e., there is an $\alpha=(i_1, \ldots, i_d)\in E^d$ such that $c(\beta)=a_0$ for all

$\beta=(j_1, \ldots, j_d)\in E^d$ with $j_k<i_k$ or $j_k>i_k+n$ for at least one k, $1\leqq k\leqq d$. Every $c\in C_F$ is an n-configuration for some n, and if c is an n-configuration then c is also an m-configuration for all $m\geqq n$.

For any set $R\subset E^d$ and $c\in C_{A,d}$, $c|R$ denotes the restriction of c to R. A $c'\in C_F$ is said to be a subconfiguration of $c\in C_F$, if

$$c|\sup(c') = c'|\sup(c').$$

For a $\Phi\in M_{A,d,a_0}$, a $c\in C_F$ will be referred to as image configuration if c may be written in the form $c'\Phi(c'\in C_F)$ and c will be called a Garden-of-Eden configuration if no image configuration containing c as a subconfiguration.

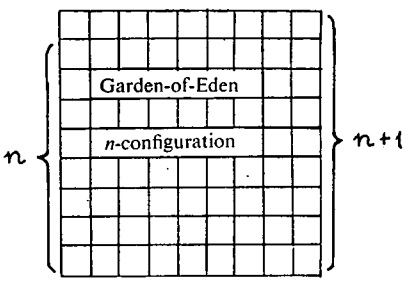


Fig. 5

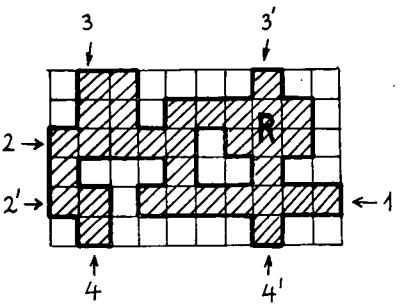


Fig. 6

A compact formulation of Moore's and Myhill's results proved in [5] and [6] (which may be found in [4]) is the following:

Theorem 6. The restriction of a $\Phi\in M_{A,d,a_0}$ to C_F is one-to-one if and only if there exists no Garden-of-Eden configuration.

Let G_n denote the number of Garden-of-Eden n-configurations and let H_n denote the number of all n-configurations

Theorem 7. For any $\Phi\in M_{A,d,a_0}$, $\lim_{n\to\infty} G_n/H_n=0$ or 1 according to whether the restriction of Φ to C_F is one-to-one or not.

Proof. If Φ is one-to-one the assertion is trivial by Theorem 6. Suppose that Φ is not one-to-one on C_F. Let $d=2$, in the case $d\neq 2$ we may proceed similarly. Take $|A|=l$; then $H_n=l^{n^2}$. First we show that G_n/H_n is a monotonous increasing sequence. If e.g. the left lower part of an $(n+1)$-configuration is a Garden-of-Eden n-configuration (as shown Fig. 5), then it is a Garden-of-Eden $(n+1)$-configuration. Using this fact we get $G_{n+1}\geqq G_n l^{2n+1}$. Thus

$$\frac{G_{n+1}}{H_{n+1}} \geqq \frac{G_n\cdot l^{2n+1}}{l^{(n+1)^2}} = \frac{G_n}{l^{n^2}} = \frac{G_n}{H_n},$$

hence G_n/H_n is a monotonous increasing sequence. Thus it is sufficient to prove the statement for a subsequence of G_n/H_n. Let $\bar{G}_n=H_n-G_n$. Since Φ is not one-to-one

on C_F there is a Garden-of-Eden m-configuration for some m. Therefore $\bar{G}_m \leqq (l^{m^2}-1)$ and for any positive integer k, we have $\bar{G}_{km} \leqq (l^{m^2}-1)^{k^2}$. So we obtain that

$$0 \leqq \frac{\bar{G}_{km}}{H_{km}} \leqq \frac{(l^{m^2}-1)^{k^2}}{l^{k^2 m^2}} = \left(1-\frac{1}{l^{m^2}}\right)^{k^2} \to 0 \quad \text{if} \quad k \to \infty.$$

Finally we get

$$\lim_{k\to\infty} \frac{G_{km}}{H_{km}} = \lim_{k\to\infty} \left(1-\frac{\bar{G}_{km}}{H_{km}}\right) = 1.$$

Let us recall some definitions from [2]. Let $R \subset E^2$ be a nonempty finite set. Identifying the positive y axis with the direction north we call the set of cells in R with maximum abscissas the eastern perimeter of R. The northern western and southern perimeters of R are similarly defined. The following cells are called extremal cells of R:

1. The northernmost and southernmost cells in the eastern perimeter of R (cell 1 in Fig. 6).
2. The northernmost and southernmost cells in the western perimeter of R (cells 2, 2′).
3. The westernmost and easternmost cells in the nothern perimeter of R (cells 3, 3′).
4. The westernmost and easternmost cells in the southern perimeter of R (cells 4, 4′).

A function $f\colon A^n \to A$ is said to be cancellative with respect to its i-th variable, if $f(a_1, \ldots, a_{i-1}, a_i, a_{i+1}, \ldots, a_n) = f(a_1, \ldots, a_{i-1}, b, a_{i+1}, \ldots, a_n)$ implies $a_i = b$ for all $a_1, \ldots, a_n, b \in A$.

Theorem 8. If for any *ICA* $\mathfrak{A} = (A, a_0, E^2, X, f)\,(X = (\xi_1, \ldots, \xi_n))$, the local transition function f is cancellative with respect to its i-th variable and ξ_i is an extremal cell of X, then the restriction of $\Phi_{\mathfrak{A}}$ to C_F is one-to-one.

Proof. We may assume without loss of generality that the extremal cell mentioned in the theorem is the northernmost cell in the western perimeter of X. Suppose that c_1 and c_2 are distinct finite configurations. Then

$$R = \langle \alpha \mid \alpha \in E^2 \quad \text{and} \quad c_1(\alpha) \neq c_2(\alpha) \rangle$$

is a nonempty finite set. Let $\beta \in E^2$ be a cell such that the northernmost cell in the western perimeter of $N(X, \beta)$ and the southernmost cell in the eastern perimeter of R are equal to each other. Using the cancellativity of f we get

$$f\big(c_1(N(X, \beta))\big) \neq f\big(c_2(N(X, \beta))\big), \quad \text{i.e.,} \quad c_1 \Phi_{\mathfrak{A}} \neq c_2 \Phi_{\mathfrak{A}}.$$

The converse of Theorem 8 fails trivially. It may be expected, however, that if we restrict ourselves to considering local transition functions depending essentially on all variables associated with extremal cells of X, then the assumption is also necessary. The next counter-example shows that this is not true.

Let us consider two *ICA*: $\mathfrak{A}^{(1)} = (\langle 0,1 \rangle, 0, E^1, (-1, 0, 1), f^{(1)})$ and $\mathfrak{A}^{(2)} = (\langle 0,1 \rangle, 0, E^1, (-1, 0, 1), f^{(2)})$, where $f^{(1)}$ and $f^{(2)}$ are defined by Table 2. The restrictions of

Table 1.

x_1	x_2	x_3	x_4	x_5	$f(x_1, x_2, x_3, x_4, x_5)$
0	0	0	0	0	0
0	0	0	0	1	0
0	0	0	1	0	1
0	0	0	1	1	0
0	0	1	0	0	0
0	0	1	0	1	0
0	0	1	1	0	0
0	0	1	1	1	1
0	1	0	0	0	0
0	1	0	0	1	0
0	1	0	1	0	1
0	1	0	1	1	0
0	1	1	0	0	1
0	1	1	0	1	1
0	1	1	1	0	1
0	1	1	1	1	0

x_1	x_2	x_3	x_4	x_5	$f(x_1, x_2, x_3, x_4, x_5)$
1	0	0	0	0	0
1	0	0	0	1	0
1	0	0	1	0	1
1	0	0	1	1	0
1	0	1	0	0	1
1	0	1	0	1	1
1	0	1	1	0	0
1	0	1	1	1	1
1	1	0	0	0	1
1	1	0	0	1	1
1	1	0	1	0	0
1	1	0	1	1	1
1	1	1	0	0	1
1	1	1	0	1	1
1	1	1	1	0	0
1	1	1	1	1	1

$\Phi_{\mathfrak{A}^{(1)}}$ and $\Phi_{\mathfrak{A}^{(2)}}$ to C_F are one-to-one, because $f^{(1)}$ (resp. $f^{(2)}$) is cancellative with respect to its first (resp. third) variable. Then $\mathfrak{A}=(\langle 0,1\rangle, 0, E^1, (-2,-1,0,1,2), f)$, where f is defined by Table 1 is the *ICA* whose global transition function $\Phi_{\mathfrak{A}}$ is equals to $\Phi_{\mathfrak{A}^{(1)}}\Phi_{\mathfrak{A}^{(2)}}$.

It can be seen that f depends on its first and fifth variables, but f is not cancellative with respect to any of them. Thus the restriction of $\Phi_{\mathfrak{A}}$ to C_F is one-to-one, but $\mathfrak{A}$ does not fulfil the assumptions of Theorem 8.

Lemma 4. If for an *ICA* $\mathfrak{A}=(A, a_0, E^d, X, f)$ (with n-ary f) the restriction of $\Phi_{\mathfrak{A}}$ to C_F is one-to-one, then all the classes of the partition $A^n/f\circ f^{-1}$ (partition on A^n induced by f) have the same cardinality.

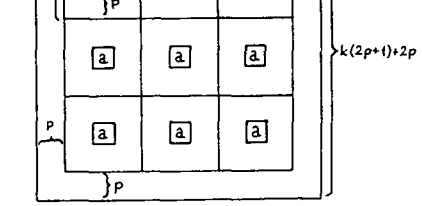


Fig. 7

Proof. Again we shall prove the statement for $d=2$ only. Take $|A|=l$. Suppose that $\Phi_{\mathfrak{A}}$ has speed p. Then X can be included in a square of size $(2p+1)\times(2p+1)$ with center $(0,0)\in E^2$. We may assume without loss of generality that this square equals to X. In this case $n=(2p+1)^2$. If there are classes of $A^n/f\circ f^{-1}$ with different cardinalities, then there is an $a\in A$ such that $|af^{-1}|\geqq l^{n-1}+1$. Let k be an arbitrary positive integer and let D_k denote the number of all $(k(2p+1)+2p)$-configuration which have $a\in A$ in the cells shown in Fig. 7. We have $D_k=l^{(k(2p+1)+2p)^2-k^2}$. Because $\Phi_{\mathfrak{A}}$ has speed p, $c\Phi_{\mathfrak{A}}$ is an $(m+2p)$-configuration for any m-configuration c. Since $\Phi_{\mathfrak{A}}$ is one-to-one with respect to C_F and $|af^{-1}|\geqq l^{n-1}+1$ $(n=(2p+1)^2)$ we obtain that D_k is at least

$(l^{(2p+1)^2-1}+1)^{k^2}$. Thus we get

$$(l^{(2p+1)^2-1}+1)^{k^2} \leqq l^{(k(2p+1)+2p)^2-k^2},$$

whence

$$1+\frac{1}{l^{(2p+1)^2-1}} \leqq l^{\frac{4p(2p+1)}{k}+\frac{4p^2}{k^2}},$$

which is not true for all k.

Remark. The converse of Lemma 4 is not true. One can easily verify that the next *ICA* is a counter-example: $\mathfrak{A}=(\langle 0,1\rangle, 0, E^1, (-1,0,1), f^{(3)})$, where $f^{(3)}$ is defined by Table 2.

For a fixed neighbourhood template X and a state alphabet A ($|A|=n$) with a quiescent state $a_0 \in A$, the symbol K_n denotes the number of all tessellation transformations induced by *ICA* (A, a_0, E^d, X, f) whose restrictions to C_F are one-to-one. S_n will denote the number of all tessellation transformations induced by *ICA* (A, a_0, E^d, X, f).

Theorem 9. $\lim_{n\to\infty} K_n/S_n = 0$.

Proof. If X has $k(\geqq 2)$ components, then $S_n = n^{(n^k-1)}$ and using the result of Lemma 4 we get

$$K_n \leqq \frac{(n^k-1)!}{(n^{k-1}!)^{n-1}(n^{k-1}-1)!} \quad (k \geqq 2).$$

Thus

$$\frac{K_n}{S_n} \leqq \frac{(n^k-1)!}{n^{(n^k-1)}(n^{k-1}!)^{n-1}(n^{k-1}-1)!} = \frac{n^k!}{n^{n^k}(n^{k-1}!)^n} \quad (k \geqq 2).$$

Using Stirling's formula $\left(n! = \left(\frac{n}{e}\right)^n \sqrt{2\pi n} \cdot e^{\frac{\Theta_n}{12n}},\ 0 < \Theta_n < 1\right)$ we get

$$\lim_{n\to\infty} \frac{n^k!}{n^{n^k}(n^{k-1}!)^n} = \lim_{n\to\infty} \frac{\left(\frac{n^k}{e}\right)^{n^k} \sqrt{2\pi n^k} \cdot e^{\frac{\Theta_{n^k}}{12n^k}}}{n^{n^k}\left(\left(\frac{n^{k-1}}{e}\right)^{n^{k-1}} \sqrt{2\pi n^{k-1}} \cdot e^{\frac{\Theta_{n^{k-1}}}{12n^{k-1}}}\right)^n} =$$

$$= \lim_{n\to\infty} \sqrt{\frac{1}{(2\pi)^{n-1} n^{n(k-1)-k}}} \cdot \lim_{n\to\infty} e^{\frac{\Theta_{n^k}}{12n^k} - \frac{\Theta_{n^{k-1}}}{12n^{k-2}}} = 0 \cdot 1 = 0 \quad (k \geqq 2).$$

If $k=1$ then $S_n = n^{n-1}$ and $K_n \leqq (n-1)!$. Thus

$$\frac{K_n}{S_n} \leqq \frac{(n-1)!}{n^{n-1}} \to 0, \quad \text{if} \quad n \to \infty,$$

In the course of a conversation B. Csákány conjectured that for fixed A, d and a_0, the set S of all bijetive transformations of C_F induced by *ICA* form a group. We shall prove that this conjecture is false by giving an example which shows that S is not closed under the formation of inverses.

Let $\mathfrak{A}=(\langle 0, 1\rangle, 0, E^1, (-1, 0, 1), f^{(1)})$ be an *ICA*, where $f^{(1)}$ is defined by Table 2. The restriction of $\Phi_{\mathfrak{A}}$ to C_F is surjective (shown by Amoroso and Cooper in [3] pp. 163) and thus it is also bijective. Suppose that there is a $\Psi \in M_{\langle 0,1\rangle,1,0}$ such that $(c\Phi_{\mathfrak{A}})\Psi = c$ for all $c \in C_F$, and let $\mathfrak{B}=(\langle 0, 1\rangle, 0, E^1, X, f)$ be an *ICA* such that $\Phi_{\mathfrak{B}}=\Psi$. Assume that $\Phi_{\mathfrak{B}}$ has speed p. In this case $(c\Phi_{\mathfrak{B}})(\alpha)(\alpha \in E^1)$ is uniquely determined by $c(\alpha-p), \ldots, c(\alpha), \ldots, c(\alpha+p)$ for any configuration c. Let us consider the following two configurations and their image configurations under $\Phi_{\mathfrak{A}}$ (see. Fig. 8).

Table 2.

x_1	x_3	x_3	$f^{(1)}(x_1, x_2, x_3)$	$f^{(2)}(x_1, x_2, x_3)$	$f^{(3)}(x_1, x_2, x_3)$
0	0	0	0	0	0
0	0	1	0	1	0
0	1	0	1	1	0
0	1	1	0	0	1
1	0	0	1	0	0
1	0	1	1	1	1
1	1	0	0	0	1
1	1	1	1	1	1

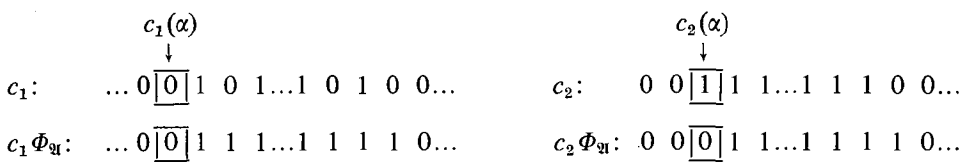


Fig. 8

It can be seen that $(c_1\Phi_{\mathfrak{A}})(\beta)=(c_2\Phi_{\mathfrak{A}})(\beta)$, $\alpha-p \leqq \beta \leqq \alpha+p$, but $((c_1\Phi_{\mathfrak{A}})\Phi_{\mathfrak{B}})(\alpha)=c_1(\alpha) \neq c_2(\alpha)=((c_2\Phi_{\mathfrak{A}})\Phi_{\mathfrak{B}})(\alpha)$, which is a contradiction.

Мозаичные преобразования

Исследуется зависимость строения полугрупп мозаичных преобразований от размерности пространства-носителя, а также от числа состояний. Далее, рассматриваются преобразования конфигураций, индуцированные мозаичными автоматами с состоянием покоя. Отвечая на вопрос, поставленный Э. Ф. Муром доказывается, что почти все мозаичные автоматы обладают взаимно стираемыми конфигурациями.

References

[1] AMOROSO, S. and G. COOPER, The Garden-of-Eden theorem for finite configurations, *Proc. Amer. Math. Soc.*, v. 26, 1970, pp. 158—164.
[2] CODD, E. F., *Cellular automata*, Academie Press, Inc., New York and London, 1968.
[3] COHN, P. M., *Universal algebra*, Harper and Row, New York, 1965.
[4] CSÁKÁNY, B., *Algebra* (lecture notes in Hungarian), Tankönyvkiadó, 1973.
[5] MOORE, E. F., Machine models of self-reproduction, *Proc. Symp. Appl. Math.*, v. 14, 1962, pp. 17—33.
[6] MYHILL, J., The converse of Moore's Garden-of-Eden theorem, *Proc. Amer. Math. Soc.*, v. 14, 1963, pp. 685—686.
[7] YAMADA, H. and S. AMOROSO, Tessellation automata, *Information and Control*, v. 14, 1969, pp. 299—317.

(Received August 16, 1973)

Logical foundations for a general theory of systems

By T. GERGELY and I. NÉMETI

1. Introduction

Science has now reached a turning-point in its development at which it is becoming increasingly urgent for us to achieve a systematization and reduction to some common denominator of the vast body of knowledge that has been accumulated in its various disciplines. The need is to construct a unifying theory with the capacity to override the barriers which at present divide and compartmentalize specialist investigations in order that we may be able to free the disciplines to interact with and reinforce one another. While unification on this scale necessarily entails raising the level of abstraction of the theories with which each discipline works, the language and concepts of this higher, more general level must retain the precision and explicitness in their interrelations that are found at their lower level. These are the objectives and constraints which a General Systems Theory (GST) must fulfil.

More exactly, the features that a General Systems Theory must display are the following:

— it must give a method in which the structural and functional aspects of the system* form a dialectical unity;

— it should be extendable in order to cater for any new aspects that may emerge in the future;

— it should be general only to the extent that it does not lose the property of reversibility, that is, of applicability to the disciplines on which it is founded;

— it should provide not only an approach but also a method of analysis whereby a sufficient body of knowledge can be acquired to permit intervention in the system (e.g. to control it);

— it should provide a mathematical apparatus which enables only the investigation and analysis but also the synthesis of systems.

For the time being there is no theory that completely satisfies all these requirements, a deficiency, which means that on many occasions only verbal methods are at our disposal. A verbal theory of systems too — essentially an initial stage of the

* The basic ideas of Systems Theory are assumed to be known already and so they are not defined here; the particular terminology with which they are expressed is not of importance for the present purposes.

GST — can give considerable help in the analysis of systems (biological, economic, social, etc.), provided it satisfies certain logical conditions to be discussed later. Nevertheless there are already certain systems theories, oriented to particular branches of science which satisfy the requirements of the last two points and among these cybernetics has achieved the most significant results. Indeed on the strength of this some branches of cybernetics (e.g. information theory, automata theory) have proclaimed themselves the base of the GST, though they give only some aspects of the GST, and their applicability depends upon the concrete problem. In problem solving, it is essential to select the level of abstraction adequate to the problem.

Let us consider these ideas more closely:

The interpretations of GST, the various specific systems theories (SST) correspond to the different types of system (and not to the different types of theoretical apparatus).

Theories of biological, psychological, economic, technical systems etc. are instances of Specific Systems Theories. Within a given SST — depending on the apparatus applied — different aspects can be distinguished, for instance, aspects of information theory, automata theory, control theory, etc. Independently of this, it is possible to distinguish various levels of abstraction within an SST. This concept can be illustrated in the following way.

On the highest level of abstraction we are restricted to the study of the relations between the system and its environment, the system being regarded as a single indifferentiated unit or a *black box*. On the second level the system may be broken down to its immediate component parts, that is, the system is analysed as the ensemble of its immediate subsystems. On the lower levels the system is seen in a more and more detailed analysis down to an apparently arbitrary degree of refinement, to whatever number of levels of abstractions we wish. The selected level of abstraction can said to be adequate to the task, if it allows to solve the problem with the minimum effort (i.e. without having to go into unnecessary details). Naturally we can speak of the adequacy of the level of abstractions only if the aspect selected is appropriate. The concepts of level and aspect are orthogonal to one other; that is to say, any level may be combined with any aspect.* To sum up, the main tasks that must be in developing and applying a GST are:

— elaboration of a mathematical base that allows the GST to satisfy the above four points;

— the elaboration of a method of applying the GST on the appropriate level of abstraction.

To these ends it is necessary first to create the logical foundations *a)* for establishing the GST; *b)* for applying the GST.

After the logical foundations have been laid it will become possible to create the theory itself. The logical foundations can of course only provide guide-lines for the creation of the theory, but then it is hardly conceivable that the GST could be built up in a single step.

Properly based logical foundations are necessary to ensure that the set of

* Selection of the adequate level and aspect is inseparable from the process of problem solving (optimalization). In its course the level and the aspect are modified alternately, until the ensemble of the two suits the given purpose.

SST-s is such that the GST which is built on it forms a dialectical unity with the SST-s themselves.

The aim of the present work is to clarify the logical foundations needed to establish the GST, taking Curry's book [1] as a starting point. The logical foundations for the application of the GST will be dealt with in a subsequent paper.

2. The language

As in building a theory the first steps are taken with the aid of natural language, it is necessary to start with the analysis of the natural language. After the basic definitions have been set out we can go on to examine the relation of language to the sphere of phenomena and then the manner in which a formal language can be produced.

From the semantical point of view any language, including a natural language, is produced by the combination of *nouns**, *statements* and *functors* (phrases).

The set of nouns will be denoted N, the set of statements E. Both sets are inductive classes** and their definitions — see later — are given by a grammar employing the auxiliary concept of functors.

Functors. Transformers with one or more arguments. The argument may be: noun, statement, functor. The value of the functor (result) may be: noun, statement, functor.

Generally all combinations are possible, which means that nine main functor-types can be distinguished. It is conceivable that a specification defining the inductive class (N) of the objects (nouns) uses all nine functor types. In the specification of N the concepts of "statement" and "functor" are auxiliary ideas.

The set E (i.e. the set of statements to be generated from the nouns and functors) is called the language. The aggregate of the (inductive) laws defining set E is called the grammar (for instance the set of nouns and functors together with the rules for their combination).

The sphere of phenomena (J) is taken here to mean the entirety of research objects. We start from the supposition that the properties of the individual objects to be analysed are recognised in the course of their interaction with other objects.

Let be J a space (set). The relation between the elements of J and the elements of N the set of some nouns gives the representation of the objects in the sphere of phenomena in language: for short we shall call this the $J \rightarrow N$ representation.

Let us look at a basic property of this representation. Let us constitute the product set $J \times N$ and map it on the closed interval $[0, 1]$. $\mu: J \times N \rightarrow [0, 1]$. The value $\mu(j, n)$ will be called *the measure of validity* of the $j-n$ relation. In other words, the value $\mu(j, n)$ specifies the degree to which noun n is congruent with the phenomenon j.

Set N is a precise representation of the sphere of phenomena only if the function μ has the values 0 or 1. It is called an imprecise representation if μ takes away value between 0 and 1.

* *Noun* denotes an expression specifying some object and corresponds to the linguistic concept *noun-phrase*. (Note that a noun may be composed of more than one word.)

** A definition of the inductive class is given later.

An imprecise function μ can be made to correspond to a precise function μ_k by giving the threshold value $0 < k \leqq 1$, where μ_k is defined as $\mu_k(j, n) = 1 \Leftrightarrow \mu(j, n) \geqq k$. Accordingly, any precise representation may be coordinated to imprecise representation by giving the threshold limit k.

The precise function μ_1 is called the kernel of the imprecise function μ, and its representation is called the precise kernel of the original imprecise representation.

Note: if N proves to be precise then the $J \rightarrow N$ relationship can also be represented as a relation.

The precision of the $J \rightarrow N$ representation depends upon the extent to which the structure of objects in the sphere of phenomena (J) corresponds to, or how homomorphic* it is to with the structure of the set N which is determined by the grammar.

As already explained, the set N and a grammar G together define a set of expressions (E), called the language. The relationship between the elements of J and E comprises the linguistic representation of the sphere of phenomena, in short the $J \rightarrow E$ representation.

Note, that if N happens to be imprecise, so too will E**. This is also true in a more general way: as it was shown, E is built on N, but we shall see later that a certain T is built on E, T_1 on T, and so on.

It therefore holds for this arbitrarily long series that if any of its terms becomes imprecise, the imprecision will be transferred to all terms that are constructed directly or indirectly upon the term in question. More exactly, the imprecision will not be transferred, provided only the precise parts (kernels) are employed in construction.

The language E is a means of describing our experiences connected with the phenomenal sphere.

The criterion of truth of a statement $e \in E$ is found in the sphere of phenomena. Let us map the set E by the function γ in to the closed interval [0, 1] of the numerical axis. $\gamma: E \rightarrow [0, 1]$. $\gamma(e)$ is now the *measure of truth* of the statement e.

It should be noted that the value $\gamma(e)$ can also be interpreted as the aprobability of the statement being true. The measure of validity μ in contrast does not admit such an interpretation. Values of the function γ lying between 0 and 1 will have significance in complete theories (in the Gödel), when we are unable to demonstrate or refute certain statements but can render them probable (and likewise their consequences) by repeated heuristic attempts. If, for instance a true but indemonstrable statement is in question, the mesure of truth of that statement converges to 1.

A subset (L) of the set E to the elements of which values of the function γ have been given is called a description.

The description L is said to be precise if none of the γ values pertaining to its elements lies between 0 and 1. Similarly to the imprecise μ function, an imprecise function γ can be coordinated with a precise function γ_k by giving a threshold value k. The function γ_1 gives the kernel of the description (L_1). If L is precise, then $L = L_1$.

* See later.

** The grammar also may be imprecise, as when we have a sentence the grammatical correctness of which lies between zero and one [2]: if G is imprecise, then plainly E will be imprecise too.

3. The theory and its interpretation

The results of completed observations can be recorded by the aid of some description L. However, L is not capable of predicting an event that will be observed in the future, based on the results of the completed observations. In order to permit predictions we must introduce the concept of rules of inference.

Rules of inference serve to produce from given true statements new, true statements.

A rule of inference r is a $n+1$ place relation defined on the set $E(r \subset E^{n+1})$ by means of which a new, true statement can be obtained from n true statements; that is, r: $E^n \to E$ or $(e_1, e_2, \ldots, e_n) \xrightarrow{r} e$. The set of rules of inference is denoted by R.

Where there is imprecision after applying the rule of inference, the measure of truth of the inferred statement is obtained from the measure of truth of the initial statements.

The possibility also exists — at least in principle — of obtaining a precise inferential statement from a large but finite number of imprecise statements.

Definition 1. We define a *theory* as a description $(T \subseteqq E)$ and a set of rules of inference (R) whereby T is *closed* under R; that is $(\forall r \in R) r(T^n) \subseteqq T$. For the sake of simplicity T will be called a theory, though the existence of some R is to be understood the same time.

T is said to be *consistent* if $T \neq E$.

If T is based on an imprecise description then it is called an imprecise theory.* It is evident that the above task can be solved by a carefully selected T theory.

However there is a need for a communication between some sphere of phenomena and the theory appropriate for its analysis, on the one hand for theoretical prediction of experimental results and, on the other hand for development of the theory based on the experimental results.

Definition 2. A theory T is said to be *extendable* within the language E, if there is some $K \subseteqq E$ for which $T' = T \cup K$ is consistent and is closed under the rules of inference

$$T' \neq E \quad \& \quad (\forall r \in R) r(T'^{n_r}) \subseteqq T'.$$

If the theory is to be extended through extension of the language E, means must be provided for, ensuring the extendability of the measure of truth.

Definition 3. A description L_1 is said to be an interpretation of the theory T_2 if there exists a mapping i: $E_2 \to E_1$ such that $L_1 \subset E_1$ and $T_2 \subset E_2$ but no stipulation is made that E_2 should differ from E_1 nor that i should be defined for every element of E_2.

The interpretation is *complete* if it is everywhere defined in E_2. The interpretation is *valid,* if $L_1 \supseteqq i(T_2)$. The interpretation is *adequate*, if $L_1 \subseteqq i(T_2)$.

* The condition of consistence for imprecise theories is $(\forall t \in T) \gamma(t) + \gamma(\neg t) = 1$. In general *every condition* defined on T is true in the case of imprecision provided the condition holds for T_k with an arbitrary k. Note that, in predicate calculus, the definition of consistence $(T \neq E)$ is equivalent to the following condition $t \in T \Leftrightarrow \neg t \notin T$ [1, 3].

If the interpretation is valid and L_1 is a theory, then T_2 can also be regarded as an extension of L_1.

Note: A theory may be have more than one interpretation.

Any theory T may be formalized by formalizing the language E (the statement of which are now obtained with the aid of *formal objects* and *predicates* in such a way that formal objects or statements are substituted for the arguments of the predicates) and the formal notation of the rules R. Nevertheless, no particular use can be made of this formalization unless T forms an inductive class, that is, unless the set cannot be generated finitely by means of R.

Definition 4. A theory T is said to be *deductive* if it contains such a finite subset A from which any element of T can be derived by a finite number of repeated applications of the rules R. T is then said to be generated by A with the aid of R, and the set A is called the set of *axioms* of the theory.

If T is deductive and extendable, then an extension of T always follows from the extension of A. (This of course refers only to extensions within the language E.)

When a deductive theory is formalized, the product is a *formal theory*.

In the following we present the concepts needed in the definition of a formal theory.

An *inductive class* is an enumerable set (E) which is defined by an algorithm.* This algorithm is called the (constructive) *specification* of the inductive class E. By this we mean that e is an element of E, iff there exists a finite n such that e will be produced by the algorithm in n steps.

In detail: By an inductive class we mean an inductively defined set. By an inductive definition we mean the listing of a finite number of laws and statements by a finite number of applications of which any element of the set can be formed.

a) Thus we can list a finite number of statements

e.g.,

$$a, b, c \in A_1$$
$$\alpha, \beta \in A_2$$
$$\vdots$$
$$b, c, \beta \in A_8$$
$$\vdots$$

b) We can also list a finite number of rules. We shall do this in terms of *variables* X and Y. (In using the rules anything can be substituted for the variables.)

$$X, Y \in A_1 \Rightarrow XYX \in A_1$$
$$X \in A_1 \quad \& \quad Y \in A_2 \Rightarrow YYX \in A_3$$
$$X, Y \in A_8 \Rightarrow XY \in A_8$$

c) We can specify the set to be defined e.g., $I = A_8 \cap A_3$.

Thus, for instance, $bccbccb \in I$.

* By algorithm a constructive procedure in the Hilbertien sense is meant, that is, a specification which unambigously defines the series of transformations to be executed on some objects. This series of transformations may be either finite or indefinite, but the specification must naturally always be finite. The execution of a single transformation is called a step, and the stipulation is made that the transformations should be realizable.

Note: This is usually called a closure condition; the elements defined above for instance are the elements of the set I only if they are elements of A_8 and A_3.

Definition 5. The minimum *base* of a language is defined as the finite set of some symbols from which the language can be generated. By a *string* we mean a series of the elements of a base (a string is therefore a series of symbols).

Definition 6. The number of the base elements included in the string x is termed the *string length* and will be denoted by $|X|$. A string of zero length is denoted by λ.

Let A, B and X be a set of strings of symbols. By a language we mean a set or strings on which the following operations hold true:

1. $$AB \stackrel{\text{def}}{=} \{ab | a \in A, b \in B\}$$

2. $$X^n \stackrel{\text{def}}{=} X X^{n-1}$$
$$X^0 \stackrel{\text{def}}{=} \{\lambda\}$$

Note: If no set of strings is involved then X^n represent a Cartesian product.

3. $$X^* \stackrel{\text{def}}{=} \bigcup_{i=0}^{\infty} X^i \stackrel{\text{def}}{=} X^0 \cup X \cup X^2 \cup \ldots$$

If X is a base, then X^* is the set of strings definable on the base X.

4. The formal theory

Let introduce the following concepts:

Definition 7. The set of *abstract objects* is denoted by O. The inductive class of strings defined on some base K is $(O \subset K^*)$.

Set of predicates $(F \subset G^*)$. This is an inductive class of strings defines on a base G, on which some mapping r: $F \to N$ is defined where N is the set of natural numbers. An $r(\varphi)$ value allocated to some predicate $\varphi \in F$ is called the order of the predicate (also denoted r_φ).

Note: $K \cap G = \emptyset$.

Set of statements (E)

$$E \stackrel{\text{def}}{=} \{\varphi(\{,\}O)^{r\varphi} | \varphi \in F\}$$

E.g., $$\varphi O_1, O_2, \ldots, O_{r_\varphi} \in E$$

Definition 8. A *formal theory* consists of the following components:

a) Inductive class of abstract objects (O).
b) Inductive class of predicates (F).
c) Set of statements (E). (This is produced from the first two.)
d) Inductive class of true statements $(T \subseteq E)$.

To get T the following are needed: 1) A finite subset of T, called the set of axioms (A). 2) A set of rules of inference (R), giving together with A the inductive definition of T.

The set E is called a formal language. (It is on this that the theory T is defined.) The specification of the inductive class E is called the grammar. The grammar consists of the specification of the inductive classes O and F and of the laws of substitution for the arguments of the predicates.

Quite obviously the properties of the verbal theories defined so far (for instance, extensibility, imprecision, etc.) refer by definition to formal theories too.

In formalizing a verbal theory T_V, a formal theory T_F is sought of which T_V is a valid, adequate interpretation.

From the foregoing it follows that, since every theory can be formed as the valid, adequate interpretation of a deductive extensible theory, every theory can be formalized in the above sense.

The function γ is defined also to the set E_F, thus the imprecision of the formal theory T_F can be handled similarly to the precision of the verbal theory T_V. For this purpose the rules of inference must be defined accordingly. Since a formal theory in the above form is a syntactic system, it is necessary to define the precision of the syntax and the concept of an imprecise syntax. A simple example of this is given by Zadeh [2].

Let us see how a natural language E_V can be formalized, in other words, how a formal language E_F can be made to correspond to a language E_V in such a way that the correspondence between the statements should be bijective and also isomorphic. The establishment of these conditions will be symbolized by $E_V \cong E_F$. In many cases it is sufficient that the relation be homomorphic, for which we use the notation $E_V \simeq E_F$.

Let us now look at what is meant by homomorphic and isomorphic correspondence. Let E and E' be two formal languages to which belong the factors O, O'; F, F', etc.

Definition 9. The mapping $i: E \to E'$ is said to be homomorphic (denoted $i: E \simeq E'$) if the mappings* $i: O \to O'$ and $i: F \to F'$ — with regard to the structure of the inductive classes — are homomorphisms,** and the condition $i(\varphi O_1 \ldots O_n) = i(\varphi) i(O_1), \ldots, i(O_n)$ holds.

The relation is isomorphic if its inverse is also homomorphic: i.e. $E \cong E' \Leftrightarrow E \simeq E' \& E' \simeq E$. Where imprecision is encounted it is necessary to define the *measure* of homomorphism.

It should be remembered that, while the grammar of some formal language E_F consists of an inductive class of abstract objects (O) and an inductive class of predicates (F), the grammar of a natural language E_V consists of the inductive class of nouns (N) and the inductive class of functors.

A formal language E_F can be made to correspond to the language E_V be the following two steps:

* The mappings $i: O \to O'$, etc. are taken to mean those correspondences between objects, etc. forming the base of the correspondence between the sentences $i: E \to E'$.

** The mapping $i: O \to O'$ is homomorphic if, when forming the elements of the sets there is no difference between making an inductive step first followed by interpretation and making the interpretation before the corresponding step. For instance let $O_1, O_2, \ldots O_n \in O$ and $\omega O_1 \ldots O_n \in O$, where O is one of the inductive steps used in the definition. If $i(\omega O_1, \ldots O_n) = i(\omega) i(O_1) \ldots i(O_n)$ is satisfied generally, then the correspondence is homomorphic.

1. By making the inductive class O congruent with the inductive class N. The measure of isomorphism between the sets of N and O with respect to the inductive definition gives the precision of the congruence. (It is conceivable that all the functor types of the language to be defined have to be considered when O is defined.)

2. By making the set of predicates congruent with the potentialities ensuing from the use of functors (singly, combined or repeated) in obtaining statements from objects.* If arbitrary combinations of the functors are allowed in the language being formalized, then the set of predicates will be an inductive class.

If the formal language E_F proves to be a properly selected formalization of the language E_V, then it is possible to find for a verbal theory $T_V \subset E_V$ a consistent theory $T_F \subset E_F$ in such a way that T_V is valid, adequate interpretation of the formal theory T_F.

The congruence i: $E_F \simeq E_V$ (giving the interpretation i: $T_F \simeq T_V$) may also be imprecise in which case it is necessary to define a function μ: $E_F \times E_V \to [0, 1]$ where the value $\mu(e_F, e_V)$ gives the measure of validity of the congruence $e_F \to e_V$, e_F and e_V being statements from the theories T_F and T_V ($e_F \to e_V$ means that e_V is the interpretation of e).

Imprecision in a formal theory T_F can derive from the theory's own logical structure, from imprecision in the relation between T_F and T_V, from the structure of T_V, or from imprecision in the relation $T_V - J$. (And if a theory T_F' is constructed of which T_F is an interpretation, this chain is continued.)

5. Simple examples of a formal theory

I. 1. formal objects: $O = \{a, b, c\}$

2. predicates: let $o \in O$,

$$\varphi_1 o,$$

$$\varphi_2 o,$$

$$\varphi_3 o.$$

3. rules of inference:

$$\varphi_1 X \Rightarrow \varphi_1 Xb,$$

$$\varphi_1 X \& \varphi_1 Y \Rightarrow \varphi_2 XcY,$$

$$\varphi_3 XcY \Rightarrow \varphi_3 XbcYb.$$

(X and Y are used here as variables; that is, any object can be substituted for them.)

4. axioms:

$$\varphi_1 a,$$

$$\varphi_3 aca.$$

Interpretation. The object of the form $abb...b$ corresponds to a natural number the value of which is the length of the series. Object c corresponds to the sign of

* Note that this represents only a fraction of the possibilities presented by the functors.

equality. Predicate $\varphi_1 o$ corresponds to the statement o is a number; $\varphi_2 o$ denotes that o is a statement and $\varphi_3 o$ that o is a true statement (theorem).

II. 1. set of formal objects O, starting object: $a \in O$
law of generation: $X \in O \Rightarrow Xb \in O$, or in short $X \Rightarrow Xb$.

2. predicates: let $o_1, o_2, \xi O$,
$$\varphi o_1 o.$$
3. rules of inference:
$$\varphi XY \Rightarrow \varphi Xb Yb.$$
4. axioms:
$$\varphi aa.$$

Interpretation. O corresponds to the set of natural numbers, while $\varphi o_1 o_2$ means that the number corresponding to o_1 equals the number corresponding to o_2.

6. Epitheory

6.1. Each statement e which refers to the theory T but for which $e \notin T$ is an *epistatement* of the theory T. By an epitheorem is meant a true epistatement, and by the set of epistatements, an epitheory. The construction of an epitheory will be examined here by means of an extension of the theory.

An *extension* of a theory T is a procedure involving the complete (or partial) execution of the following steps:

1. Extension of language E to language $E' \supset E$ (by the addition of new statements).
   a) Extension of the set of objects to the set $O' \supset O$.
   b) Extension of the set of predicates to set $F' \supset F$.

2. Extension of theory T to the theory $T' \supset T$.
   a) Extension of the set of axioms to the set $A' \supset A$.
   b) Extension of the set of rules of inference to the set $R' \supset R$.

From here onwards, after the introduction of each type of epitheory it will be shown how it is reducible to the concept of extension defined above. Conversely, all general statements made about extensions.

In each of the above steps (in the case of imprecise theories) care must be taken that the measure of truth be continuable to new statements of the extended language. (In precise cases we are careful to ensure that the extended theory remains consistent.)

Let us look more closely at the various possible kinds of extension.

6.2. Inductive extension. Suppose we have several theories $T_1, T_2, \ldots, T_n$. These theories — in the course of development — may become so complicated that the need arises to achieve some clarification on a higher abstraction level. For this purpose those theorie should be selected (let us say $T_1, \ldots, T_m$) for which the following procedure seems to be efficiently performable. The resultant theory T we shall form in the way $T = T_1 \cup T_2 \cup \ldots \cup T_m$. Upon this we can define an

equivalence relation the equivalence classes pertaining to which give the elements of the theory T'. The theories $T_1, \ldots, T_m$ are now interpretations of the theory T'.

In carrying out this abstraction cover* could be used instead of partition,* but is this case special attention must be paid to the continuability of the measure of truth. (The above process will vary in different cases, depending on the aims and aspects that must be considered. For instance, contradictory aspects would be an increase in m, minimalization of the complexity of T' and maximalization of the useful information content of the specific $T' \rightarrow T_i$ interpretations.)

This process can be reduced to extension inasmuch as when we have formed theory $\hat{T}$ (defined in the language $\hat{E}=E_1 \cup \ldots \cup E_m$), we then extend it to a theory $T \cup T'$ (by extension of the language to $E \cup E'$) and admit appropriate statements (predicates with two arguments) to fix the interpretational relations $T' \rightarrow T_i$.

A particular case of the above is when, in looking for a more abstract theory T' (with the interpretation $T' \rightarrow T$) for a single theory T, a third theory is employed. Instances of this sort are encountered when a theory is being developed in interaction with the sphere of phenomena. Through study of a certain sphere of phenomena we may come across new facts, and these will be inclued in the theory as new axioms extension of type (2/*b*); or we may discover new objects and new ideas (1/*a*); or we may find the need to formulate new kinds of statements (1/*b*); and these result in the extension of the language, etc. These instances all correspond to the case described above; for the third, auxiliary theory will be furnished by the new observations gained from the sphere of phenomena.

(Note that the definition of inductive extension comprehends such general epitheorems as, for instance, Gödel's incompleteness theorem.)

A distinction can be made between algorithmic and heuristic inductive statements.

Algorithmic extension. The possibility of being able to accomplish a precise extension without studying the phenomenal world is demonstrated by Myhill's theorem [4], according to which there exists an algorithm (constructive procedure) by which for every theory T that incomplete in the Gödel sense** there can be found an extended theory T' such that $T \subset T'$ and $E'=E$ and $T \neq T'$. Extensions of this sort (i.e. those which produce precise theorems) are called algorithmic extensions.

Heuristic extension. If the theory T is incomplete, then the truth or falsity of a statement, in a case where this cannot be decided, can be made more probable by converging to it with a series of heuristic attempts. The probability of the statement in question being true will become the new measure of truth [5]. In this way the theory will be extended, though at the same time the extended theory will necessarily be imprecise.

* Both partition and cover mean the breaking down of some set into subsets. The difference between them is that whereas a partition must resolve into disjunct classes, in the case of cover this stipulation is not made. More specifically, let the partition of the set A be $P=\{P_1, P_2, \ldots, P_n\}$ and the cover of A be $K=\{K_1, K_2, \ldots, K_n\}$. This means $A=P_1 \cup \ldots \cup P_n$ and $A=K_1 \cup \ldots \cup K_n$, but while $i \neq j \Rightarrow P_i \cap P_j = \emptyset$, this is not specified for K. On the other hand, all specifications serving to preserve the structure of the relations, operations, etc. defined on the set hold just as rigorously for K as for P. (See also later under homomorphic interpretation.)

** I.e. In the sense of Gödel's incompleteness theorem.

6.3. Deductive extension. The above process works in reverse in the following case. Consider theory T, the interpretations of which are the theories $T_1, \ldots, T_m$. As our definition of the resulting theory T we take $\hat{T}=T\cup T_1\cup\ldots\cup T_m\cup\{T\to T_1, , \ldots, T\to T_m$ are statements describing the interpretations$\}$. In theory $\hat{T}$ the objects of T are variables which display a range of interpretation $T_1\cup\ldots\cup T_m$.

If we are now presented with a new problem, we must first decide whether it is necessary or not to extend the scope of interpretation $T_1\cup\ldots\cup T_m$ in order to be able to solve the problem. If extension is unnecessary, then the theory T can be utilized unchanged to solve the problem. Where this is not possible, then we can make use of theory $\hat{T}$ if to obtain a solution such a theory T_{m+1} is needed which, although differing from all the above theories, can be produced as the interpretation of $\hat{T}$.

T_{m+1} is created by extending theory $\hat{T}$ in such a way that the extended theory will be $\hat{T}\cup T_{m+1}$ (including the statements describing the interpretational relations $T\to T_{m+1}$). For this we need a language $\hat{E}\cup E_{m+1}$. In the creation of this language the generative rules of the original inductive classes must be extended to the new objects and predicates. (For a more detailed treatment of the interrelations of the inductive classes, see the later section on homomorphic interpretation.)

To sum up: In inductive extension the theory T (more precisely the concepts of the theory; that is, the set of objects and the set of predicates) is broken down to equivalence classes, and for the identification of the equivalence classes new, more abstract concepts (objects and predicates) are introduced. Lastly, the rules of substitution are determined (the way in which an element of an equivalence class can be substituted for the abstract concept denoting that class).

By inductive extension we form a new theory T', the interpretation of which is the original theory T (both are of course parts of some theory $\hat{T}=T\cup T'\cup\ldots$) — though it is also possible only to extended the original theory T.* (The latter generally implies a simplification, since in most cases $|T|=\aleph_0$**; in other words, the size of T does not grow with extension, but the axioms and inferences are simplified by virtue of the more general relations.)

In deductive extension we proceed in the reverse manner: the cardinality (not the number) of the equivalence classes which correspond to some more abstract concepts is increased. (Here, too, it is possible to produce a new theory or extend the old one.)

7. The effect of extension on the measure of truth

As we have seen, the continuability of the measure of truth is a fundamental point. One way of providing for this continuability is to try to make the new interpretations homomorphic in the following sense:

Definition 10. An interpretation $i\colon T\to T'$ is homomorphic if $i\colon E\to E'$ providing its base is homomorphic. (It is especially important to ensure homomorphism when cover is used instead of partition in the abstraction.)

* In the development of a theory (or theories) the latter procedure corresponds to a continuous, the former to a jump, stage. These occur alternately, the path to the jump stage being prepared for by the continuous.

** $\aleph_0$ (aleph-zero) is the usual symbol for the cardinality of the set of natural numbers.

By the *precision* of a theory T we mean the relation of cardinality between its nucleus and periphery (imprecise part).

Let

$$M = \{t|\gamma(t) = 1, t \in T\}$$

$$P = T \setminus M$$

$$P = \frac{|M|}{|M|+|P|}$$

where "p" is the measure of the theory's precision. In accordance with the above, extensions can be ranged in three main classes.

Let the original theory be T and the extended theory T'. Then:

Definition 11. *a)* An extension is proportional if $p=p'$. (We then say that the theories T and T' resemble one other); *b)* An extension achieves an advance in precision if $p<p'$; *c)* An extension loses precision if $p>p'$.

If we want an extension to remain proportional and not to include steps achieving greater precision, then it must not contain steps resulting in greater imprecision either. It is not allowed, for instance, to use hypotheses during the extension. More exactly, theory T is incomplete, in Gödel's sense, if the language $E(E \supset T)$ includes statements such that, though they are not theorems of T, if T were to be (axiomatically) extended with them, a consistent theory would be generated.* If, during the extension, such a statement becomes a theorem of the new theory T', then we say that "hypotheses have been used" during the extension.

With regard to proportional extension it will be remembered that in defining interpretations no stipulation was made that the correspondance i: $T_0 \rightarrow T_1$ should be defined everywhere in T. It should therefore be possible to define i with respect to only the nucleus of T and thereby obtain, by deductive extension, a precise theory T_1 from the imprecise theory T. The same can be said of inductive extension, with the difference that here we can utilize the fact that the range of i may also be a subset of T_1, i.e. we do not have to make the condition $i(T)=T_1$.

8. The structure of the Systems Theory (ST)

8.1. The ST is a set of many theories between which connections and interactions of the sort described for an epitheory are possible.

The most abstract part of Systems Theory is General Systems Theory (GST), the interpretations of which are the SST-s (Specific Systems Theories). The SST-s are theories oriented to the individual types of systems (biological, technical, and so on). This two-level classification (GST and SST) is only a rough approximation of the real situation, however. The GST is steadily developing, newer and more abstract levels appear (in the case of inductive extension a whole new theory may be generated), new SST-s are thrown up as new interpretations of the GST; at the same time the articulation of the ST grows more refined, and new levels appear between GST and the SST-s. Nevertheless in the present study all the intermediate

* These are called statements insoluble in T.

levels are oncluded either in the GST, or in the SST category, and only these two are distinguished.

Since all the SST-s are homomorphic interpretations of the GST, it is sufficient to analyse in detail the structure of a single SST; the GST and all the other SST-s will be of similar construction.

Any SST can be partitioned into analytical aspects (AA) and — independently — into analytical levels (AL). These two kinds of partitioning are orthogonal to each other and their joint application is a basic step in using a SST.

An AL is grounded either on a theory or a subtheory. Such a base may be provided by an abstract theory, like information theory, or part of a theory, or even just a statement. An instance is the analysis of a computing centre in terms of some specific parameter (e.g. income or reliability). This theory or subtheory is called the base of the aspect (BA).

As already mentioned in the introduction, the analytical aspects differ from each other in the point of view from which they examine the given type of systems, and consequently in what kind of apparatus they utilize. For instance, there are information theoretical, automata theoretical, control theoretical, and energetical aspects. (A comparative analysis of some of these is provided by Kukhtenko [6].)

The AL-s differ from each other in the detail of the analysis of the given system type; that is, how small are the subsystems that are analysed functionally *only* (like a black box) and how large those that are analysed structurally as well.

Let as look more closely at the development of some interdisciplinary theory.

Consider a variety of theories $T_1, T_2, \ldots, T_n$ dealing with different system types (different phenomenal spheres). (These will correspond later to the SST-s.) If we want to base an interdisciplinary theory on these theories, we must form a theory $\hat{T}=T_1 \cup T_2 \cup \ldots \cup T_n$; then, by extending this theory in an inductive way, we can create a theory T' the interpretations of which are the theories $T_1, T_2, \ldots$*. First of all it is necessary to choose the *analytical aspects.* To do this we shall need to utilize some new subtheories (which are independent of $\hat{T}$), such as information theory, automata theory, etc. If the information theory, for instance, is represented by T_{inf}, the information theoretical aspect will have the form $\hat{T} \cap T_{\text{inf}}$. The analytical aspect obtained are then broken down to the level of the theories $T_1, T_2, \ldots$ and used to form the classification needed for the abstraction.

In contrast to the aspects, the *analytical levels* are obtained as the result of the reverse process. As was mentioned in connection with deductive interpretation, the solving of new problems often requires new interpretations. Such interpretations are the AL-s, the choice of which is determined by the depth of analysis necessary for the solution of the problem. Consequently, the AL-s arise during the search for adequate interpretations of the problems.

Let us investigate in detail the process of forming the analytical aspects (AA-s) and analytical levels (AL-s). This question, it will be noted, belongs more to the application of the ST than to its construction.

* In the case of Systems Theory, $T_1, T_2, \ldots$ correspond to $\text{SST}_1, \text{SST}_2, \ldots$ and T' corresponds to the GST.

8.2. Formation of the AA. This process will be analysed using the example of the biological ST. Let us take some SST, say Biological Systems Theory (further on BST), and let us search for its information theoretical aspect. (This implies, of course that the information theory (IT) is given too.) The process starts with the deductive extension of the information theory, and we must look for a homomorphic and valid interpretation i: IT→BST.* For this purpose a suitable extension of the BST is needed; this will, in fact, be made in the course of looking for the interpretation.

The interpretation eventually reached will permit us to make conclusions in the scope of the BST by means of information theory, though the manipulation is obviously likely to be extremely difficult. The difficulties can be surmounted by the creation of a congruence relation** $i^{-1} \circ i$ over the BST.

The congruence classes defined in this manner will form a theory, because the mapping i is homomorphic. The new theory is symbolized by $T' = \mathrm{BST}/(i^{-1} \circ i)$ to. Theory T' (the biological information theory) is isomorphic to IT and can be interpreted for the BST in a homomorphic way.

Generalizing to the formation of the aspects of an optional SST in accordance with some BA, the above process can be interpreted as a deductive extension of the memory BA SST in which the rules of deduction (drawing of a valid interpretation i) and the language (provision of a classification $\mathrm{SST}/(i^{-1} \circ i)$) have been extended.***

8.3. Formation of an AL. Let us investigate the application of a theory (T) to solving a problem (p), which will be a statement of some language. (We are not concerned at the moment with selecting an adequate theory T for the given problem; this will be discussed later.)

We introduce the following notation:

$$T \overset{\simeq}{\supset} p \overset{\mathrm{def}}{\Leftrightarrow} (\exists T' \subseteq T) T' \simeq T'' \,\&\, p \in T''.$$

* It is generally not possible to prescribe that an interpretation be adequate as well, but the adequacy will be prescribed for the minimum subtheory of the BST in which the problem can still be solved.

** By operation ∘ is meant a function composition, e.g. $(f \circ g)\ (x) = g(f(x))$. A congruence relation is an equivalence relation compatible with the operation. In other words, a relation is a congruence relation, if it can be formed as the component of a homomorphism and its inverse; that is, if there exists a homomorphism h to the relation r such that $r = h^{-1} \circ h$.

*** In other words, the factor theory $T' = \mathrm{SST}/(i^{-1} \circ i)$ is a partition on the SST. The set of objects (O') of theory T' is a partition of the objects of the SST. Therefore, to each object of T' can be added an arbitrary number (say n) of fixed variables $(x_1, \ldots, x_n)$ the range of which exactly covers the subclass of the objects of the SST pertaining to the given object. Applying the λ conversion of Church [7] to these variables, an unambiguous correspondance is gained between theory T' and the SST: $e = \lambda(x_1, \ldots, x_n) \circ e'$, where $e \in \mathrm{SST}$ and $e' \in T'$.

To examine this in more detail: Let F' stand for the set of predicates of T'. For every $o \in O'$ we form a word algebra $W_{F'}\ (o \cup \{x_1, \ldots, x_n\})$. Since every word algebra is free over the generator set, and the union of free algebras is free over the union of generator sets, the mapping i: $o \cup \{x_1, \ldots, x_n\} \to O$ unambiguously defines the homomorphism sought. Here O is the object set of SST.

The theory T' is applicable in the BST because the selected interpretation was homomorphic and thus a congruence partition compatible with the relations was obtained. Ensuring this homomorphism is a basic task in forming the AL's.

A well-know example of an application that has been unsuccessful is biological thermodynamics; a successful example would be biological information theory [8].

If $T \tilde{\supset} p$, then we say that problem p is *embeddable* in theory T. Let T be a subalgebra of the σ algebra of some universe J, and let $p \in J$. T'' is now a subalgebra of T. To solve p we must look for the minimal *subalgebra* (T'') of T in which p can still be solved ($p \in T''$); this T'' is called the *level* of the theory adequate to problem p. It is easy to see that this level will not be "homogeneous"; that is, the classes of the minimal partition of the selected subalgebra σ will not be of equal cardinality.* If the theory T'' which is adequate to the problem necessitates decisions which cannot be made within the theory T**, then the problem can be solved by the theory T only after this has been extended. In other words, T must be extended to such an algebra σ of the universe J as has a subalgebra adequate to p.

Say the problem is discovering the mechanism of metabolism in the human body. On this universe we can define certain subalgebras σ: e.g. anabolism, catabolism, chemical relations, etc. If the individual organs are described on the anabolic level, the union of the set of algebras we obtain will solve the problem, but it will not be minimal (e.g. we know that anabolic processes taking place in the liver differ from those in the muscles). On the other hand, if the solution of the problem is analysed at the level of chemical reactions, then the number of classes gained in the partition (i.e. the number of concepts) will be a minimum. This means that if such chemical laws are chosen as are common to the metabolism of the liver, muscles, etc., then the concepts that are obtained will be able to describe the whole metabolic process, e.g. the synthesis of starch and dextrose.

References

[1] CURRY, H. B., *Foundations of Mathematical Logic*, McGraw-Hill Book Company, Inc. 1963.
[2] LEE, E. T. and L. A. ZADEH, Note on Fuzzy Languages, *Information Sci.* v. 1, 1969, pp. 421—434.
[3] POST, E. L., Finite Combinatory Processes — Formulation 1. *J. Symbolic Logic*, v. 1, 1936, pp. 103—105.
[4] MYHILL, J., Notes for a series of lectures on recursive functions, In lecture notes for a summer conference on automata theory, University of Michigan, 1963.
[5] Бердзин, Я. Н., О вычислимости на вероятностных машинах, *Докл. АН СССР* v. 189, № 4, 1969, pp. 609—702.
[6] Кухтенко, А. И., Обзор основных направлений развития общей теории систем, Материалы координационного совещания секции технической кибернетики, Киев, 1969, pp. 3—62.
[7] COHN, P. M., *Universal Algebra*, New York and London, Harper Row. 1965.
[8] QUASTLER, H., *The Emergence of Biological Organizator*, New Haven and London, Yale University Press, 1964.

(Received October 17, 1973)

* The minimal partition of an algebra σ is taken to mean the infimum (or greatest lower bound) of the partitions generating the algebra.

** Let k be the most detailed partition formed by theory T on the universe (i.e. k is the greatest lower bound of the partitions generating T) and let k'' be one of the partitions generating T''. It follows that the decisions of T'' can be generated in T provided $k'' \geqq k$.

Релаксационные процедуры в задаче идентификации

В. И. Салыга, О. Г. Руденко

Выбор модели в задаче идентификации обусловливается не только априорными знаниями о структуре исследуемого объекта, режимах его функционирования, но и требуемой степенью точности, соответствием модели реальному объекту и сложностью ее реализации. Использование же в реальных системах гипотетических моделей лишает их гибкости и универсальности, так как сложные системы требуют быстродействующих вычислительных машин с большой емкостью блока памяти.

В данной работе рассматривается решение задачи идентификации статических и динамических объектов в режиме их нормального функционирования.

В общем виде многомерные объекты можно описать уравнениями:

а) статический объект

$$Y_n = \sum_{i=1}^{N} C_i X_{in} + \xi_n, \tag{1}$$

б) динамический объект

$$Y_n = \sum_{i=1}^{N} C_i A_i(X_{in}) + \xi_n, \tag{2}$$

где Y_n, X_{in} $(i=\overline{1, N})$ — наблюдаемые выходная и входные переменные, соответственно;

A_i $(i=\overline{1, N})$ — известные операторы преобразования входных переменных:

ξ_n — неконтролируемая помеха измерения выходной координаты;

C_i $(i=\overline{1, N})$ — компоненты неизвестного вектора параметров;

$n=0, 1, \ldots$ — дискретное время.

В случае описания исследуемых объектов уравнениями (1), (2) задача идентификации сводится к оцениванию неизвестного вектора параметров C.

Для стационарных объектов задача оценивания вектора неизвестных параметров по конечному числу наблюдений M $(M \geqq N)$ входных и выходных переменных эквивалентна решению системы линейных алгебраических уравнений

$$RC = F. \tag{3}$$

Большинство итерационных алгоритмов решения этой системы тесно связано с рекуррентной формой метода наименьших квадратов [1]. Для практической реализации этих алгоритмов требуется довольно большой объем вычислений, вызванный необходимостью учета всей предыдущей информации.

Существуют, однако, итерационные алгоритмы оценивания, которые позволяют учитывать не всю предыдущую информацию, а лишь часть ее или же только вновь поступающую. В этом случае задача оценивания неизвестного вектора параметров C сводится к задаче минимизации некоторого функционала $J(C)$.

Близость C к решению C^* в том или ином смысле характеризует некоторая вещественная функция $\varepsilon(C)$. В конкретных задачах $\varepsilon(C)$ можно выбирать различными способами, например, близость C к C^* может быть измерена ошибкой (абсолютной погрешностью) $\|C-C^*\|$, невязкой $\|J'(C)\|$ или J-невязкой

$$\varepsilon(C) = J(C) - J^*,$$

где J^*-минимальное значение функционала, соответствующее C^*.

Далее строится релаксационная последовательность $\{C_n\}$, обеспечивающая выполнение условия

$$J(C_0) \geqq J(C_1) \geqq J(C_2) \geqq \ldots \tag{4}$$

или

$$\varepsilon_0 \geqq \varepsilon_1 \geqq \varepsilon_2 \geqq \ldots,$$

где

$$\varepsilon_n = J(C_n) - J^*.$$

Построение такой последовательности можно осуществить по методике, предложенной в [2] и основанной на понятии расстояния между двумя последовательными оценками C_n и C_{n+1}. В нашем случае алгоритм оценки параметров будет иметь вид

$$C_{n+1} = C_n - \frac{J(C_n) - J^*}{\|J'(C_n)\|^2} J'(C_n), \tag{5}$$

где

$$J'(C_n) = \left(\frac{\partial J(C)}{\partial C_1}, \quad \frac{\partial J(C)}{\partial C_2}, \ldots, \quad \frac{\partial J(C)}{\partial C_N} \right), \quad \|J'(C_n)\| < \infty.$$

Алгоритм (5), напоминающий алгоритм Качмажа, достаточно быстро сходится при отсутствии помех.

Релаксационный эффект шага (C_n, C_{n+1}) зависит не только от направления, но и от относительной глубины смещения внутрь поверхности уровня, определяемой множителем релаксации p $(0<p<2)$. С учетом этого, алгоритм (5) несколько видоизменяется

$$C_{n+1} = C_n - p \frac{J(C_n) - J^*}{\|J'(C_n)\|^2} J'(C_n). \tag{6}$$

Сходимость этих алгоритмов следует из того, что величина разности между последующими оценками искомых коэффициентов и оптимальными значениями этих коэффициентов монотонно убывает.

Следует отметить, однако, что алгоритм (6), в отличие от остальных градиентных методов не является релаксационным, т. е. последовательность

$J(C_n)$ не обязательно монотонно убывает. Основным же преимуществом (6) является его чрезвычайная универсальность: он сходится практически для любых выпуклых функционалов, без всяких предположений о их гладкости [3].

Алгоритм (6) является частным случаем более общего s-шагового итерационного процесса, использующего для построения очередной итерации s-пре-

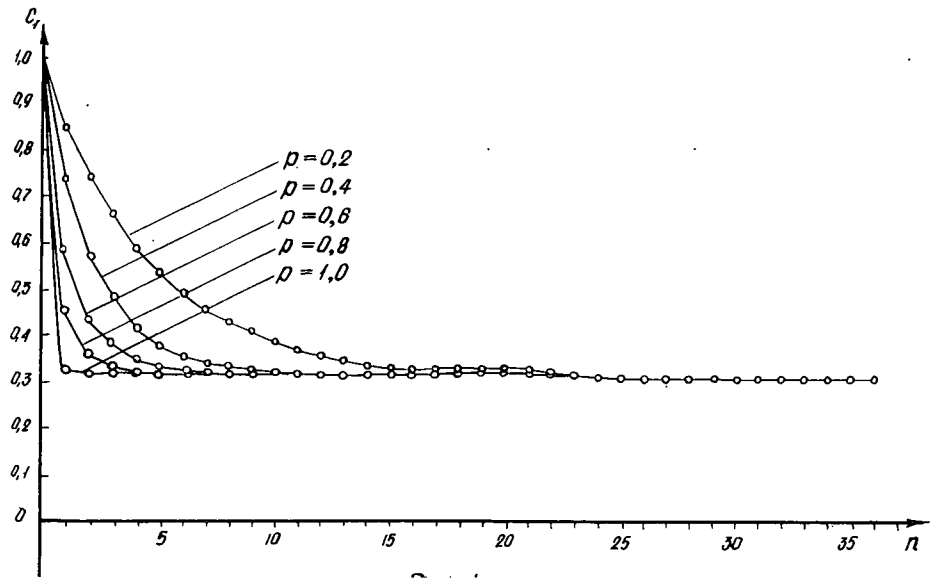


Рис. 1

дыдущих. Эти s-шаговые алгоритмы аналогичны алгоритмам групповой релаксации, применяемым в линейной алгебре [4]. В общем виде s-шаговый процесс можно записать

$$C_{n+1} = C_n - \left(J(C_n) - J^*\right) J_s'(C_n) \left(\|J_s'(C_n)\|^2\right)^{-1} e, \tag{7}$$

где

$$J_s'(C_n) = \left(J'(C_n), J'(C_{n-1}), \ldots, J'(C_{n-s})\right)$$

— матрица размерности $N \times s$, $e=(1, 0, \ldots, 0)$ — вектор $1 \times s$.

Для примера построим двухшаговый алгоритм идентификации, что соответствует $s=2$. Матрица $J_2'=\|J_2'(C_n)\|^2$ имеет вид

$$J_2' = \begin{vmatrix} \|J'(C_n)\|^2 & \left(J'(C_{n-1}), J'(C_n)\right) \\ \left(J'(C_n), J'(C_{n-1})\right) & \|J'(C_{n-1})\|^2 \end{vmatrix}.$$

Обратная ей вычисляется следующим образом [5]:

$$|J_2'|^{-1} = |i_{2km}^{(-1)}|_{k,m=1,2} = \frac{J_{2km}'}{\Delta},$$

где J_{2km}' — алгебраическое дополнение элемента i_{2km} в определителе Δ $(k, m=1, 2)$.

В нашем случае Δ является определителем Грама, т. е.

$$\Delta = \Gamma\left(J_2'(C_n), J_2'(C_n)\right) = \|J'(C_n)\|^2\|J'(C_{n-1})\|^2 - \left(J'(C_n), J'(C_{n-1})\right)^2,$$

а $|J_2'|^{-1}$ имеет вид

$$|J_2'|^{-1} = \Gamma^{-1}\left(J_2'(C_n), J_2'(C_n)\right)\begin{vmatrix} \|J'(C_{n-1})\|^2 & -\left(J'(C_n), J'(C_{n-1})\right) \\ -\left(J'(C_{n-1}), J'(C_n)\right) & \|J'(C_n)\|^2 \end{vmatrix}.$$

Подставив выражение для матрицы $|J_2'|^{-1}$ в (7) и произведя умножение, окончательно получим

$$C_{n+1} = C_n - \alpha_n J'(C_n) - \beta_n J'(C_{n-1}), \tag{8}$$

где

$$\alpha_n = \frac{\left(J(C_n) - J^*\right)\|J'(C_{n-1})\|^2}{\|J'(C_n)\|^2\|J'(C_{n-1})\|^2 - \left(J'(C_n), J'(C_{n-1})\right)^2},$$

$$\beta_n = -\frac{\left(J(C_n) - J^*\right)\left(J'(C_n), J'(C_{n-1})\right)}{\|J'(C_n)\|^2\|J'(C_{n-1})\|^2 - \left(J'(C_n), J'(C_{n-1})\right)^2}.$$

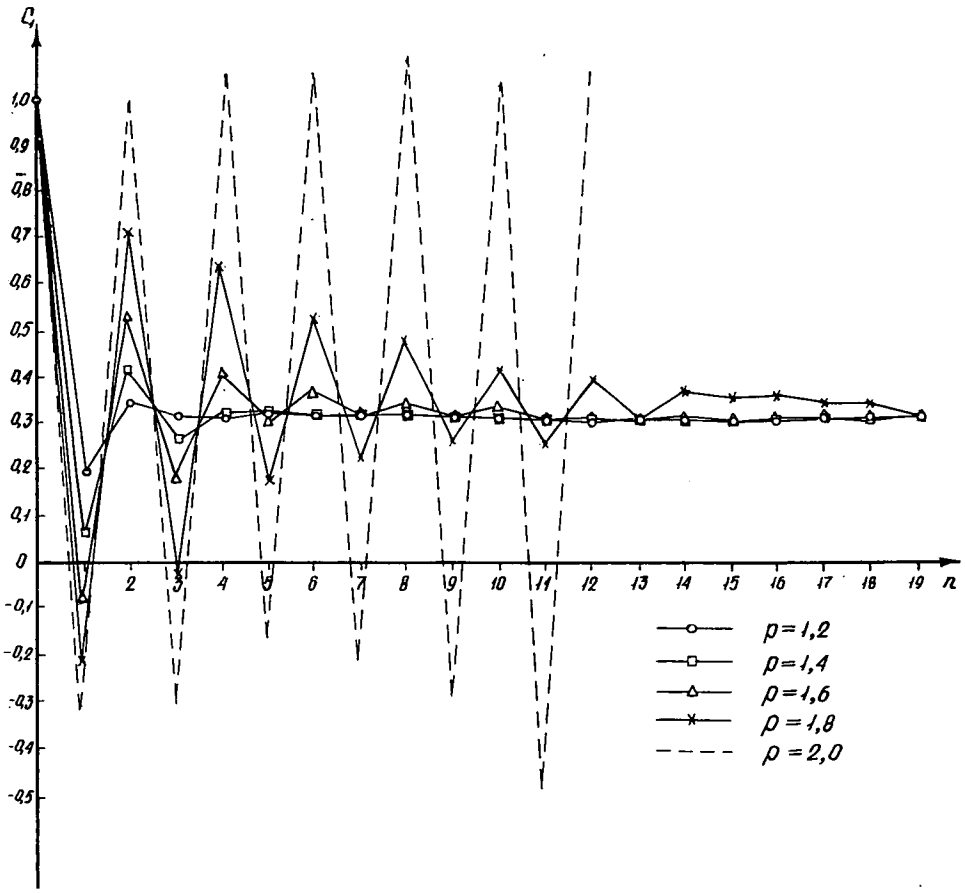


Рис. 2

Если же выполняется условие $\left(J'(C_n), J'(C_{n-1})\right) \geqq 0$, то следует взять

$$\alpha_n = \frac{J(C_n) - J^*}{\|J'(C_n)\|^2}, \quad \beta_n = 0.$$

При этом сходимость процесса оказывается более быстрой по сравнению со сходимостью соответствующего одношагового алгоритма (5), (6).

Для выявления характера функционирования и сравнительного анализа алгоритмов, а также для практической проверки полученных результатов и создания рабочих программ идентификации линейных объектов было проведено экспериментальное исследование полученных алгоритмов на ЦВМ. Исследовался процесс идентификации параметров многомерного линейного объекта обогатительной технологии [6].

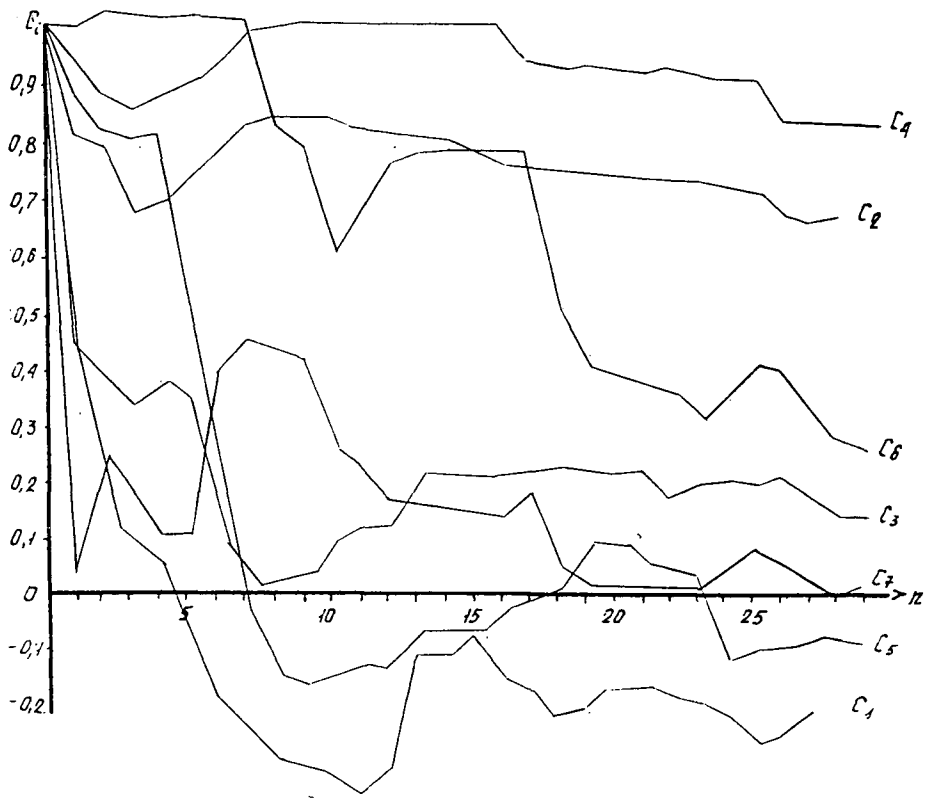


Рис. 3

Опыт исследования флотации углей показывает, что наиболее существенное влияние на эффективность процесса разделения исходного шлама на концентраты и отходы оказывают следующие переменные (входные переменные): расход пульпы Q_n, плотность пульпы δ, зольность исходного шлама A_u^c,

ситовый состав C, фракционный состав φ, количество реагента-вспенивателя Q_{pb} и реагента-собирателя Q_{pc}.

Выбор выходных координат модели обусловлен критерием эффективности процесса, который сформулирован на данном этапе как минимизация потерь угля в отходах при постоянном качестве концентрата. Таким образом, рассматривались две выходные переменные: зольность концентрата A_k^c и зольность хвостов A_u^c.

Рассмотренные входные и выходные переменные являются стационарными случайными процессами, и текущая информация может быть получена, в основном, путем пассивного эксперимента в режиме нормального функционирования объекта.

Исследования, проведенные в области флотопроцесса, позволяют сделать вывод, что операторы связи квазистационарны и в пределах рабочего диапазона изменения переменных близки к линейным.

На рис. 1—3 приведены результаты идентификации флотопроцесса по семи каналам управления. Графики (рис. 1, 2) показывают настройку коэффициента модели по координате A_k^c с использованием алгоритма (6) для разных значений величины p. Рис. 1 соответствует процессам нижней $0<p<1$ и полной релаксации $p=1$. На рис. 2 показана настройка коэффициентов модели при значениях $1<p<2$, что соответствует процессам верхней релаксации. Из графиков видно, что настройка коэффициентов при нижней релаксации протекает более плавно и монотонно, при верхней — настройка носит колебательный характер, и чем большее значение принимает в интервале (1, 2), тем резче выражен колебательный характер настройки. Выбор $p=2$ приводит к тому, что процесс не сходится. На рис. 3 представлены кривые настройки остальных параметров при $p=1$.

Проведенные исследования были использованы при синтезе модели промышленного флотационного процесса; алгоритмы идентификации входят в состав внешнего математического обеспечения ЭВМ в создаваемой автоматизированной системе управления обогатительными фабриками.

Relaxation methods in the identification problem

By V. I. Salyga, O. G. Rudenko

Summary

The identification of static and dynamic systems in normal functioning is considered. Here identification means the estimation of an unknown vector C of parameters. To simplify the computing process the use of relaxation algorithms of identification is proposed. This allows us to consider merely a part of the preceeding information or the last one, not all of it. In this case estimating the vector C is equivalent to the minimization of a certain functional $J(C)$.

In the paper identification algorithms of S steps are discussed and results of their experimental examination by computer are presented.

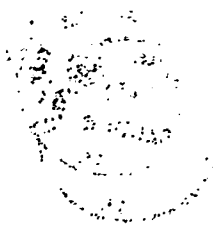
INSTITUTE OF RADIOELECTRONICS
USSR 310059, KHARKOV,
LENIN AVENUE 14.

Литература

[1] Ли, Р., Оптимальные оценки, определение характеристик и управление, *Наука*, 1966.
[2] Аведьян, Э. Д., Релаксационные алгоритмы идентификации линейных объектов, *Автореф. канд. дисс.*, 1972.
[3] Поляк, Б. Т., Минимизация негладких функционалов, *Журнал вычислительной математики и мат. физики*, 9, № 3, 1969.
[4] Фаддеев, Д. К.&В. Н. Фаддеева, Вычислительные методы линейной алгебры, ГИФМЛ, 1963.
[5] Гантмахер, Ф. Р., Теория матриц, *Наука*, 1967.
[6] Салыга, В. И., Автоматическое управление и построение моделей технологических процессов на обогатительных фабриках, 1972.

(Поступило 18-ого 1973 г.)


010 012008 290010 01



INDEX — TARTALOM

Felelős szerkesztő és kiadó: Kalmár László
A kézirat a nyomdába érkezett: 1975. április hó
Megjelenés: 1975. december hó
Példányszám: 1000. Terjedelem: 7,68 (A/5) ív
Készült monó szedéssel, íves magasnyomással,
az Msz 5601-90 és az Msz 5602-55 szabvány szerint
75-2200 — Szegedi Nyomda