Tomus 2. Fasciculus 2.

ACTA CYBERNETICA

FORUM CENTRALE PUBLICATIONUM CYBERNETICARUM HUNGARICUM

Szeged, 1973

Curat: Universitas Szegediensis de Attila József nominata

2. kötet 2. füzet

ACTA CYBERNETICA

A HAZAI KIBERNETIKAI KUTATÁSOK KÖZPONTI PUBLIKÁCIÓS FÓRUMA

Szeged, 1973. november
A szegedi József Attila Tudományegyetem gondozásában

Mathematische Fassung der sogenannten „Entscheidungs-Tabellen"

VON R. PÉTER

§ 1

In der Programmierung ist die Verfertigung von Skizzen, „flow-diagram" genannt, seit langem üblich. Allgemein werden diese ohne exakte Definition gebraucht. KALUŽNIN hat für diesen Begriff mit der Benennung „Graphschema" eine exakte Definition angegeben.[1]

Hier gebe ich ein Beispiel für ein flow-diagram zur Berechnung bei gegebenem k der k-ten binären Ziffer s_k zweier binär gegebenen Zahlen:

$$\ldots a_2 a_1 a_0 + \ldots b_2 b_1 b_0$$

wo links vom letzten Ziffer 1 beliebig viele Ziffern 0 stehen können. Man hat vor Augen zu halten, daß für alle n die Summenziffer s_n nicht nur von a_n und b_n, sondern auch vom Rest r abhängt, der von der rechtseitig bereits durchführten Addition der Ziffern übriggeblieben ist.

Es handelt sich um einen endlichen, zusammenhängenden, gerichteten Graphen, aus dessen Knotenpunkten (die ich kurz „Punkte" nennen werde) höchstens zwei Kanten hinauslaufen. Es gibt ein ausgezeichneter Punkt I („Input"), wohin keine Kante hinein-, und ein ausgezeichneter Punkt O („Output"), woraus keine Kante hinausläuft. Punkte mit einer einzigen hinauslaufenden Kante und auch O werden mathematische Punkte genannt, und Punkte mit zwei hinauslaufende Kanten heißen logische Punkte. Die aus den letzteren hinauslaufenden zwei Kanten werden durch J („Ja") bzw. N („Nein") bezeichnet.

Den Punkten des Graphen werden Funktionen zugeordnet. Vorläufig werde ich aber, wie in der Praxis üblich, ohne exakte Definition den logischen Punkten Fragen, den mathematischen Punkten Anweisungen zuordnen, und daher das Schema nur „Vorgraphschema" nennen. Auf die exakte Definition komme ich im § 10 zurück.

Je einem mathematischen Punkt wird in unserem Beispiel eine Anweisung der Form

$$c \Rightarrow v$$

zugeordnet, wodurch verlangt wird, daß einer Variable v der Wert c gegeben werden

[1] Siehe z.B. R. PÉTER: *Graphschemata und rekursive Funktionen.* Dialectica **12** (1958) S. 373—393 und R. PÉTER: *Über die Partiell-Rekursivität der durch Graphschemata definierten zahlentheoretischen Funktionen.* Annales Univ. Sci. Budapestiensis **2** (1959) S. 41—48.

soll (ohne Hinsicht darauf, daß v vielleicht früher schon einen anderen Wert erhalten hat); und je einem logischen Punkt eine Frage der Form

$$c=b?$$

Unser Vorgraphschema ergibt sich folgenderweise (da man mit der Addition von a_n und b_n für $n=0$ beginnt, wobei noch kein Rest vorhanden ist):

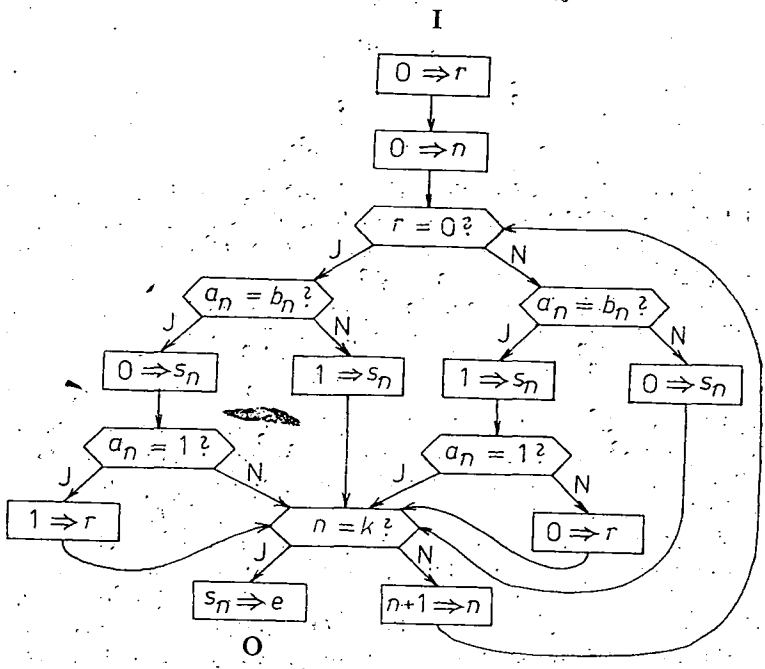

Abb. 1

Diese Einrichtung funktioniert wie folgt: Von I beginnend hat man an Kanten in ihrer Richtung entlanggehend die begegneten Anweisungen durchzuführen; wobei man aus einem logischen Punkt auf der durch J oder N bezeichneten Kante weiterzugehen hat, je nachdem die Antwort auf die in diesem Punkt gestellten Frage „Ja" oder „Nein" ist. Gelangt man zu O, daraus führt der Weg nicht weiter; man kann nachprüfen, daß der hier erhaltene Wert der „Ergebnisvariablen" e die gesuchte k-te Ziffer der betrachteten Summe ist.

§ 2

Bereits das zur vorherigen einfachen Aufgabe gehörige Vorgraphschema ist ziemlich verwickelt; besonders dort ist die Sachlage nicht leicht zu überblicken, wo mehrere logische Punkte aufeinander folgen. Seit einer Zeit begann die Tendenz,

derartige Graphenteile durch besser überblickbare sogenannte „Entscheidungstabellen" zu ersetzen[2] (die ich kurz „Tabellen"-nennen werde).

Eine Entscheidungs-Tabelle wird immer auf vier Quadranten geteilt:

I		II	
III		IV	

In I kommen verschiedene Fragen, in III verschiedene Anweisungen. Der übrigbleibende — aus II und IV bestehende — Teil der Tabelle wird auf eine Anzahl von Spalten geteilt. Der „obere" d. h. zu II gehörige Teil je einer Spalte enthält eine Variation von J, N und „leer" Zeichen (ein Leerzeichen kann ein Strich oder nichts sein); der „untere" d. h. zu IV gehörige Teil je einer Spalte enthält eine Variation von X und „leer" Zeichen. Zur Bedeutung der Tabelle betrachten wir ein Beispiel: Ist I und III, ferner eine der Spalten wie folgt ausgefüllt:

F_1		J	
F_2			
F_3		N	
A_1			
A_2		X	
A_3		X	
A_4		X	

das bedeutet, daß falls die Antwort auf die Frage F_1 „Ja" und auf die Frage F_3 „Nein" ist, so hat man — unabhängig davon, wie die Antwort auf F_2 ausfällt — den Anweisungen A_2, A_3 und A_4 zu genügen.

Die oberen Teile zweier Spalten dürfen nicht übereinstimmen; denn stimmten dabei auch ihre unteren Teile überein, so wäre überflüssig zweimal dasselbe zu verlangen; und stimmten ihre unteren Teile nicht überein, so könnte damit ein Widerspruch verlangt werden.

Betrachten wir einen solchen Teil des im § 1 angegebenen Graphen, wo mehrere logische Punkte aufeinander folgen:

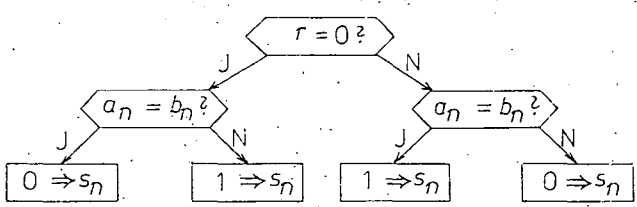

Abb. 2

Daraus würde man so eine dasselbe leistende Tabelle herstellen, daß man vom Anfangspunkt ausgehend alle mögliche gerichtete Kantenzüge (diese werde ich

[2] Siehe R. THURNER: *Entscheidungs-Tabellen* (Düsseldorf, 1972), mit der darin angegebenen Literatur.

kurz „Linien" nennen) begeht; die unterwegs gefundenen Fragen — jede nur einmal — in I, die Anweisungen — auch jede nur einmal — in III einführt, und für jede Linie je eine Spalte derart ausfüllt, daß in die Zeile jeder Frage „leer", J oder N kommt, je nachdem auf dieser Linie die betreffende Frage nicht aufgetreten ist, oder in einem solchen Punkt aufgetreten ist, aus welchem die Linie auf der J-Kante bzw. auf der N-Kante weitergeführt hat; endlich in die Zeile jeder Anweisung X oder nichts kommt, je nachdem ein Punkt mit dieser Anweisung zur Linie gehört oder nicht.

Immer die möglichst linkseitigen Linien wählend ergibt sich so die folgende Tabelle:

$r = 0?$	J	J	N	N
$a_n = b_n?$	J	N	J	N
$0 \Rightarrow s_n$	X			X
$1 \Rightarrow s_n$		X	X	

Wollte man aber aus dieser Tabelle den betrachteten Teilgraphen rekonstruieren, das würde aufs erste nicht eindeutig ausfallen. Aus den beiden ersten Spalten könnte man noch eindeutig den Teil

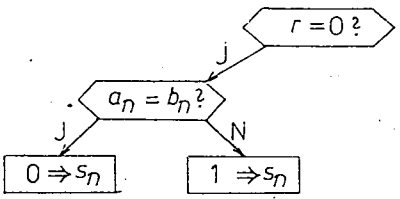

Abb. 3

zurückerhalten. Auch das, daß die zur dritten Spalte gehörige Linie mit der aus dem Anfangspunkt auslaufenden N-Kante beginnt. Diese Kante könnte aber auch in den bereits gezeichneten Punkt mit der Frage „$a_n = b_n$?" einlaufen, und weiter der daraus hinauslaufenden J-Kante entlang, was mit der Endanweisung der betrachteten dritten Spalte einen Widerspruch geben würde:

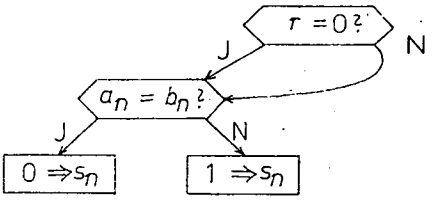

Abb. 4

Deswegen ist es ratsam, die Forderung, daß die in I eingetragenen Fragen und die in III eingetragenen Anweisungen verschieden sein sollen, fallen zu lassen, und die Zeilen der Tabelle nicht als zu verschiedenen Fragen bzw. Anweisungen, sondern zu verschiedenen Punkten gehörig zu betrachten (wobei dieselbe Frage

oder Anweisung auch in verschiedenen Punkten auftreten kann). So ist die zum betrachteten Teilgraphen gehörige Tabelle:

T_1^*:

$P_{1,1}$	$r = 0?$	J	J	N	N
$P_{1,2}$	$a_n = b_n?$	J	N		
$P_{1,3}$	$a_n = b_n?$			J	N
$P'_{1,1}$	$0 \Rightarrow s_n$	X			
$P'_{1,2}$	$1 \Rightarrow s_n$		X		
$P'_{1,3}$	$1 \Rightarrow s_n$			X	
$P'_{1,4}$	$0 \Rightarrow s_n$				X

wobei $P_{1,i}$ bzw. $P'_{1,i}$ den bei der Herstellung der Tabelle T_1^* verwendeten i-ten logischen bzw. mathematischen Punkt bezeichnet.

Daraus läßt sich der Teilgraph eindeutig rekonstruieren.

§ 3

Betrachtet man wieder den ganzen Graphen des § 1, so sieht man, daß die Fortsetzungen des im § 2 behandelten Teilgraphen wieder zu logischen Punkten führen. Mit einem von diesen (z. B. mit dem linkseitigen) beginnend betrachten wir wieder den aus jenen Linien bestehenden Teilgraphen, die sich bis zum ersten mathematischen Punkt oder — wenn dies der Fall wäre — bis zu einem bereits erreichten Punkt erstrecken. (Würden auf den mathematischen Endpunkt einer der Linien weitere mathematische Punkte folgen, so hätte man die Linie bis zum ersten neuen logischen Punkt bzw. bis zur Rückkehr zu einem früher erreichten Punkt zu verlängern.) Der genannte Teilgraph ist:

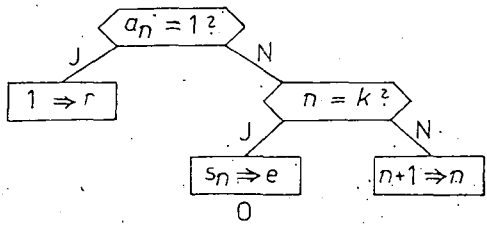

Abb. 5

und die dazu gehörige Tabelle ist:

T_2^*:

$P_{2,1}$	$a_n = 1?$	J	N	N
$P_{2,2}$	$n = k?$		J	N
$P'_{2,1}$	$1 \Rightarrow r$	X		
$P'_{2,2}$	$s_n \Rightarrow e$		X	
$P'_{2,3}$	$n+1 \Rightarrow n$			X

wobei alle Punkte von den Punkten von T_1^* verschieden sind.

Nun gibt es im Graphen des § 1 bereits nur ein einziger bisher nicht verwendeter logische Punkt. Damit beginnend soll auf ähnliche Art wie bisher ein Teilgraph abgesondert werden:

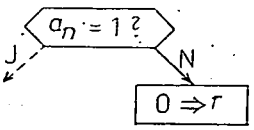

Abb. 6

Die mit Strichellinie gezeichnete J-Kante führt in den logischen Punkt $P_{2,2}$ des vorangehenden Teilgraphen T_2^*. Die Fortsetzung ist der mit diesem Punkt beginnende Teil dieses Teilgraphen. Diesem entspricht die folgende „Untertabelle" von T_2^*:

$T_{2,2}^*$:

$P_{2,2}$	$n = k$?	J	N
$P'_{2,2}$	$s_n \Rightarrow e$	X	
$P'_{2,3}$	$n+1 \Rightarrow n$		X

So ist es zweckmäßig zu den Anweisungen — als „Ausgang" aus der Tabelle — auch solche wie „go to $T_{i,j}$" („gehe zu $T_{i,j}$") hinzuzunehmen, welche die Durchführung der mit dem Punkt $P_{i,j}$ beginennden Untertabelle der Tabelle T_i verlangt.

Damit gestaltet sich die zum letzten Teilgraphen gehörige Tabelle wie folgt:

T_3^*:

$P_{3,1}$	$a_n = 1$?	J	N
$P'_{3,1}$	$0 \Rightarrow r$		X
	go to $T_{2,2}^*$	X	

wobei $P_{3,1}$ und $P'_{3,1}$ sowohl von den Punkten von T_1^* als auch von den Punkten von T_2^* verschieden sind.

Auch sonst lohnt es sich einen „Ausgangsteil" zu den Tabellen hinzuzufügen (der nicht zum „unteren Teil" der Tabelle gehört), mit Hinweise darauf, wohin die letzte Kante, die zu je einer Spalte gehört, führen muß.

T_3^* gestaltet sich dadurch — mit der ergänzten Form $T_{2,2}$ statt $T_{2,2}^*$ — als

T_3:

$P_{3,1}$	$a_n = 1$?	J	N
$P'_{3,1}$	$0 \Rightarrow r$		X
	go to $T_{2,2}$	X	X

und man hat auch T_1^* und T_2^* ähnlich zu ergänzen.

Ferner kommen noch die beiden mathematischen Punkte, die an der Linie vom Input bis zum ersten logischen Punkt zu finden sind, in keiner der bisherigen Tabellen vor. Sei für diese die folgende, eine einzige Spalte enthaltende Tabelle T_0 mit leerem oberen Teil angegeben:

T_0:

$P'_{0,1}$	$0 \Rightarrow r$	X
$P'_{0,2}$	$0 \Rightarrow n$	X
	go to T_1	X

(wobei ein T_i immer mit der Untertabelle $T_{i,1}$ von T_i identisch ist).

§ 4

So gehört zum Graphen des § 1 das folgende Tabellensystem:

T_0:

$P'_{0,1}$	$0 \Rightarrow r$	X
$P'_{0,2}$	$0 \Rightarrow n$	X
	go to T_1	X

T_1:

$P_{1,1}$	$r = 0?$	J	J	N	N
$P_{1,2}$	$a_n = b_n?$	J	N		
$P_{1,3}$	$a_n = b_n?$			J	N
$P'_{1,1}$	$0 \Rightarrow s_n$	X			
$P'_{1,2}$	$1 \Rightarrow s_n$		X		
$P'_{1,3}$	$1 \Rightarrow s_n$			X	
$P'_{1,4}$	$0 \Rightarrow s_n$				X
	go to T_2	X			
	go to $T_{2,2}$		X		X
	go to T_3			X	

T_2:

$P_{2,1}$	$a_n = 1?$	J	N	N
$P_{2,2}$	$n = k?$		J	N
$P'_{2,1}$	$1 \Rightarrow r$	X		
$P'_{2,2}$	$s_n \Rightarrow e$		X	
$P'_{2,3}$	$n+1 \Rightarrow n$			X
	go to $T_{2,2}$	X		
	stop		X	
	go to T_1			X

Dabei ist

$T_{2,2}$:

$P_{2,2}$	$n = k?$	J	N
$P'_{2,1}$	$s_n \Rightarrow e$	X	
$P'_{2,2}$	$n+1 \Rightarrow n$		X
	stop	X	
	go to T_1		X

T_3:

$P_{3,1}$	$a_n = 1?$	J	N
$P'_{3,1}$	$0 \Rightarrow r$		X
	go to $T_{2,2}$	X	X

Man findet, daß das Berechnungsverfahren dadurch etwas übersichtlicher geschildert wird als durch den Graphen des § 1.

Ferner wird auch als nützlich geschetzt, daß die einzelnen Tabellen von verschiedenen Mitarbeitern bearbeitet (einige etwa auch erweitert oder sonstwie abgeändert) werden können, ohne ihre Zusammenhänge zu stören. (Das „go to T_i" bzw. „go to $T_{i,j}$" bedeutet dann tatsächlich ein Hingehen — zum Arbeitstisch jenes Mitarbeiters, der die TabelleT_i, bzw. die Untertabelle $T_{i,j}$ bearbeitet. Gewisse Kanten des flow-diagrams werden durch solche Spaziergänge vertreten.)

§ 5

Es hätte vorkommen können, daß an einer der Linien, die zur Bildung der Spalten einer Tabelle dienten, zwei verschiedenen Punkten dieselbe Frage F zugeordnet wurde, so daß in der entsprechenden Spalte dieselbe Frage zweimal beantwortet wurde: entweder überflüssig, oder einander widersprechend. Doch der zugrundegenommene Graph kann immer durch einen zum selben Ergebnis führenden anderen Graph vertreten werden, worin keine solche Situation vorkommt (die also in dieser Hinsicht „normiert" ist).

Sei nämlich P_1 der erste und P_2 der zweite unter den aufeinander folgenden logischen Punkten der Linie L, denen dieselbe Frage F zugeordnet wurde. Die Antwort auf F ist in beiden Punkten dasselbe, also gehören zur Linie L eine aus P_1 und eine aus P_2 hinauslaufende Kante mit derselben Bezeichnung. Führt die letztere zu einem Punkt Q, und führte aus einem Punkt P der Linie L eine Kante zu P_2, so kann die Kante PP_2 gestrichen, und dafür eine Kante PQ aufgenommen werden. Führte im Graphen zu P_2 keine andere Kante als die gestrichene, so hat man auch die Kante P_2Q zu streichen, und jeden mit danach nur durch aus P_2 hinauslaufende Linien erreichbaren Punkt (P_2 inbegriffen).

Noch eine Bemerkung: wäre eine Linie zum mathematischen Punkt $P'_{0,2}$ (wohin die vom Input hinauslaufende Kante führt) zurückgelangt, so hätte man danach den Teil

$P'_{0,2}$	$0 \Rightarrow n$	X
	go to T_1	X

von T_0 auszuführen. Dieser soll die zu $P'_{0,2}$ gehörige Untertabelle $T'_{0,2}$ von T_0 genannt werden. So können im Ausgang einer Tabelle auch Anweisungen „go to $T'_{i,j}$" vorkommen.

§ 6

Allgemein erhält man aus einem (nach § 5 normierten) Vorgraphschema folgender Weise ein zum selben Ergebnis führendes Tabellensystem:

Ist das Input ein mathematischer Punkt $P'_{0,1}$, so gehen wir der davon auslaufenden Linie entlang bis wir entweder einen logischen Punkt $P_{1,1}$ finden, oder ohne einen logischen Punkt zu finden, zum Output gelangen; und man hat dieser Linie entsprechend die nur mathematische Punkte

$$P'_{0,1}, P'_{0,2}, \ldots, P'_{0,I}$$

und eine einzige Spalte enthaltende Tabelle T_0 (mit leerem oberen Teil) so zu bilden, wie die zum Beispiel des § 2 gehörige spezielle Tabelle T_0 in § 4 gebildet wurde; doch mit der Ausgangsanweisung „stop" statt „go to T_1", falls $P_{0,1}$ das Output ist.

Ist das Input ein logischer Punkt, so soll dieser mit $P_{1,1}$ bezeichnet (und keine Tabelle T_0 gebildet) werden.

Mit $P_{1,1}$ beginnt jedenfalls die Bildung einer Tabelle T_1 wie folgt (wobei der exakte Sinn der „möglichst linkseitigen Wahl" angegeben wird):

Zur Bildung der ersten Spalte wird eine von $P_{1,1}$ ausgehende Linie L_1 folgender Beschaffenheit verwendet: L_1 durchläuft solange wie möglich immer neue J-Kanten, dann immer neue unbezeichnete (d.h. aus mathematischen Punkten auslaufende) Kanten (dieser spätere Teil kann auch fehlen), bis eine Kante (1) zum Output, oder (2) zu einem bereits (bei der Bildung von T_0 oder vom bisherigen Teil von L_1) verwendeten Punkt $P_{i,j}$ oder $P'_{i,j}$ ($i \leqq 1$) zurück, oder aber (3) zu einem noch nicht verwendeten (kurz: „neuen") logischen Punkt führt. Dieser Linie entsprechend werden (wie bei der Bildung der speziellen Tabelle T_1 des § 4) die erste Spalte und die dazu gehörigen Teile der Quadranten I und III von T_1 gebildet; im Fall (1) mit dem Spaltenausgang „stop", im Fall (2) mit dem Ausgang „go to $T_{i,j}$" bzw. „go to $T'_{i,j}$", im Fall (3) mit dem Ausgang „go to T_2".

Wurden bereits Linien $L_1, L_2, \ldots, L_n$ und die entsprechenden ersten n Spalten mit den zu ihnen gehörigen Teilen der Quadranten I und III von T_1 gebildet, und enthält die n-te Spalte auch J-Zeichen, dann hat L_{n+1} den Anfangsteil von L_n bis zur letzten darin auftretenden J-Kante zu folgen, statt dieser aber die aus demselben logischen Punkt auslaufende N-Kante, nachher solange wie möglich lauter neue J-Kanten, endlich, wenn möglich, und solange möglich, lauter neue unbezeichnete Kanten zu durchlaufen; bis eine Kante (1) zum Output, oder (2) zu einem bereits (bei der Bildung von T_0, von den ersten n zu T_1 gehörigen Spalten, oder vom bisherigen Teil von L_{n+1}) verwendeten Punkt $P_{i,j}$ oder $P'_{i,j}$ ($i \leqq 1$) zurück, oder aber (3) aus einem mathematischen Punkt zu einem noch nicht verwendeten logischen Punkt führt. Dieser Linie entsprechend werden die $(n+1)$-te Spalte und die dazu gehörigen Teile der Quadranten I und III von T_1 ausgefüllt; im Fall (1) mit dem Spaltenausgang „stop"; im Fall (2) mit dem Ausgang „go to $T_{i,j}$" bzw. „go to $T'_{i,j}$"; im Fall (3) mit einem Ausgang „go to T_k", wobei $k=2$ ist, wenn Fall (3) bei der Bildung der ersten n Spalten von T_1 nicht vorgekommen ist; sonst ist k entweder gleich einem jener i, für welche unter den ersten n Spaltenausgängen von T_1 „go to T_i" vorkommt, oder um 1 größer als das größte dieser Zahlen i.

Enthält die n-te Spalte von T_1 schon keine J-Zeichen, so sind keine weitere Spalten zu bilden; die Tabelle T_1 ist fertig.

Ist Q ein Punkt des betrachteten Graphen, der weder in T_0 noch in T_1 aufgetreten ist, so muß Q auf einer aus $P_{1,1}$ ausgehenden Linie L liegen, doch weder auf dem ersten, lauter logische Punkte enthaltenden, noch auf dem darauf folgenden, lauter mathematische Punkte enthaltenden Teil von L; so muß auf dem weiteren Teil von L ein neuer logischer Punkt auftreten, und daher als Ausgang jener Spalte von T_1, die einem Anfangsteil von L entspricht, ein „go to T_i" mit $i \geqq 2$; doch so muß unter den Spaltenausgängen von T_1 auch „go to T_2" vorkommen.

Tritt also „go to T_2" nicht unter den Spaltenausgängen von T_1 auf, so kommen schon alle Punkte des betrachteten Graphen in T_0 oder in T_1 vor; also besteht das entsprechende Tabellensystem allein aus T_0 und T_1 (im Fall eines logischen Inputs allein aus T_1).

Sonst hat man in T_1 die erste Spalte mit dem Ausgang „go to T_2" zu betrachten. Der „neue" logische Punkt, zu dem die letzte Kante der zu dieser Spalte gehörigen Linie führt, sei durch $P_{2,1}$ bezeichnet. Damit beginnend ist T_2 ähnlich zu bilden, wie T_1 mit $P_{1,1}$ beginnend gebildet wurde.

Nehmen wir an, daß schon $T_1, T_2, \ldots, T_n$ ähnlich gebildet wurden, und Q ein Punkt des betrachteten Graphen ist, der in keinem von diesen auftritt. Sei $1 \leqq i \leqq n$ die größte Zahl, womit eine von $P_{1,1}$ ausgehende und zu Q führende Linie L den Anfangspunkt $P_{i,1}$ von T_i enthält. Den mit $P_{i,1}$ beginnenden Teil von L betrachtend schließt man ähnlich wie im Spezialfall $n=i=1$ darauf, daß ein Spaltenausgang von T_i eine Anweisung „go to T_j" mit $j>i$ sein muß; nach der Wahl von i kann aber j keines der Indizes

$$i+1, i+2, \ldots, n$$

sein. Dann mußte aber auch „go to T_{n+1}" als Spaltenausgang in einem der Tabellen $T_1, \ldots, T_n$ auftreten. Sei T_k die erste in der Folge dieser Tabellen, und darin die l-te Spalte die erste, worin dies der Fall war. Dann soll der logische Punkt, zu dem die letzte Kante der zur l-ten Spalte von T_k gehörigen Linie führt, durch $P_{n+1,1}$ bezeichnet, und damit beginnend die Tabelle T_{n+1} ähnlich wie die vorher entstandenen Tabellen gebildet werden.

Kam unter den Spaltenausgängen von $T_1, \ldots, T_n$ kein „go to T_{n+1}" vor, so wurden zur Bildung dieser Tabellen bereits alle Punkte des betrachteten Graphen verwendet, und so besteht das gesuchte Tabellensystem aus $T_1, \ldots, T_n$ und eventuell noch T_0.

Es muß noch der allgemeine Begriff der Untertabellen exakt angegeben werden.

Eine Untertabelle $T_{i,j}$ einer Tabelle T_i wird so erhalten, daß daraus erst die ersten $j-1$ Zeilen, dann jene Spalten, die in der j-ten Zeile leer sind, gestrichen werden; dann auch alle Zeilen, in welchen nachher keines der Zeichen J, N, X übriggeblieben ist.

Bei der Bildung einer Untertabelle $T'_{i,j}$ hat man nach Streichung des oberen Teils von T_i genau so zu verfahren (wobei also „j-te Zeile" die j-te Zeile des übriggebliebenen Teils von T_i bedeutet). Zu $T'_{i,j}$ gehört so immer eine einzige Spalte.

Z. B. ist für die spezielle Tabelle T_1 des § 4:

$T_{1,3}$:

$P_{1,3}$	$a_n = b_n$?	J	N
$P'_{1,3}$	$1 \Rightarrow s_n$	X	
$P'_{1,4}$	$0 \Rightarrow s_n$		X
	go to $T_{2,2}$		X
	go to T_3	X	

und

$T'_{1,3}$:

$P'_{1,3}$	$1 \Rightarrow s_n$	X
	go to T_3	X

Natürlich ist $T'_{0,1}$ die Tabelle T_0 selbst, und $T_{i,1}$ für jedes $i \neq 0$ die Tabelle T_i selbst.

Die Reihenfolge (und auch der dadurch beeinflusste Inhalt) der Tabellen hätte auch anders gewählt werden können.

§ 7

Ist umgekehrt ein Tabellensystem gegeben, so ist es wichtig, dies in ein zum selben Ergebnis führendes Graphschema (vorläufig nur „Vorgraphschema") zu umwandeln; denn das kann unmittelbar auf Programmierungsprachen übersetzt werden[3]).

Aus dem in § 6 geschilderten Verfahren ergeben sich nicht beliebige Tabellensysteme, sondern nur gewisse „regelmäßige", mit folgenden Eigenschaften:

(a) Das Tabellensystem besteht aus endlich vielen Tabellen ohne gemeinsame Punkte

$$T_1, T_2, \ldots, T_n \text{ und eventuell } T_0.$$

Kommt T_0 vor, so enthält T_0 — aber nur T_0 — keinen oberen Teil, und in keiner der Tabellen tritt ein Ausgang „go to T_0" auf. Kommt T_0 nicht vor, so tritt in keiner der Tabellen ein Ausgang „go to T_1" auf. Für jede andere Tabelle T_i gibt es mindestens ein Spaltenausgang „go to T_i" (eventuell in der Form „go to $T_{i,1}$").

(b) Als Spaltenausgänge jeder der Tabellen T_m ($m \leqq n$) können Anweisungen folgender Form vorkommen:

$$\text{stop}, \quad \text{go to } T_k, \quad \text{go to } T_{i,j}, \quad \text{go to } T'_{i,j}$$

(wobei $i \leqq m$ ist, das ist aber nach der Bemerkung zum Schluß des § 6 ohne Belang).

(c) Zu einer einzigen Tabellenspalte gehört der Ausgang „stop".

Ferner sind alle zum System gehörige Tabellen „regelmäßig"; dies betrifft die oberen und unteren Teile der Tabellen, und bedeutet die folgenden Eigenschaften:

(d) In der (zum ersten Punkt der Tabelle gehörigen) ersten Zeile sind keine Leerstellen (da jede zur Bildung der Tabelle verwendete Linie von diesem Punkt ausging).

(e) Im oberen Teil der ersten Spalte folgen die nicht leeren Zeichen — die alle J-Zeichen sind — lückenlos auf einander. In der letzten Spalte — und nur in der letzten — kommen keine J-Zeichen vor.

(f) (1) Für jedes in Frage kommende i stimmt der Inhalt der $i+1$-ten Spalte mit dem Inhalt der i-ten Spalte bis zum letzten J-Zeichen der letzteren überein, doch statt diesem J steht N in der $i+1$-ten Spalte, (2) ferner gehört das nach diesem N eventuell noch vorhandene erste nicht leere Zeichen des oberen Teils der $i+1$-ten Spalte zur ersten solchen Zeile, die von der 1-ten bis zur i-ten Spalte weder J-noch N-Zeichen enthält (da nach der Abzweigung einer neuen Linie von der früheren lauter neue Punkte von der neuen Linie durchlaufen werden); und im damit beginnenden Stück des oberen Teils der $i+1$-ten Spalte folgen die nicht leeren Zeichen — die alle J-Zeichen sind — lückenlos auf einander.

[3] Siehe R. PÉTER: *Die prinzipielle Ausschaltbarkeit der rekursiven Prozeduren aus der Programmierungsprache Algol 60.* Acta Cybernetika **1** (1972) S. 219—231.

(g) Die zu berücksichtigenden (nicht zu leeren Zeichen gehörigen) Fragen je einer Spalte, und die eventuell nach leeren unteren Teilen in anderen Tabellen sich zu diesen anschließenden Fragen sind (zufolge der in § 5 angegebenen Normierung des Graphen) verschieden. (Diese Eigenschaft betrifft auch die Ausgänge.)

(h) Im unteren Teil jeder Spalte folgen die X-Zeichen lückenlos auf einander; und zwar in der ersten Spalte gleich von der ersten Zeile an, und für jedes in Frage kommende i in der $i+1$-ten Spalte von der Zeile unter dem letzten X im unteren Teil der i-ten Spalte an (da die mathematischen Punkte der zur Bildung der Spalten verwendeten Linien alle verschieden sind).

Auf Grund eines regelmäßigen Tabellensystems kann leicht ein zum selben Ergebnis führendes Vorgraphschema hergestellt werden.

Zuerst hat man für jede Tabelle T_i des Systems die dazu gehörigen logischen bzw. mathematischen Punkte $P_{i,1}, \ldots$; $P'_{i,1}, \ldots$ mit den ihnen zugeordneten Fragen bzw. Anweisungen aufzunehmen.

Dann hat man, die Spalten von T_i nach einander betrachtend, diese Punkte durch die entsprechenden Kanten zu verbinden: Ist das j-te Zeichen einer Spalte J dzw. N, und kam das nächste nicht leere Zeichen (vor dem Ausgang) dieser Spalte in ber k-ten Zeile vor, so hat man aus $P_{i,j}$ in den zur k-ten Zeile gehörigen Punkt eine J-Kante bzw. eine N-Kante zu ziehen; ist sowohl das j-te als auch das $j+1$-te Zeichen im unteren Teil einer Spalte X, so hat man aus $P'_{i,j}$ in $P'_{i,j+1}$ eine Kante uz ziehen.

Kommt vor dem Ausgang das letzte Zeichen J, N oder X einer Spalte in der j-ten Zeile vor, so hat man, falls der Ausgang der Spalte nicht „stop", sondern „go to T_k", oder „go to $T_{k,l}$", oder aber „go to $T'_{k,l}$" ist, vom zur j-ten Zeile gehörigen Punkt eine J-Kante, bzw. N-Kante, bzw. unbezeichnete Kante in $P_{k,1}$ bzw. $P_{k,l}$ bzw. $P'_{k,l}$ zu ziehen.

Damit wurde das gewünschte Vorgraphschema hergestellt.

§ 8

Die in der Praxis (und in der Literatur) verwendeten Tabellen sind aber allgemein nicht regelmäßig.

Gemäß der letzten Bemerkung des § 6 ist die in (a) und (b) des § 7 formulierte Regelmäßigkeit der Verbindungen unter mehreren Tabellen nicht unerläßlich. Irgendwie werden diese Verbindungen immer angegeben; und geschieht dies sinnvoll, so kann es immer zu Ausgangsteile der Tabellen im in dieser Arbeit eingeführten Sinn umformuliert werden.

Auch die — in der Praxis meistens nicht erfüllte — Forderung (c) kann fallen gelassen werden, nach welcher der entsprechende Graph einen einzigen Endpunkt enthalten müßte. Gibt es mehrere Punkte im Graphen, aus welchen keine Kante hinausführt, so kann dieser nur ein Teil eines Graphschemas sein (zu einem solchen kann aber immer auch ein denselben Zwecken entsprechendes Graphschema konstruiert werden), doch auch das Wirken solcher Teile eines Graphschemas kann auf Programmierungsprachen übersetzt werden.

Aus ähnlichen Gründen kann auch die — nicht immer erfüllte — Forderung, daß in einen gewissen Punkt (Input) keine Kante führen soll, fallen gelassen werden.

Jedenfalls hat man sich auf solche Tabellensysteme zu beschränken, die über die Eigenschaft (g) des § 7 verfügen.

Nach den Vorherigen hat man sich um die Ausgänge der üblichen Tabellen nicht mehr zu kümmern. Im übrigen Teil kann aber jede übliche (nur verschiedene Fragen und Anweisungen aufzeichnende) Tabelle, die keine Spalten mit gleichem oberen Teil enthält (auch „implizite" nicht, in einem sobald zu erklärenden Sinn), durch eine — dasselbe leistende — regelmäßige Tabelle vertreten werden.

Der untere Teil einer üblichen Tabelle T_i kann leicht regelmäßig gemacht werden. Nehmen wir an, daß in diesem Teil die Anzahl der X-Zeichen in der ersten Spalte x_1, in der zweiten x_2, usw., in der letzten Spalte x_t ist. Dann sind (mathematische) Punkte

$$P'_{i,1}, \ldots, P'_{i,x_1}, P'_{i,x_1+1}, \ldots, P'_{i,x_1+x_2}, \ldots, \ldots, P'_{i,x_1+x_2+\ldots+x_t}$$

und in dieser Reihenfolge mit diesen bezeichnete Zeilen — statt der früheren Zeilen des unteren Teils von T_i — aufzunehmen, die X-Zeichen der ersten Spalte, samt der zu ihnen gehörigen Anweisungen der Reihe nach in die zu $P'_{i,1}, \ldots, P'_{i,x_1}$ gehörigen Zeilen, die X-Zeichen der zweiten Spalte, samt der zu ihnen gehörigen Anweisungen (worunter auch mit vorher aufgetretenen gleiche vorkommen können) der Reihe nach in die zu $P'_{i,x_1+1}, \ldots, P'_{i,x_1+x_2}$ gehörigen Zeilen einzutragen, usw.

Dadurch wird (h) des § 7 erfüllt sein, und man hat sich nur noch um die oberen Teile der üblichen Tabellen zu kümmern.

Betreffend der oberen Teile der üblichen Tabellenspalten ist es Brauch ein Leerzeichen so zu betrachten, daß die zur Spalte gehörigen Anweisungen von der entsprechenden Frage unabhängig, also bei Antworten „Ja" und „Nein" auf diese Frage dieselben sind. Daher ist es üblich die betrachtete Spalte durch zwei andere zu ersetzen, die von ihr nur darin abweichen, daß in der ersten J, in der zweiten N für das betrachtete Leerzeichen gesetzt wird. So könnte vorkommen, daß der obere Teil einer der neuen Spalten mit dem oberen Teil einer alten Spalte übereinstimmt.

Deshalb muß die „wesentliche Abweichung" der oberen Teile je zweier Spalten verlangt werden, d. h. daß für je zwei Spalten mindestens eine Zeile geben soll, worin eine der Spalten J, die andere N enthält. Gilt dies, so enthält die Tabelle auch „implizite" keine Spalten mit gleichem oberen Teil.

Ferner ist es Brauch auch neue Spalten mit einer einheitlichen, etwa durch „error" bezeichneten Anweisung zu einer Tabelle hinzuzunehmen, um zu betonen, daß die in einer solchen Spalte gegebene Variation der Antworten auf die aufgeworfenen Fragen für das gesetzte Ziel nicht in Frage kommt. Man könnte auch statt der Aufnahme einer Anweisung „error", nach leerem unteren Teil im Ausgang eine Rückkehr der zur Spalte gehörigen letzten Kante zu ihrem Ausgangspunkt vorschreiben, wodurch ein ewiger Kreislauf bewirkt würde; dieser würde dann zeigen, daß das Ergebnis des durch die Tabelle veranschaulichten Verfahrens für die betreffende Variation undefiniert ist.

(Es bedeutet eine andere Situation, wenn für die ursprünglich auch implizite nicht in die oberen Teile der Spalten einer Tabelle aufgenommenen Variationen einheitliche Anweisungen bisheriger Art angegeben werden. Für diese ist es üblich eine Spalte mit der Aufschrift „else" zur Tabelle hinzuzunehmen, was aber nur

eine kürzere Schreibweise ist für Spalten mit oberen Teilen, welche die genannten Variationen enthalten, und mit gleichen unteren Teilen und Ausgängen.)

Mit Aufnahme neuer Spalten kann der obere Teil jeder üblichen Tabelle, nach Behebung auf der vorhin geschilderten Art der Leerzeichen, so ergänzt werden, daß in den oberen Teilen der Spalten alle mögliche J, N-Variationen vorkommen (bei Fragen der Anzahl n ist ihre Anzahl 2^n).

Werden diese Variationen — wie üblich — so nach einander gebildet, daß am Anfang J so lange wie möglich beibehalten wird, und wird die Reihenfolge der Spalten demgemäß modifiziert, das entspricht gerade der möglichst linkseitigen Wahl jener Linien, gemäß welchen aus einem Teilgraph eines Vorgraphschemas die Spalten der entsprechenden Tabelle gebildet wurden. Man hat nur noch für das Bestehen von (f) (2) des § 7 zu sorgen (nämlich für die Spiegelung der Tatsache, daß nach jeder Wahl der N-Kante eines Verzweigungspunktes in der Bildung der genannten Linien lauter neue Punkte vorkommen).

Das geht leicht. Betrachten wir z. B. den Fall von 3 verschiedenen Fragen F_1, F_2, F_3. Der obere Teil der alle Variationen in der genannten Reihenfolge enthaltenden üblichen Tabelle ist:

F_1	J	J	J	J	N	N	N	N
F_2	J	J	N	N	J	J	N	N
F_3	J	N	J	N	J	N	J	N

Um (f) (2) zu erfüllen, bildet man daraus leicht den folgenden (dieselben Variationen enthaltenden) Tabellenteil:

F_1	J	J	J	J	N	N	N	N
F_2	J	J	N	N				
F_3	J	N						
F_3			J	N				
F_2					J	J	N	N
F_3					J	N		
F_3							J	N

Das ist bereits der obere Teil einer regelmäßigen Tabelle.

§ 9

In den einzelnen Spezialfällen müssen nicht unbedingt alle Leerstellen ausgefüllt, alle Variationen gebildet werden; in der Praxis strebt man auf möglichst einfache Übergänge zu einem entsprechenden flow-diagram.

Batrachten wir z. B. eine zur Betriebsorganisation verwendete Entscheidungstabelle, die auf S. 19 des in Fußnote[2] zitierten Buches angegeben wird. Diese kann mit den Bezeichnungen

$$F_1, F_2, F_3, F_4 \quad \text{bzw.} \quad A_1, A_2, A_3, A_4$$

für die darin enthaltenen Fragen bzw. Anweisungen (deren Bedeutung für unsere Untersuchungen belanglos ist) wie folgt aufgezeichnet werden:

F_1	J	J		N
F_2	J	N	N	J
F_3				N
F_4		J	N	
A_1	X			
A_2				X
A_3		X		
A_4			X	

Es sind die in § 7 angegebenen Eigenschaften der regelmäßigen Tabellen vor Augen zu halten.

Erstens sieht man, daß wegen der leeren Stelle in der ersten Zeile (d) nicht erfüllt ist. Deshalb sollen statt der dritten Spalte zwei neue Spalten aufgenommen werden (hier hätte man auch die ersten beiden Zeilen vertauschen können):

F_1	J	J	J	N	N
F_2	J	N	N	N	J
F_3					N
F_4		J	N	N	
A_1	X				
A_2					X
A_3		X			
A_4			X	X	

In der 4-ten Spalte kommt kein J-Zeichen vor, so wird (e) nicht erfüllt. Daran kann durch Vertauschung der beiden letzten Spalten geholfen werden:

F_1	J	J	J	N	N
F_2	J	N	N	J	N
F_3				N	
F_4		J	N		N
A_1	X				
A_2				X	
A_3		X			
A_4			X		X

In der 4-ten und 5-ten Spalte wird die letzte Forderung von (f) (2) nicht erfüllt (nämlich, daß nach einer „Abzweigung" lauter J-Kanten auf dem Anfangsteil der zu einer Spalte gehörigen Linie auftreten sollten). Deshalb soll je eine neue Spalte

mit der Anweisung „error" eingeschaltet werden:

F_1	J	J	J	N	N	N	N
F_2	J	N	N	J	J	N	N
F_3				J	N		
F_4		J	N			J	N
A_1	X						
A_2					X		
A_3		X					
A_4			X				X
error				X		X	

Jetzt sind noch die nicht leeren Zeichen je einer Spalte in die durch (f) (2) und (h) vorgeschriebenen Zeilen zu rücken (samt den zu ihnen gehörigen Frage- bzw. Anweisug-Zeichen):

P_1	F_1	J	J	J	N	N	N	N
P_2	F_2	J	N	N				
P_3	F_4		J	N				
P_4	F_2				J	J	N	N
P_5	F_3				J	N		
P_6	F_4						J	N
P'_1	A_1	X						
P'_2	A_3		X					
P'_3	A_4			X				
P'_4	error				X			
P'_5	A_2					X		
P'_6	error						X	
P'_7	A_4							X

Das ist schon eine regelmäßige Tabelle; mit 7 und nicht $2^4=16$ Spalten, wieviele zu bilden wären, wenn für die 4 verschiedenen Fragen alle Antworte-Variationen aufgenommen würden.

Von dieser Tabelle sind die von P_1 ausgehenden Linien des entsprechenden Vorgraphschemas Spalte für Spalte verfolgend unmittelbar abzulesen. So ergibt sich:

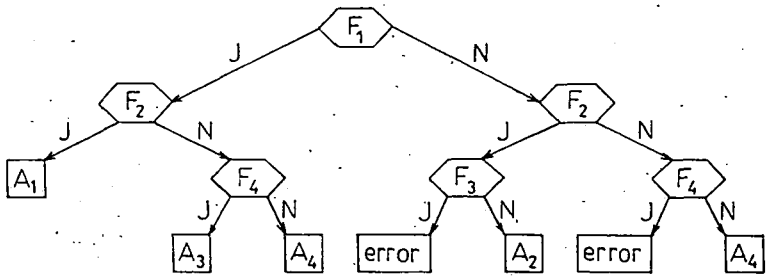

Abb. 7

§ 10

Ich bin noch mit der zu-Graphschema-Präzisierung des auf Grund des in § 1 gegebenen Beispiels ohne exakte Definition benutzten Vorgraphschema-Begriffes schuldig.

Die Struktur des zum Graphschema gehörigen Graphen — samt den J, N-Bezeichnungen gewisser Kanten — wurde bereits in § 1 exakt definiert. Doch zu einem Graphschema G gehört auch eine Menge M. Jedem logischen Punkt wird eine „logische Funktion" zugeordnet; d. h. eine Funktion, die auf einer Teilmenge von M definiert ist, und logische Werte: „wahr" oder „falsch" annimmt (diese ist eigentlich eine Relation zwischen ihren Argumenten; ob sie für eine Stelle wahr oder falsch ist, bedeutet, daß auf die Frage: „Besteht hier diese Relation?" — die Antwort „Ja" bzw. „Nein" ist). Jedem mathematischen Punkt ist eine „mathematische Funktion" zugeordnet; d. h. eine Funktion, die auf einer Teilmenge von M definiert ist, und auch als Werte Elemente von M annimmt (diese kann für jede Stelle als eine Anweisung zur Berechnung des Funktionswertes betrachtet werden; welcher Wert dann als Argument für die nachfolgende Funktion dient).

Durch das Graphschema wird eine mathematische Funktion definiert, und zwar auf folgende Weise:

Ist ein $m \in M$ gegeben, dann soll dem Input I der für m angenommene Wert der zu I gehörigen Funktion zugeordnet werden. Ist I ein mathematischer Punkt, und wurde ihr derart m_1 zugeordnet, so hat man auf der einzigen von ihm auslaufenden Kante zum nächsten Punkt zu gehen, und diesem Punkt den für m_1 angenommenen Wert der zu ihm gehörenden Funktion zuzuordnen. Ist I ein logischer Punkt, so gehen wir auf der daraus hinauslaufenden J- oder N-Kante zum nächsten Punkt, je nachdem dem I „wahr" oder „falsch" zugeordnet wurde; und diesem nächsten Punkt ordnen wir dann den für m angenommenen Wert der dazu gehörigen Funktion zu. Aus dem erreichten Punkt gehen wir ebenso zum nächsten Punkt weiter, usw. Gelangen wir zu einem Punkt, welcher bereits früher ein Wert zugeordnet wurde, dann soll dieser Wert im Sinne der vorangehenden Vorschrift abgeändert werden. Es ist möglich, daß das Verfahren in endlich vielen Schritten, noch bevor man das Output O erreicht, stecken bleibt: z. B. gleich beim ersten Schritt, falls die zum Input gehörige Funktion für m nicht definiert ist. Es kann auch vorkommen, daß ein Kreisweg unendlich oft beschrieben werden muß. Gelangt man aber in endlich vielen Schritten zum Output O, mit einem Wert, für den die zu O gehörige (jedenfalls mathematische) Funktion definiert ist, so erhält man hier eindeutig einen Wert $m^* \in M$. Dieses m^* gilt als Wert an der Stelle m der durch G definierten Funktion. Könnte nun das — in den folgenden kurz durch (Vg) bezeichnete — Vorgraphschema des § 1 zu ein Graphschema präzisiert werden?

Jedenfalls sollten ins Input die Daten hineinkommen, also k und die binären Ziffern der Summanden

$$a_0, a_1, \ldots, a_k; \quad b_0, b_1, \ldots, b_k.$$

In der n-ten Phase der Berechnung wird außer a_n und b_n auch der Rest r der vorangehenden Ziffernaddition betrachtet. Bis zur Bestimmung des gesuchten s_k werden als Zwischenwerte s die früheren Ziffern der Summe auftreten. Für die genannten, sich während der Berechnung verändernden Werte werden Hilfsvariablen r, n und s eingeführt. Als nächste Information sind Anfangswerte für die Hilfs-

variablen anzugeben. In der $n=0$-ten Phase ist $r=0$, und für den noch nicht vorhandenen Wert von s kann z. B. auch 0 gewählt werden.

Es werden also in den einzelnen Schritten Wertefolgen

$$(k, a_0, \ldots, a_k, b_0, \ldots, b_k)$$

und

$$(k, a_0, \ldots, a_k, b_0, \ldots, b_k, r, n, s)$$

eine Rolle spielen; speziell in der 0-ten Phase die Folge

$$(k, a_0, \ldots, a_k, b_0, \ldots, b_k, 0, 0, 0).$$

Zur Menge M des Graphschemas müssen daher gewisse endliche Zahlenfolgen gehören; es zeigt sich auch weiter, daß für M zum Beispiel die Menge der höchstens $2k+6$-gliedrigen Folgen natürlicher Zahlen (darunter auch selbst die natürlichen Zahlen als 1-gliedrige Folgen) gewählt werden kann.

Das Input muß ein mathematischer Punkt sein, zu dem jene Funktion f_1 gehört, die einer Folge

$$(k, a_0, \ldots, a_k, b_0, \ldots, b_k)$$

die Folge

$$(k, a_0, \ldots, a_k, b_0, \ldots, b_k, 0, 0, 0)$$

zuordnet.

Zu Beginn von (Vg) wurden nur den Hilfsvariablen r und n Anfangswerte gegeben; die einzutragenden Daten und die Hilfsvariable s (als s_n) wurden nur verheimlicht in das Schema eingeschmuggelt.

Vom neuen Input führt die einzige Kante mit dem erhaltenen Wert zum ersten logischen Punkt, wo im (Vg) die Frage „$r=0$?“ steht. Für diese Frage ist folgende logische Funktion F_1 einzusetzen:

$$F_1(k, a_0, \ldots, a_k, b_0, \ldots, b_k, r, n, s) = \begin{cases} \text{J}, & \text{falls } r=0 \\ \text{N}, & \text{falls } r\neq 0. \end{cases}$$

(In der 0-ten Phase gilt natürlich $r=0$; aber es führt auch eine andere Kante in diesen Punkt — diese kann auch andere Werte mit sich bringen.)

An den von hier ausgehenden Kanten läuft jede hierher eingetroffene Folge in je einen solchen logischen Punkt, dem die logische Funktion

$$F_2(k, a_0, \ldots, a_k, b_0, \ldots, b_k, r, n, s) = \begin{cases} \text{J}, & \text{falls } a_n=b_n \\ \text{N}, & \text{falls } a_n\neq b_n \end{cases}$$

zugeordnet wird.

Betrachten wir den auf der J-Kante erreichten Punkt. Je nachdem die hier eingetroffene Folge daraus an der J- oder N-Kante weiterläuft, gelangt sie zu einem mathematischen Punkt, dem die (den Anfangswert 0 bzw. 1 für s einführende) mathematische Funktion

$$f_2(k, a_0, \ldots, a_k, b_0, \ldots, b_k, r, n, s)=(k, a_0, \ldots, a_k, b_0, \ldots, b_k, r, n, 0)$$

bzw.

$$f_3(k, a_0, \ldots, a_k, b_0, \ldots, b_k, r, n, s)=(k, a_0, \ldots, a_k, b_0, \ldots, b_k, r, n, 1)$$

zuzuordnen ist.

Auch weiter geht das ähnlich. Z. B. wird dem untersten rechts liegenden Punkt von (Vg) mit einem bestimmten Index i die mathematische Funktion

$$f_i(k, a_0, \ldots, a_k, b_0, \ldots, b_k, r, n, s)=(k, a_0, \ldots, a_k, b_0, \ldots, b_k, r, n+1, s)$$

zugeordnet.

Doch dem Output O wird folgende mathematische Funktion zugeordnet:

$$f_O(k, a_0, \ldots, a_k, b_0, \ldots, b_k, r, n, s)=s.$$

Dann ist der Wert der durch das erhaltene Graphschema definierten Funktion an der Stelle

$$(k, a_0, \ldots, a_k, b_0, \ldots, b_k)$$

(die aus den im Input eingegebenen Daten besteht) der zum Schluß erhaltene Wert von s. Man kann nachprüfen, daß dies die gesuchte k-te binäre Ziffer der in Frage stehenden Summe ist.

Die Fragen bzw. Anweisungen eines in der Praxis sinnvoll konstruierten flow-diagram's können (eventuell durch eine geeignete Kodierung) immer durch logische bzw. mathematische Funktionen im vorher definierten Sinn vertreten werden.

§ 11

Das in § 10 erhaltene Graphschema ist bereits fast ein „Normalschema", und kann leicht durch ein Normalschema ersetzt werden, das dieselbe Funktion definiert.

Dabei ist ein Normalschema ein derartiges Graphschema, dessen Punkten nur „Anfangsfunktionen" zugeordnet sind; wobei durch jede mathematische Anfangsfunktion f endliche Folgen natürlicher Zahlen gegebener Gliederzahl u in ebenfalls derartige Folgen gegebener Gliederzahl v übertragen werden:

$$f(n_1, \ldots, n_u)=(m_1, \ldots, m_v)$$

wo jedes m_i $(i=1, 2, \ldots, v)$ entweder mit einem der n_j oder n_j+1 $(j=1, 2, \ldots, u)$ übereinstimmt, oder aber 0 ist; ferner jede logische Anfangsfunktion für Folgen natürlicher Zahlen gegebener Gliederzahl u definiert und der Form

$$F(n_1, \ldots, n_u) = \begin{cases} \text{J}, & \text{falls } m_1=m_2 \\ \text{N}, & \text{falls } m_1 \neq m_2 \end{cases}$$

ist, wo sowohl m_1 als auch m_2 mit einem der n_i $(i=1, 2, \ldots, u)$ übereinstimmt.

Nach den Vorherigen gibt es auch zu jedem Normalschema ein regelmäßiges Tabellensystem, das zum selben Ergebnis führt. In diesem treten statt Fragen logische Anfangsfunktionen und statt Anweisungen mathematische Anfangsfuntionen auf. Solche Tabellensysteme können „Normaltabellensysteme" genannt werden.

Ich habe bewiesen[1], dass die durch Normalschemata definierbaren Funktionen mit den sogenannten „partiell-rekursiven" Funktionen identisch sind. Nach den Bisherigen gilt dasselbe auch für die durch Normaltabellensysteme definierbaren Funktionen.

Die Wirkung eines Komputers kann auch so betrachtet werden, daß man gewisse Daten einträgt, und dann von diesen abhängig gewisse Ergebnisse davon herauskommen. Da sowohl die eingetragenen Daten als auch die Folge der herauskommenden Ergebnisse durch natürliche Zahlen kodiert werden können, kann eigentlich das Wirken eines Komputers immer als die Berechnung der Werte einer zahlentheoretischen Funktion betrachtet werden. Wird für die Zelleninhälte keine Schranke gestellt, so kann man beweisen, daß die durch Komputer berechenbaren zahlentheoretischen Funktionen mit den partiell-rekursiven Funktionen identisch sind[4]. So sind diese nach den Vorherigen auch mit den durch Normaltabellensysteme berechenbaren Funktionen identisch. Man kann sagen: was die Komputer können, das ist im wesentlichen dasselbe, als das, was durch Normaltabellensystemen erreicht werden kann.

Математическое понятие так называемых «таблиц решения»

В последнее время на практике вошло в привычку наиболее сложные части "flow" диаграмм заменять более легко и отдельно разрешимыми «таблицами решения» (decision tables), хотя оба понятия используются без сторого го определения. Для" flow" диаграмм уже раньше Калзнин ввел математическое определение с названием «графной схемы». Автор доказал, что эти функции теории чисел, определяемые частным случаем «нормальной схемы» — которая содержит простейшие основные функции — тождественны частичным рекурсивным функциям. Настоящая статья содержит методы преобразования «графных схем» в «правильные табличные системы», и обратного преобразования «правильных табличных систем» в «графную схему». Последний имеет большое значение при трансляции на некоторый програмный язык. Также описывается метод преобразования целесообразно заданных «таблиц решения» в правильные. Освещается, что «нормальные табличные системы», соответствующие нормальным схемам, в действительности, служат для использования возможностей вычислительных машин.

EÖTVÖS LORÁND UNIVERSITÄT
BUDAPEST

(Eingegangen am 19. September 1972)

[4] Siehe J. C. SHEPHERDSON und H. E. STURGIS: *Computability of Recursive Functions.* Journ. of the ACM **10** (1963) S. 217—255 und R. PÉTER: *Programmierung und partiell-rekursive Funktionen.* Acta Math. Ac. Sci. Hung. **14** (1963) S. 373—401; ferner R. PÉTER: *Automatische Programmierung zur Berechnung der partiell-rekursiven Funktionen.* Studia Sci. Math. Hung. **4** (1969) S. 447—463. Siehe auch das Buch (in Vorbereitung): R. PÉTER: *Rekursive Funktionen in der Komputer-Theorie.*

On the computation of union-extensions of finite semigroups

By R. BRÖCK and H. JÜRGENSEN

In his dissertation of 1968 [3] Verbeek proposed a generalization of the theory of semigroup extensions, which until that date consisted of the two nearly disjoint parts of Schreier- and ideal-extensions. According to Verbeek we define a semigroup extension as follows:

Definition 1. Let A, S, E be semigroups and δ a congruence on E. The pair (E, δ) is a semigroup extension of A by S, iff $E/\delta \cong S$ and there is a subsemigroup A' of E, isomorphic to A, which is a δ-class.

In the rest of this paper we shall often say that some semigroup E is an extension of A by S in the sense that there is a congruence δ, such that (E, δ) is a semigroup extension of A by S.

Schreier- and ideal-extensions are semigroup extensions according to this definition. Verbeek proved that there is an extension of A by S, iff S contains an idempotent element. Thus for finite S there is always an extension of arbitrary A by S. The idempotent concerned is the image of A' in S and is called the extension idempotent.

For ideal-extensions the homomorphism δ_{nat} induced by δ is a very special one: it is a bijection of $E \setminus A'$. Generalization of this idea led Verbeek to the concept of union-extensions:

Definition 2. Let A and S be semigroups, (E, δ) a semigroup extension of A by S. (E, δ) is a union-extension of A by S, iff the restriction of δ to $E \setminus A'$ is the identity relation, where A' is as in definition 1.

As for ideal-extensions for finite A and S the set of all union-extensions (up to isomorphism) may be obtained in a rather simple way.

Theorem 1. (Verbeek). Let A, S be disjoint semigroups, $i \in S$ an idempotent element. For $E = A \cup S^-$, where $S^- = S \setminus \{i\}$, define an associative multiplication $*$ such that the following conditions hold for all $a, b \in A$, $s, t \in S^-$

$$a * b = ab, \tag{1}$$

$$a * s \begin{cases} = is & \text{if } is \neq i, \\ \in A & \text{if } is = i, \end{cases} \tag{2}$$

$$s*a\begin{cases}=si & \text{if } si\neq i,\\ \in A & \text{if } si=i,\end{cases} \tag{3}$$

$$s*t\begin{cases}=st & \text{if } st\neq i,\\ \in A & \text{if } st=i.\end{cases} \tag{4}$$

Then $((E, *), \delta)$ is a union-extension of A by S for

$$\delta=A\times A\cup\{(x, x)|x\in S^{-}\}.$$

Moreover, any union-extension (E', δ') of A by S is isomorphic to one constructed in this way, where i is the extension idempotent.

Theorem 1 indicates a combinatorial method of computing the set of all union-extensions of A by S (disjoint) with extension idempotent i as follows. For A and S both finite, given by their Cayley-tables T^A and T^S, consider column c_i and row r_i of i in S; the entry t^S_{ii} belonging to ii will be replaced by A; the rest of c_i and r_i will be copied $|A|$ times to obtain a full table again; then, wherever it appears, i will be replaced by a cross indicating that the corresponding position is unknown; call the resulting partial table $T^{A,S}$.

Example

T^A	a	b
a	a	b
b	b	b

T^S	s	t	i	u	v
s	t	i	s	s	s
t	i	t	t	t	t
i	s	t	i	i	i
u	s	t	i	u	i
v	s	t	i	i	v

$T^{A,S}$	s	t	a	b	u	v
s	t	+	s	s	s	s
t	+	t	t	t	t	t
a	s	t	a	b	+	+
b	s	t	b	b	+	+
u	s	t	+	+	u	+
v	s	t	+	+	+	v

One obtains all union-extensions of A by S with extension idempotent i by replacing the crosses in $T^{A,S}$ by elements of A in all possible ways, such that the resulting table will be associative. Of course, this purely combinatorial method would soon lead to enormous computing time.

A solution to this problem is indicated by Verbeek's discussion of the composition of S with respect to i and by his theorems on the existence of union-extensions of A by S, when S has some special composition. The set of all possible compositions of semigroups has been described in parts by Verbeek [3, 4] and fully by van Leeuwen; unfortunately, he published his results in an abstract [2] only up to now.

We took a quite different and a rather naive way for computing the set of all union-extensions of A by S with extension idempotent i; all the same the computing time needed is very well below the time for the purely combinatorial method, at least when the number of extensions is small compared to the number of tables to be checked.

For $x, y\in A\cup S^{-}$ let $x*y$ be undefined in $T^{A,S}$. This entry of $T^{A,S}$ is considered as an unknown $u_{x,y}$ over A. Then by associativity one has a set G of equations over $A\cup S^{-}$ with unknowns $u_{x,y}$ over A such that exactly the solutions of G are

the allowable ways of replacing the crosses in $T^{A,S}$. We classify the equations according to their forms as follows:

$$G_1 = \{x * u_{y,z} = u_{x,y} * z\} \quad x, z \in A, \quad G_6 = \{u_{x,u_{y,z}} = u_{u_{x,y},z}\},$$
$$G_2 = \{x * u_{y,z} = u_{xy,z}\} \quad x \in A, \quad G_7 = \{u_{x,u_{y,z}} = u_{xy,z}\},$$
$$G_3 = \{u_{x,yz} = u_{x,y} * z\} \quad z \in A, \quad G_8 = \{u_{x,yz} = u_{u_{x,y},z}\},$$
$$G_4 = \{x * u_{y,z} = u_{u_{x,y},z}\} \quad x \in A, \quad G_9 = \{u_{x,yz} = u_{xy,z}\}.$$
$$G_5 = \{u_{x,u_{y,z}} = u_{x,y} * z\} \quad z \in A,$$

It is the aim of the following method for solving G to successively narrow the domains of the unknowns and thus to avoid unnecessary trials.

We denote the domain of the unknown u by $D(u)$. In the computer programme the set of the $D(u)$ is realized by an $n \times |A|$-integer-array DOM, where n is the number of unknowns, such that

$$DOM_{u,a} = \begin{cases} 0 & \text{if } a \notin D(u), \\ 1 & \text{if } a \in D(u). \end{cases}$$

To enable an easy test, whether G has been solved, we put $|D(u)| = \sum_{a \in A} DOM_{u,a}$ in another array, which of course will be changed whenever DOM is changed. In the beginning all the $D(u)$ are A, i.e. $DOM_{u,a} = 1$ for all u and all $a \in A$.

Step 1 consists of evaluating each of the equations in $K_1 = G_1 \cup G_2 \cup G_3$. An equation $x * u_{y,z} = u_{x,y} * z$ in G_1 leads to $xD(u_{y,z}) = D(u_{x,y})z$, which, however, will not be valid in most cases. Clearly there is a solution $u_{y,z} = w_1 \in D(u_{y,z})$, $u_{x,y} = w_2 \in D(u_{x,y})$ to the equation only, if

$$xw_1 \in D = D(u_{x,y})z \cap xD(u_{y,z}) \ni w_2 z.$$

Hence we can cancel all those $w_1 \in D(u_{y,z})$ $(w_2 \in D(u_{x,y}))$ in DOM, for which $xw_1 \notin D$ $(w_2 z \notin D)$ and thus narrow the domains $D(u_{x,y})$ and $D(u_{y,z})$. Furthermore, all equations from G_9 in which $u_{x,y}$ $(u_{y,z})$ appears lead to restrictions; let $u_{x,y} = u$ be such an equation; then $D(u)$ will be narrowed to $D(u_{x,y})$. For the equations in G_2 or G_3 one proceeds analogously. Some special cases arise when x and (or) z are (one-sided) identity- or zero-elements of A; they may result in transferring the corresponding equation to another type G_j (e.g. to G_9 if $x = z$ is the identity-element of A).

Since a change of $D(u)$ for an unknown u might lead to consequences from equations which have already been evaluated, step 1 is repeated until there is no $D(u)$ that can be narrowed any more.

Performing step 1 might result in one of the following three situations; otherwise we continue with step 2.

(1) For each u, $|D(u)| = 1$. Then DOM represents the only solution of G.

(2) For some u, $|D(u)| = 0$. Then G has no solution.

(3) For some u, $|D(u)| = 1$. Wherever u appears in equation $e \in K_2 = G_4 \cup G_5 \cup G_6 \cup G_7 \cup G_8$ as a subscript of an unknown, it is replaced by its unique value. As a consequence in most cases e must be transferred to another class G_j. If by this procedure K_1 or G_9 is extended, e is evaluated and if this results in a restriction of some $D(u)$ execution of step 1 is resumed; otherwise step 2 is started.

In *step* 2 combinatorics comes in. G, *DOM* and all other information relevant to the situation are saved. Then for one unknown u we assume $u=a$ for arbitrary $a \in D(u)$, i.e. restrict $D(u)$ to be $\{a\}$ in *DOM*, and try to solve G applying step 1 again. With G, *DOM* etc. restored this is repeated until $D(u)$ is exhausted. Evidently in this way we compute exactly the set of solutions of G.

Some care has to be taken with the choice of u in step 2. It is chosen in such a way that changing $D(u)$ is likely to induce changes of the domains of as many other unknowns as possible; hence, with priority as stated, the following criteria are applied:

(1) The number of unknowns u is equal to by equations in G_9 (using transitivity, too) is maximal.

(2) The number of equations in K_2, in which u appears as a subscript, is maximal.

(3) $|D(u)|$ is maximal.

The algorithm has been realized as an ALGOL 60 programme [1] and is run on an ELECTROLOGICA X8 computer (cycle time 2.5 μsec).

Whereas it is evident that for the combinatorial method the time is $\geqq O(n^{|A|})$, where n is the number of unknowns, it seems to be impossible to give a rather correct estimate for our method; it is bad, of course, when the number of union extensions is approximately $n^{|A|}$; but in this case any method should be bad. The

Table 1

Example No.	1	2	3	4	5	6	7	8
|A|	3	3	4	4	5	5	6	6
|S|	2	5	2	5	2	5	2	5
with ideal-extensions	yes	no	yes	no	yes	no	yes	no
unknowns	6	16	8	20	10	24	12	28
combinations	729	$>4\cdot10^7$	65 536	$>10^{13}$	$>9\cdot10^6$	$>5\cdot10^{16}$	$>2\cdot10^9$	$>6\cdot10^{21}$
union-extensions	26	163	4	15	8	3	16	0
our time	20 s	5.5 m	11 s	80 s	25 s	17 s	67 s	11 s
time for combinatorial method	14 s	≈140 h	≈10 m	≈870 years	≈30 h	$\approx5\cdot10^7$ years	≈300 days	$\approx3\cdot10^{12}$ years

following table 1 allows a comparison of actual computing times; of course the figures in the last line can be considered just as hints to the approximate size, since they were calculated from the state of the pogramme after a short run only. The corresponding semigroups are listed in table 2.

Table 2

Example No.	1	2	3	4	5	6	7	8
Semigroups	A_1+S_1	A_1+S_2	A_2+S_1	A_2+S_2	A_3+S_1	A_3+S_2	A_4+S_1	A_4+S_2
Multiplication tables	S_1 | *ab* *a* | *aa* *b* | *ab*	S_2 | *abcde* *a* | *abcaa* *b* | *bcabb* *c* | *cabcc* *d* | *abcda* *e* | *abcae*	A_1 | *xyz* *x* | *xxx* *y* | *xxx* *z* | *xxx*	A_2 | *wxyz* *w* | *wwww* *x* | *wxww* *y* | *wwyz* *z* | *wwzy*	A_3 | *vwxyz* *v* | *vvvvv* *w* | *vvvww* *x* | *xxxxx* *y* | *vvvyy* *z* | *vvxyz*	A_4 | *uvwxyz* *u* | *uuwwyz* *v* | *uuwwyz* *w* | *wwuuzy* *x* | *wwuuzy* *y* | *yyzzwu* *z* | *zzyyuw*		

The element a is the extension idempotent.

О получении с помощью вычислительной машины обьединённого расширения конечных полугрупп

В 1968. году Verbeek дал определение для понятия обьединённого расширения полугрупп —, как обобщение этого понятия для идеальных расширений.

Как и для идеального расширения, мы имеем простой алгоритм для получения на вычислительной машине семейства обьединённого расширений двух конечных полугрупп, но этот алгоритм требует большого количества машинного времени. Эта статья описывает один такой алгоритм, который в общем требует значительно меньшего времени, он реализован как программа на языке ALGOL—60.

INSTITUT FÜR INFORMATIK
UND PRAKTISCHE MATHEMATIK
DER UNIVERSITÄT KIEL

MATHEMATISCHES SEMINAR
DER UNIVERSITÄT KIEL
D—23 KIEL 1, GERMANY

References

[1] BRÖCK, R., *Ein Programm zur Berechnung aller Vereinigungserweiterungen zweier Semigruppen*, Diplomarbeit, Kiel 1972.
[2] VAN LEEUWEN, J., On compositions of semigroups, *Notices Amer. Math. Soc.*, v. 71T—A56, 1971, p. 405.
[3] VERBEEK, L. A. M., *Semigroup extensions*, Proefschrift, Delft, 1968.
[4] VERBEEK, L. A. M., Union extensions of semigroups, *Trans. Amer. Math. Soc.*, v. 150, 1970, pp. 409—423.

(Received December 15, 1972)

Funktionen, die von pushdown-Automaten berechnet werden

Von G. WECHSUNG

In der Literatur über formale Sprachen nehmen die pushdown-Automaten (PDA) als Akzeptoren kontextfreier Mengen einen hervorragenden Platz ein (vgl. z.B. [2]). Was die von PDA mit Ausgabe (pushdown transducer, [1], [3]) berechneten Wortfunktionen betrifft, so sind zwar umfangreiche Ergebnisse über das Verhalten formaler Sprachen verschiedener Typen bei Anwendung solcher Abbildungen bekannt (vgl. z.B. [3]), aber bisher fehlt eine automatenunabhängige Charakterisierung dieser Wortfunktionen. Diese Aufgabe wird in der vorliegenden Arbeit für den Fall deterministischer PDA, die überall definierte Funktionen berechnen, in Angriff genommen. In § 1 werden die nötigen Grundbegriffe und zwei verschiedene Berechnungsbegriffe für PDA eingeführt. § 2 gibt eine Charakterisierung der im ersten Sinne PDA-berechenbaren Wortfunktionen (die Berechnung ist beendet, wenn das Eingabewort abgearbeitet ist). Es ergibt sich eine Klasse sequentieller Funktionen, die in gewisser Weise aus sequentiellen Funktionen endlichen Gewichts stückweise zusammengesetzt werden können. Dabei entstehen auch Funktionen *unendlichen Gewichts*. Die genaue Beschreibung dieser Klasse ist zwar etwas aufwendig, aber begrifflich durchsichtig.

Für die Berechnungen im 2. Sinne (die Berechnung ist beendet, wenn Eingabewort und Speicher abgearbeitet bzw. geleert sind) reichen die sequentiellen Funktionen nicht mehr aus. Es erweist sich aber eine in § 3 beschriebene Verallgemeinerung der in [6] eingeführten quasisequentiellen Funktionen als geeignet. § 4 ist den „*treu ausspeichernden*" PDA gewidmet, wobei der Spezialfall der längentreuen Wortfunktionen eingangs gesondert betrachtet wird. Die Klasse der von treu ausspeichernden PDA im zweiten Sinne berechneten Funktionen ergibt sich als wohldefinierte Klasse quasisequentieller Wortfunktionen (Satz 6′ bzw. Satz 6 für den Spezialfall der längentreuen Funktionen). Diese Funktionen haben die Gestalt $\varphi \circ s$, wobei φ eine sequentielle Funktion einer bestimmten Art und s eine sogenannte quasisequentielle Wortpermutation ist und φ und s eine bestimmte Verträglichkeitsbedingung erfüllen müssen. Die größere Leistungsfähigkeit der PDA gegenüber den endlichen Automaten kommt hier dadurch zum Ausdruck, daß φ auch unendliches Gewicht haben kann, wobei freilich die Klasse der zugelassenen φ verhältnismäßig begrenzt ist. Schließlich wird in § 5 auf die Voraussetzung des treuen Ausspeicherns verzichtet. Die im 2. Sinne berechenbaren Wortfunktionen haben zwar auch hier noch die Gestalt $\varphi \circ s$, jedoch braucht φ nicht mehr sequentiell, sondern nur noch „fastsequentiell" zu sein. Die Klasse der zugelassenen φ wird genau beschrieben (Satz 6″).

§ 1 Pushdown — Automaten mit Ausgabe

Wir interessieren uns hier nur für die folgende deterministische Variante des in [3] definierten pushdown transducers, die durch die Definitionen 1 und 2 präzisiert wird. In jedem Takt führt der Automat in Abhängigkeit vom inneren Zustand, vom gelesenen Eingabebuchstaben und vom letzten Kellerbuchstaben in eindeutig bestimmter Weise folgende Operationen durch: Er ändert den Zustand, ersetzt den gelesenen Eingabebuchstaben durch #, entscheidet, ob das Eingabeband weitergerückt werden soll und führt die Verrückung gegebenenfalls durch, ändert den Speicherinhalt und gibt ein Ausgabewort aus.

Definition 1. $P=[Z, X, Y, K, \mu, z_0]$ heißt deterministischer pushdown-Automat (PDA) oder Kellerautomat $=_{Df}$

1. X ist eine nichtleere endliche Menge (das Eingabealphabet). $\varnothing$, $\# \notin X$. $\varnothing$ ist ein Symbol zur Bezeichnung leerer Zellen auf den betrachteten Bändern.

2. Y ist eine nichtleere endliche Menge (das Ausgabealphabet).

3. Z ist eine endliche Menge (die Menge der Zustände), und $z_0 \in Z$ heißt Anfangszustand von P.

4. K ist eine endliche Menge (das Kelleralphabet). K enthält ein ausgezeichnetes Symbol Δ.

5. μ ist eine eindeutige Abbildung von $Z \times K \times (X \cup \{\#, \varnothing\})$ in $Z \times K^* \times Y^* \times \times \{0, 1\}$. (Für eine beliebige Menge M bezeichnen wir mit M^* die von M erzeugte freie Worthalbgruppe, deren Einselement (leeres Wort) immer e heißt.) μ hat die folgenden Eigenschaften: Ist $\mu(z, k, x)=(z', p, q, \varepsilon)$, so ist $p=\Delta p'$ mit $p' \in (K \setminus \{\Delta\})^*$, falls $k=\Delta$, und es ist $p \in (K \setminus \{\Delta\})^*$, falls $k \neq \Delta$. (Δ ist also ein Markierungszeichen, das den Anfang auf dem Kellerband angibt, nie verändert wird und nie außerhalb des Speicherwortanfangs vorkommen soll.)

Die Arbeitsweise des Automaten wird wie üblich durch Konfigurationen beschrieben.

Definition 2

(a) $[u, v, w]$ heißt Konfiguration von $P=_{Df}$

$$1)\ u \in Z\{\#, e\}X^*,$$

$$2)\ v \in \Delta(K \setminus \{\Delta\})^*,$$

$$3)\ w \in Y^*.$$

(b) $[u, v, w]$ heißt Anfangskonfiguration von $P=_{Df}$ $u \in z_0 X^* \wedge v=\Delta \wedge w=e$.
(c) $[u', v', w']$ heißt unmittelbare Folgekonfiguration von $[u, v, w]$ bezüglich P

$$([u, v, w] \vdash_P [u', v', w'])=_{Df} \quad \text{Falls } u=zx_1 \ldots x_n,$$

wobei

$$x_1 \in \{\#\} \cup X, \quad x_2, \ldots, x_n \in X, \quad n \geqq 0,$$

(ist $n=0$, so setzen wir $u=z$) und $v=\Delta k_1 \dots k_m$, wobei $k_1, \dots, k_m \in K \setminus \{\Delta\}$, $m \geqq 0$, (ist $m=0$, so setzen wir $v=\Delta$ und $k_m=\Delta$) und $\mu(z, k_m, x_1)=(z', p, q, \varepsilon)$, so ist

$$u' = \begin{cases} z' x_2 \dots x_n, & \text{falls } \varepsilon=1 \wedge n>1 \\ z', & \text{falls } (\varepsilon=1 \wedge n=1) \vee n=0 \\ z' \# x_2 \dots x_n, & \text{falls } \varepsilon=0 \wedge n \geqq 1, \end{cases}$$

$v'=\Delta k_1 \dots k_{m-1} p$ (p kann leer sein), $w'=wq$ (q kann leer sein).

(d) $[u', v', w']$ heißt Folgekonfiguration von $[u, v, w]$ bezüglich $P([u, v, w] \models_P [u', v', w'])=_{Df}$ Es existieren Konfigurationen $[u_1, v_1, w_1], \dots, [u_n, v_n, w_n]$ mit

$$[u_1, v_1, w_1]=[u, v, w] \wedge [u_n, v_n, w_n]=[u', v', w'] \wedge$$

$$\wedge [u_1, v_1, w_1] \vdash_P [u_2, v_2, w_2] \vdash_P \dots \vdash_P [u_n, v_n, w_n].$$

Was die Verwendung von PDA als Berechnungsalgorithmen von Wortfunktionen, d. h. von eindeutigen Abbildungen von X^* in Y^*, betrifft, so bieten sich zwei Möglichkeiten an. Erstens kann in Anlehnung an [3] die folgende Berechnungskonzeption gewählt werden.

Definition 3. Die Wortfunktion $f: X^* \to Y^*$ heißt PDA-berechenbar im ersten Sinne $=_{Df}$ Es existiert ein PDA $P=[Z, X, Y, K, \mu, z_0]$ mit

$$\forall w \exists z \exists p (w \in X^* \to z \in Z \wedge p \in K^* \wedge [z_0 w, \Delta, e] \models_P [z, p, f(w)])$$

$$\wedge \sim \exists z' \exists p' \exists v ([z_0 w, \Delta, e] \models_P [z', p', v] \vdash_P [z, p, f(w)]).$$

Hierbei kann der Fall eintreten, daß ein gewisser Anteil an Informationen über w in p und nicht in $f(w)$ eingeht. Daher liegt die Betrachtung einer zweiten Art der Berechnung nahe, die erst dann als beendet angesehen wird, wenn das Wort w gelesen und der Keller vollständig geleert ist.

Definition 4. Die Wortfunktion $f: X^* \to Y^*$ heißt PDA-berechenbar im 2 Sinne $=_{Df}$ Es existiert ein PDA $P=[Z, X, Y, K, \mu, z_0]$ mit

$$\forall w \exists z (w \in X^* \to z \in Z \wedge [z_0 w, \Delta, e] \models_P [z, \Delta, f(w)]).$$

Beispiel. Es sei P ein PDA mit $Z=\{z_0\}$, $X=Y=K$ und

$$\mu(z_0, k, x)=(z_0, kx, e, 1)$$

$$\mu(z_0, k, \varphi)=(z_0, e, k, 1), \text{ falls } k \neq \Delta.$$

Die von P im ersten Sinne berechnete Funktion ist die Konstante e, während die im zweiten Sinne berechnete Funktion f_2 die Wortinversion

$$f_2(x_1 \dots x_n)=x_n \dots x_1$$

ist, die wir in Zukunft auch mit $(x_1 \dots x_n)^{-1}$ bezeichnen wollen.

Die Berechnungen im ersten Sinne werden offenbar nur sequentielle Funktionen realisiert, während die Berechnungen im zweiten Sinne über die Klasse der sequentiellen Funktionen hinausführen, was schon durch das vorangehende Beispiel deutlich wird. Dieses Ergebnis steht dem bekannten Sachverhalt gegenüber ([2], Kapitel

2.5), daß die beiden Entscheidungsbegriffe für PDA, die unseren Definitionen 3 und 4 entsprechen, völlig äquivalent sind.

Für die beabsichtigte Analyse des Berechnungsverhaltens von PDA ist es bequem, nur solche PDA zu betrachten, die in jedem Takt den letzten gespeicherten Buchstaben entweder löschen oder ihn als Anfangsbuchstaben des zu speichernden Teilwortes übernehmen. Solche PDA nennen wir *einfach*.

Definition 5. Der PDA $P=[Z, X, Y, K, \mu, z_0]$ heißt einfach $=_{Df}$ $\mu(z, k, x)=$ $=(z', p, q, \varepsilon)\wedge p\neq e\rightarrow k\sqsubset p$. (Hierbei bezeichnet $\sqsubset$ die übliche Anfangswortrelation.)

In den §§ 2, 4 und 5 dieser Arbeit wollen wir grundsätzlich nur noch einfache PDA betrachten, ohne das jedesmal ausdrücklich zu erwähnen. Daß hierin keine wesentliche Beschränkung der Allgemeinheit liegt, zeigen die beiden folgenden Lemmata.

Lemma 1. Zu jedem PDA $P=[Z, X, Y, K, \mu, z_0]$ gibt es einen einfachen PDA $P_1=[Z_1, X, Y, K, \mu_1, z_0^{(1)}]$, der im ersten Sinne die gleiche Wortfunktion berechnet wie P.

Beweis. Wir setzen $Z_1=_{Df}Z\times K$, $z_0^{(1)}=[z_0,\Delta]$ und für jedes $k'\in K$

$$\mu_1([z, k], k', x)=$$

$$=\begin{cases}([z', k''], k', p, q, \varepsilon), & \text{falls } \mu(z, k, x)=(z', pk'', q, \varepsilon)\wedge k''\in K\setminus\{\Delta\}\\ ([z', k'], e, q, \varepsilon), & \text{falls } \mu(z, k, x)=(z', e, q, \varepsilon)\wedge k'\neq\Delta\\ ([z', \Delta], \Delta, q, \varepsilon), & \text{falls } \mu(z, k, x)=(z', e, q, \varepsilon)\wedge k'=\Delta.\end{cases}$$

Wie leicht zu sehen ist, berechnet P_1 im ersten Sinne die gleiche Wortfunktion wie P.

Lemma 2. Berechnet der PDA P im zweiten Sinne die auf ganz X^* definierte Funktion f, so berechnet der PDA P_1, der dem P durch die Konstruktion im Beweis von Lemma 1 zugeordnet werden kann, eine auf ganz X^* definierte Funktion f_1, die mit f in folgender Beziehung steht. Es gibt eine Zerlegung von X^* in endlich viele kontextfreie Mengen $C_1, \ldots, C_s$, und es gibt Wörter $r_1, \ldots, r_s\in Y^*$, so daß

$$f(w)=f_1(w)r_\sigma, \text{ falls } w\in C_\sigma.$$

Beweis. Es sei

$$[z_0w, \Delta, e]\models_P[z_1, p_1k_1, q_1]\vdash_P[z_2, p_2k_2, q_2]\vdash_P\ldots\vdash_P[z_n, \Delta, f(w)].$$

Hierbei sei $[z_1, p_1k_1, q_1]$ die erste Konfiguration der durch $[z_0w, \Delta, e]$ bestimmten Folge, deren erste Komponente nur aus einem Element aus Z besteht. Ferner gelte $k_1, k_2, \ldots, k_{n-1}\in K$. Dieser Konfigurationenfolge entspricht für P_1 die Folge

$$[[z_0, \Delta]w, \Delta, e]\models_{P_1}[[z_1, k_1], p_1, q_1]\vdash_{P_1}[[z_2, k_2], p_2, q_2]\vdash_{P_1}\ldots\vdash_{P_1}[[z_n, \Delta], \Delta, f(w)].$$

Wenn der Fall eintritt, daß für irgendein $v<n$ erstmalig $p_v=\Delta$ ist, so hat die diesem v entsprechende Konfiguration der zweiten Folge die Form $[[z_v, k_v], \Delta, q_v,]$ womit laut Definition 4 die Berechnung von $f_1(w)$ beendet ist. Das Wort r_v, das

$f_1(w)$ noch hinzugefügt werden muß, damit $f(w)$ entsteht, hängt offensichtlich nur von z_ν und k_ν ab. Daher gilt für alle w, die in der Menge

$$C_\nu =_{Df} \{w\colon \exists q([[z_0, \Delta]w, \Delta, e] \models_{P_1} [[z_\nu, k_\nu], \Delta, q])\}$$

liegen, die Beziehung

$$f(w) = f_1(w) r_\nu.$$

Wenn kzM die Kardinalzahl der Menge M bedeutet, gibt es höchstens $kzX \cdot kzK$ verschiedene derartige C_ν, die nach [3] alle kontextfrei sind. Da wegen der Voraussetzung, daß f auf ganz X^* definiert sein soll, jedes $w \in X^*$ in genau einer dieser Mengen C_ν liegen muß, ist das Lemma bewiesen.

§ 2 Funktionen, die im ersten Sinne von PDA berechnet werden

Es kann vorkommen, daß eine von einem PDA im ersten Sinne berechnete Wortfunktion nicht für alle $w \in X^*$ definiert ist. Das ist genau dann der Fall, wenn der PDA während der Abarbeitung von w in eine Situation kommt, in der er # liest und von der aus er nie wieder einen Befehl mit $\varepsilon = 1$ erreicht. Wir wollen hier nur auf ganz X^* definierte Funktionen untersuchen und beweisen dazu

Lemma 3. Zu jedem PDA P, der nach jeder Situation $(z, k, \#)$ wieder einen Befehl mit $\varepsilon = 1$ erreicht, gibt es einen PDA P', der im ersten und zweiten Sinne die gleiche Funktion berechnet wie P und der folgende Eigenschaft hat: Ist

$$\mu'(z, k, \#) = (z', p, q, \varepsilon), \tag{1}$$

so ist $p = e \wedge \varepsilon = 0$ oder $p = k \wedge \varepsilon = 1$.

Beweis. Es sei

$$\mu(z_1, k_1, x) = (z, k_1 p_1 k, q_1, 0), \tag{2}$$

so daß sich P im nächsten Takt in der Situation $(z, k, \#)$ befindet. Wir geben ein System von Reduktionsregeln an, mit denen P' aus P konstruiert werden kann.

1. *Fall.* Es gilt

$$\mu(z, k, \#) = (z', p, q, 0) \tag{3}$$

mit $p \neq e$.

Dann ersetzen wir für jedes Quintupel (z_1, k_1, x, p_1, q_1), das (2) erfüllt, die Befehle (2) und (3) durch

$$\mu(z_1, k_1, x) = (z', k_1 p_1 p, q_1 q, 0).$$

2. *Fall.* Es gilt

$$\mu(z, k, \#) = (z', p, q, 1) \tag{4}$$

mit $p = e$.

In diesem Falle wird ein neuer Zustand z'' eingeführt, und wir ersetzen (2) und (4) durch

$$\mu(z, k, \#) = (z'', e, q, 0) \quad \text{und}$$

$$\mu(z'', l, \#) = (z', l, e, 1) \quad \text{für alle } l \in K.$$

3. *Fall.* Es gilt

$$\mu(z, k, \#)=(z', kp', q, 1) \tag{5}$$

mit $p' \neq e$.

Dann ersetzen wir (2) und (5) durch

$$\mu(z_1, k_1, x)=(z', k_1 p_1 k p', q_1 q, 1)$$

für jedes (2) erfüllende Quintupel (z_1, k_1, x, p_1, q_1).

Nach Voraussetzung über den Definitionsbereich von f treten, ausgehend von $(z, k, \#)$ keine Zyklen auf. Daher können durch endlich viele Anwendungen der vorstehenden Ersetzungsregeln unter eventueller Vergrößerung der Zustandmenge alle Befehle eliminiert werden, die nicht der Bedingung (1) genügen. Den entstehenden PDA nennen wir P'. Es ist klar, daß bei Anwendung dieser Reduktionen das Berechnungsverhalten von P im ersten und zweiten Sinne nicht beeinflußt wird.

Der folgende Hilfssatz zeigt, daß die Arbeitsweise der PDA in gewisser Weise normiert werden kann. Für die geplante Analyse ist es angenehm, diese Normiertheit voraussetzen zu können.

Lemma 4. Zu jedem PDA $P=[Z, X, Y, K, \mu, z_0]$ existiert ein PDA $P'=$ $=[Z', X, Y, K, \mu', z_0]$, der im ersten und zweiten Sinne die gleiche Funktion berechnet wie P und folgende Bedingung erfüllt

$$\mu(z, k, x)=(z', p, q, \varepsilon) \wedge x \in X \rightarrow p \neq e.$$

Beweis. Man setzt $\mu'=\mu$ für alle Tripel (z, k, x), die der Bedingung des Lemmas genügen. Ist $\mu(z, k, x)=(z', e, q, 0)$ für $x \in X$, so führt man einen neuen Zustand z_1 ein und ersetzt diesen Befehl durch

$$\mu'(z, k, x)=(z_1, k, q, 0),$$

$$\mu'(z_1, k, \#)=(z', e, e, 0).$$

Ist $\mu(z, k, x)=(z', e, q, 1)$, so führt man zwei neue Zustände z_2 und z_3 ein und ersetzt diesen Befehl durch

$$\mu'(z, k, x)=(z_2, k, q, 0),$$

$$\mu'(z_2, k, \#)=(z_3, e, e, 0),$$

$$\mu'(z_3, k', \#)=(z', k', e, 1),$$

für alle $k' \in K$. Der Zwischenzustand z_3 mußte eingeführt werden, damit der neue Automat der Bedingung von Lemma 3 genügt. Z' unterscheidet sich von Z um die hierbei neu hinzugefügten Zustände. Daß P' bezüglich beider Berechnungsarten zu P äquivalent ist, ist trivial.

Von jetzt an betrachten wir grundsätzlich nur solche PDA, die der Bedingung (1) von Lemma 3 genügen und gemäß Lemma 4 normiert sind.

Gegeben sei der PDA $P=[Z, X, Y, K, \mu, z_0]$, und es sei o.B.d.A. $Y \cap K=\varnothing$. Um sein Berechnungsverhalten zu untersuchen, ordnen wir P zwei miteinander gekoppelte endliche partielle Automaten $Q_1=[U, X, Y \cup K, f_1, g]$ und $Q_2=$ $=[V, K, Y, f_2, h]$ zu. In diesen Quintupeln bedeuten die Komponenten der Reihe nach die Zustandsmenge, Eingabemenge, Ausgabemenge, Überführungsfunktion und Ausgabefunktion ([4]).

Das Automatenpaar Q_1, Q_2 soll die Arbeit von P beschreiben. Solange P einen Teil des Eingabewortes w abarbeitet (d.h. Befehle mit $\varepsilon=1$ anwendet), simuliert Q_1 die Tätigkeit von P. Sobald aber P die Abarbeitung von w zum Zwecke der Ausspeicherung unterbricht (d.h. Befehle mit $\varepsilon=0$ auftreten), erreicht Q_1 einen Zustand aus V, und der Ausspeicherungsprozeß wird nun von Q_2 simuliert, wobei als Eingabe das bis dahin von P gespeicherte invertierte Wort dient. Ist die Ausspeicherungsperiode vorüber, so wird ein Zustand aus U erreicht, und Q_1 tritt wieder in Aktion.

Im einzelnen sind Q_1 und Q_2 folgendermaßen definiert.

$$U = U' \cup U'' \text{ mit}$$

$$U' =_{Df} \{[z_0, \Delta]\} \cup \{[z, k] : \exists z' \exists k' \exists x \exists p \exists q \, (\mu(z', k', x) = (z, pk, q, 1))\},$$

$$U'' =_{Df} \{z : \exists z' \exists k \exists x \exists p \exists q \, ([z', k] \in U' \wedge \mu(z', k, x) = (z, kp, q, 0))\}.$$

Für $[z, k] \in U'$ sind die Überführungsfunktion f_1 und die Ausgabefunktion g definiert durch

$$f_1([z, k], x) =_{\mathrm{Df}} \begin{cases} [z', k'], & \text{falls } \mu(z, k, x) = (z', pk', q, 1) \\ z', & \text{falls } \mu(z, k, x) = (z', kp, q, 0) \end{cases}$$

$$g([z, k], x) =_{Df} pq, \quad \text{falls} \quad \mu(z, k, x) = (z', kp, q, \varepsilon).$$

Auf U'' sind f_1 und g nicht definiert.

$$V_h = V' \cup V'' \text{ mit}$$

$$V' =_{Df} \{z : \exists z' \exists k \exists q \, (\mu(z', k, \#) = (z, e, q, 0))\} \cup U'',$$

$$V'' =_{Df} \{[z, k] : \exists z' \exists q \, (z' \in V' \wedge \mu(z', k, \#) = (z, k, q, 1))\}.$$

Für $z \in V'$ sind Überführungs- und Ausgabefunktion von Q_2 folgendermaßen definiert:

$$f_2(z, k) =_{\mathrm{Df}} \begin{cases} z', & \text{falls } \mu(z, k, \#) = (z', e, q, 0) \\ [z', k], & \text{falls } \mu(z, k, \#) = (z', k, q, 1). \end{cases}$$

$h(z, k) =_{\mathrm{Df}} q$, falls $\mu(z, k, \#) = (z', p, q, \varepsilon)$. Auf V'' werden f_2 und h nicht definiert. Wir vereinbaren noch, die auf X^* bzw. K^* erweiterten Ausgabefunktionen von Q_1 bzw. Q_2 ([4]) ebenfalls mit g bzw. h zu bezeichnen. Ferner nehmen wir zur Erleichterung der folgenden Untersuchungen die Mengen U'' und V'' in der Form $V'' = \{u_1, \ldots, u_n\}$, $U'' = \{v_1, \ldots, v_m\}$ an und setzen $u_0 =_{\mathrm{Df}} [z_0, \Delta]$.

Mit g_i (bzw. h_j) bezeichnen wir die von Q_1 im Zustand u_i $(i=0, \ldots, n)$ bzw. von Q_2 im Zustand v_j $(j=1, \ldots, m)$ berechneten Funktionen. Es seien δ_Y und δ_K die durch

$$\delta_Y(w) = \begin{cases} w, & \text{falls } w \in Y \\ e, & \text{falls } w \in K \end{cases}$$

$$\delta_K(w) = \begin{cases} e, & \text{falls } w \in Y \\ w, & \text{falls } w \in K \end{cases}$$

definierten Homomorphismen von $(Y \cup K)^*$ auf Y^* bzw. K^*. Damit gewinnen wir die für das weitere wichtigen Funktionen

$$d_i =_{\mathrm{Df}} g_i \circ \delta_Y \quad \text{und} \quad s_i =_{\mathrm{Df}} g_i \circ \delta_K.$$

(Wir bezeichnen mit fog die Funktion, die durch $\mathrm{fog}\,(x) = g(f(x))$ aus f und g entsteht.) Ferner setzen wir

$$M_i^j =_{\mathrm{Df}} \{p : p \in X^* \wedge f_1(u_i, p) = v_j\} \quad \text{für } i = 0, \ldots, n;\ j = 1, \ldots, m,$$
$$M_i =_{\mathrm{Df}} \{p : p \in X^* \wedge \forall q (q \sqsupset p \to f_1(u_i, q) \notin U'')\} \quad \text{für } i = 0, \ldots, n,$$
$$N_j^i =_{\mathrm{Df}} \{w : w \in K^* \wedge f_2(v_j, w) = u_i\} \quad \text{für } i = 0, \ldots, n;\ j = 1, \ldots, m.$$

Wir beschaffen uns jetzt eine Funktion s, die das Verhalten des Speichers von P in dem Sinne beschreibt, daß $s(p)$ den Inhalt des Speicherbandes nach Abarbeitung von p (einschließlich sich eventuell anschließender Ausspeicherungsoperationen) angibt. Der expliziten Darstellung von s stellen wir eine anschauliche Erläuterung voran. Bei Abarbeitung von p wächst s zunächst so lange an, bis ein erstes $v_{j_1} \in U''$ erreicht wird. Dies möge für das Anfangswort p_1 von p der Fall sein. Nun setzt eine Ausspeicherungsphase ein, in deren Verlauf der bisherige Speicherinhalt $s_0(p_1)$ um ein eindeutig bestimmtes Endwort q_1 verringert wird. Das dabei übrigbleibende Anfangsstück von $s_0(p_1)$ ist $s(p_1)$. Die Eingabe von q_1^{-1} auf v_{j_1} führt in Q_2 zu einem eindeutig bestimmten $u_{i_1} \in V''$, mit dem der Prozeß fortgesetzt wird, wenn p noch nicht abgearbeitet ist.

Mit der Schreibweise

$$w \setminus v =_{\mathrm{D}} \begin{cases} u, & \text{falls } w = uv \\ \text{nicht definiert} & \text{sonst} \end{cases}$$

können wir s explizit angeben:

$$\left.\begin{array}{l} s(p) = (((\ldots((s_0(p_1) \setminus q_1) s_{i_1}(p_2)) \setminus q_2 \ldots) s_{i_{t-1}}(p_t)) \setminus q_t) s_{i_t}(p_{t+1}), \\ \text{falls } p = p_1 \ldots p_{t+1} \text{ mit} \\ \exists j_1 \exists q_1 \exists i_1 (p_1 \in M_0^{j_1} \wedge q_1^{-1} \in N_{j_1}^{i_1}) \wedge \exists j_2 \exists q_2 \exists i_2 (p_2 \in M_{i_1}^{j_2} \wedge q_2^{-1} \in N_{j_2}^{i_2}) \wedge \ldots \\ \ldots \wedge p_{t+1} \in M_{i_t} \wedge \text{ die } q_i \text{ sind Endwörter derjenigen Wörter, von} \\ \text{denen sie abgezogen werden.} \\ \text{Für } t = 0 \text{ wollen wir die Darstellung so verstehen: } s(p) = s_0(p) \text{ und } p \in M_0. \end{array}\right\} \quad (6)$$

Man beachte daß alle vorkommenden Größen p_ν, q_ν, i_ν und j_ν durch p und P vollkommen eindeutig festgelegt sind. Wir führen zwei abkürzende Sprechweisen ein.

Definition 6. $w \alpha N_j^i =_{\mathrm{Df}} \exists w_1 (w_1 \sqsubseteq w \wedge w_1 \in N_j^i)$. Weil N_j^i total ungeordnet ist bezüglich $\sqsubseteq$, gibt es für $w \alpha N_j^i$ genau ein solches w_1. Deshalb hat folgende Definition einen Sinn.

Definition 7. Ist $w \alpha N_j^i$, so setzen wir $h_j(w) = h_j(w_1)$ mit

$$w_1 \sqsubseteq w \wedge w_1 \in N_j^i.$$

Damit sind wir in der Lage, die von P im ersten Sinne berechnete Funktion f folgendermaßen zu beschreiben.

$$\left.\begin{array}{l} f(p) = d_0(p_1)h_{j_1}(s(p_1)^{-1})d_{i_1}(p_2)h_{j_2}(s(p_1p_2)^{-1})\dots \\ \dots d_{i_{k-1}}(p_k)h_{j_k}(s(p_1\dots p_k)^{-1})d_{i_k}(p_{k+1}),\ \text{falls}\ p=p_1\dots p_{k+1}\wedge \\ \exists j_1\exists i_1(p_1\in M_0^{j_1}\wedge s(p_1)^{-1}\alpha N_{j_1}^{i_1})\wedge\exists j_2\exists i_2(p_2\in M_{i_1}^{j_2}\wedge s(p_1p_2)^{-1}\alpha N_{j_2}^{i_2})\wedge\dots\wedge p_{k+1}\in M_{i_k}. \\ \text{Für } k=0 \text{ wollen wir diese Darstellung so verstehen: } f(p)=d_0(p) \text{ und } p\in M_0. \end{array}\right\} (7)$$

Wir definieren nun automatenunabhängig eine Klasse von sequentiellen Wortfunktionen von X^* in Y^*, die sich genau als die Klasse derjenigen (auf ganz X^* definierten) Wortfunktionen erweisen wird, die von PDA im ersten Sinne berechnet werden.

Definition 8. $f\in P_1(X, Y)=_{\mathrm{Df}}$

1. Es gibt eine endliche Menge K, und es existieren reguläre Mengen M_i, $M_i^j\subseteq X^*$ $(i=0, \dots, n; j=1, \dots, m)$ und reguläre Mengen $N_j^i\subseteq K^*$ $(i=1, \dots, n;\ j=1, \dots, m)$ mit den Eigenschaften

(a) $\forall i\forall j\,(M_i\cap M_i^j=\varnothing)$

(b) $\forall i\forall j\forall j'\,(j\neq j'\to M_i^j\cap M_i^{j'}=\varnothing)$

(c) $\forall i\forall p\,(p\notin M_i\leftrightarrow\exists j\exists q(q\sqsubseteq p\wedge q\in M_i^j))$

(d) $\forall i\forall j$ ($\bigcup\limits_{\nu} M_i^{\nu}$ und $\bigcup\limits_{\mu} N_j^{\mu}$ sind bezüglich $\sqsubseteq$ total ungeordnet)

(e) $\forall j\forall i\forall i'\,(i\neq i'\to N_j^i\cap N_j^{i'}=\varnothing)$

(f) $\exists\Delta(\Delta\in K\wedge\forall i\forall w\forall j(w\Delta\alpha N_i^j))$.

2. Es gibt sequentielle Funktionen endlichen Gewichts

$$s_0, \dots, s_n : X^*\to K^*,$$

$$d_0, \dots, d_n : X^*\to Y^*,$$

$$h_1, \dots, h_m : K^*\to Y^*,$$

so daß f durch (7) festgelegt ist, wobei die dazu benötigte Funktion s sich nach (6) ergibt. Wir wollen kurz sagen, f sei durch $\{M_i, M_i^j, N_j^i, s_i, h_j\}_{n,m}$ bestimmt.

Es gilt der

Satz 1. Eine auf ganz X^* definierte Wortfunktion mit Werten in Y^* ist genau dann PDA-berechenbar im ersten Sinne, wenn sie zu $P_1(X, Y)$ gehört.

Beweis

1. Daß eine von einem PDA berechnete überall auf X^* definierte Funktion zu $P_1(X, Y)$ gehört, (wobei Y das Ausgabealphabet das berechnenden PDA ist), geht aus der bisherigen Analyse hervor.

2. Es sei $f\in P_1(X, Y)$.

Wir geben einen PDA P an, der f im ersten Sinne berechnet. Durch elementare automatentheoretische Konstruktionen kann man zunächst zwei endliche partielle Mealy-Automaten $Q_1=[U, X, Y, K, f_1, g]$ und $Q_2=[V, K, Y, f_2, h]$ (mit $Y\cap K=\varnothing$)

konstruieren, die folgende Eigenschaft haben. Es gibt eine Teilmenge $U''=\{v_1, \ldots, v_m\}\subseteq U\cap V$ und eine Teilmenge $V''=\{u_1, \ldots, u_n\}\subseteq U\cap V$ und ein $u_0\in U$, so daß gilt: Versieht man Q_1 mit dem Anfangszustand u_i $(i=0, \ldots, n)$, so akzeptiert dieser initiale Automat durch v_j $(j=1, \ldots, m)$ die Menge M_i^j, und er berechnet eine Funktion g_i, die aus den Funktionen d_i und s_i durch

$$g_i(x_1, \ldots, x_n)=_{\mathrm{Df}}\sigma_1\omega_1\sigma_2\omega_2\ldots\sigma_n\omega_n, \quad \text{falls} \quad d_i(x_1\ldots x_\nu)=d_i(x_1\ldots x_{\nu-1})\omega_\nu$$

und

$$s_i(x_1\ldots x_\nu)=s_i(x_1\ldots x_{\nu-1})\sigma_\nu$$

hervorgeht. Versieht man Q_2 mit dem Anfangszustand v_j $(j=1, \ldots, m)$, so akzeptiert dieser initiale Automat durch den Finalzustand u_i $(i=1, \ldots, n)$ die Menge N_j^i, und er berechnet die Funktion h_j.

Ausgehend von diesen beiden gekoppelten Automaten geben wir jetzt einen PDA $P=[Z, X, Y, K, \mu, u_0]$ an:

$$Z=_{\mathrm{Df}}U\cup V.$$

μ wird folgendermaßen definiert:

a) $v\in V\setminus V''$

Gilt $f_2(v, k)=v'$ und $h(v, k)=q$, so setzen wir

$$\mu(v, k, \#)=\begin{cases}(v', e, q, 0), & \text{falls } v'\notin V''\\ (v', k, q, 1), & \text{falls } v'\in V''.\end{cases}$$

b) $u\in U\setminus U''$

Gilt $f_1(u, x)=u'$ und $g(u, x)=pq$ mit $p\in K^*\wedge q\in Y^*$, so setzen wir für alle $k\in K$

$$\mu(u, k, x)=\begin{cases}(u', k, q, 0), & \text{falls } u'\in U''\\ (u', kp, q, 1), & \text{falls } u'\notin U''.\end{cases}$$

Der so entstandene PDA berechnet ersichtlich genau die gegebene Funktion f im ersten Sinne. Damit ist Satz 1 bewiesen.

Wir bemerken noch, daß $P_1(X, Y)$ sequentielle Funktionen unendlichen Gewichts enthält. Dazu geben wir folgendes Beispiel an:

$$f(a^{n_1}b^{m_1}\ldots a^{n_k}b^{m_k})=\begin{cases}ba^{n_1}b^{m_1}\ldots a^{n_k}b^{m_k-1}, & \text{falls } m_k\geqq 1\\ ba^{n_1}b^{m_1}\ldots a^{n_{k-1}}b^{m_{k-1}-1}, & \text{falls } m_k=0.\end{cases}$$

Diese Funktion wird durch den PDA $P=[\{z_0, z_1\}, \{a, b\}, \{a, b\}, \{\Delta, a'\}, \mu, z_0]$ berechnet, dessen μ durch die Tabelle

Z	K	X	Z	K	Y	ε
z_0	Δ	a	z_0	$\Delta a'$	e	1
z_0	Δ	b	z_0	Δ	b	1
z_0	a'	a	z_0	$a'a'$	e	1
z_0	a'	b	z_1	a'	b	0
z_1	a'	#	z_1	e	a	0
z_1	Δ	#	z_0	Δ	e	1

gegeben ist. f ist demnach eine Funktion aus $P_1(X, X)$ mit $X=\{a, b\}$. Wegen $f(a^n)=e$ und $f(a^nb)=ba^n$ folgt für den durch a^n bestimmten Zustand f_an von f die Beziehung $f_an(b)=ba^n$. Demnach gilt $f_an\neq f_am$ für $n\neq m$, womit gezeigt ist, daß f unendliches Gewicht hat.

§ 3 Quasisequentielle Funktionen

Für die Analyse des Berechnungsverhaltens von PDA im zweiten Sinne, die über die sequentiellen Funktionen hinausführt, stellen wir hier eine geeignete Funktionenklasse bereit, die durch Verallgemeinerung eines Ansatzes aus [6] gewonnen wird.

X sei ein beliebiges endliches Alphabet. Mit $|w|$ bezeichnen wir die Länge des Wortes $w\in X^*$.

Definition 9

(a) Es sei $\mathfrak{T}$ die Menge aller allgemein rekursiven Funktionen t von X^* in Nz mit der Eigenschaft

$$\forall w(w\in X^*\to 1\leqq t(w)\leqq|w|).$$

(b) Es sei $t\in\mathfrak{T}$. Wir setzen für beliebige Wörter p mit $|p|=|w|$ über beliebigem Alphabet

$$\pi_{t(w)}(p)=\xi_1\ldots\xi_{t(w)-1}\xi_{t(w)+1}\ldots\xi_n,\quad\text{falls}\quad p=\xi_1\ldots\xi_n.$$

(Die Definition ist so zu verstehen, daß im Falle $t(w)=1$ oder $=n$ der erste bzw. letzte Buchstabe von p gestrichen wird.)

Definition 10

(a) Eine längentreue Funktion f von X^* in Y^* (d.h. eine solche, für die gilt $\forall w(w\in X^*\to|f(w)|=|w|)$ heißt $\langle t, t'\rangle$-sequentiell $=_{Df}$

$$\forall w\big(w\in X^*\to f(\pi_{t(w)}(w))=\pi_{t'(w)}(f(w))\big).$$

(b) f heißt quasisequentiell $=_{Df}$ Es gibt $t, t'\in\mathfrak{T}$, so daß f $\langle t, t'\rangle$-sequentiell ist.

(c) $S_{tt'}$ bezeichne die Menge aller $\langle t, t'\rangle$-sequentiellen Funktionen (von X^* in Y^*).

Die Verallgemeinerung gegenüber [6] besteht im Verzicht auf die zusätzliche Eigenschaft an die $t\in\mathfrak{T}$, daß $(t(w)$ nur von der Länge von w abhängt. Obwohl durch Definition 10 eine größere Funktionenklasse festgelegt wird als durch die entsprechende Definition in [6], wollen wir die dort eingeführte Bezeichnung „quasisequentielle Funktionen" beibehalten und künftig im Sinne von Definition 10 verstehen.

Die sequentiellen Funktionen ([5], [4]) ergeben sich genau als $\langle l, l\rangle$-sequentielle Funktionen, wobei l die durch $l(w)=_{Df}|w|$ definierte Funktion ist. Künftig wird l immer in dieser Bedeutung gebraucht.

Für unsere Zwecke sind folgende Sätze aus [6] wichtig, deren Beweise unabhängig von der hier vorgenommenen Verallgemeinerung von dort übernommen werden können.

Satz 2. Für $t\neq l$ ist S_{lt} von der Klasse der sequentiellen Funktionen verschieden. Die einzigen sequentiellen Funktionen in S_{lt} sind die Homomorphismen.

Unter einer Wortpermutation verstehen wir eine Funktion s mit der Eigenschaft $s(x_1 \ldots x_n) = x_{\sigma_w(1)} \ldots x_{\sigma_w(n)}$ für jedes $w = x_1 \ldots x_n \in X^*$, wobei σ_w eine Permutation der Indexmenge $\{1, \ldots, |w|\}$ ist.

Satz 3. In jedem S_{lt} gibt es genau eine Wortpermutation s_{lt}. s_{lt} ist die identische Abbildung genau dann, wenn $l = t$ ist.

Zur Illustration geben wir für s_{lt} die Folge der erzeugenden Indexpermutationen an:

$$\sigma_x(1) =_{\mathrm{Df}} 1$$

$$\sigma_{wx}(i) =_{\mathrm{Df}} \begin{cases} \sigma_w(i) & \text{für } i < t(wx) \\ \sigma_w(i-1) & i > t(wx) \\ n+1 & i = t(wx). \end{cases}$$

Definition 10′

(a) Ist $\{\sigma_w : w \in X^*\}$ die Schar der erzeugenden Indexpermutationen für s_{lt}, so setzen wir für beliebige Wörter $p = y_1 \ldots y_n$ mit $n = |w|$ über beliebigem Alphabet Y

$$s_{lt}^w(p) =_{\mathrm{Df}} y_{\sigma_w(1)} \cdots y_{\sigma_w(n)}.$$

(b)
$$(\varphi * s_{lt})(w) =_{\mathrm{Df}} s_{lt}^w(\varphi(w)).$$

Satz 4. $f \in S_{lt}$ genau dann, wenn es eine sequentielle Funktion φ gibt mit

$$f = \varphi * s_{lt},$$

wobei s_{lt} die in S_{lt} vorhandene Wortpermutation ist.

Auch nichtlängentreue quasisequentielle Funktionen werden benötigt. Wir wollen sie in Anlehnung an Satz 4 erklären. Um eine geeignete Wortpermutation zu erklären, die an die Stelle von s_{lt} aus Satz 4 tritt, führen wir eine Funktion r ein, die jedem $w \in X^*$ eine natürliche Zahl $\zeta(w)$ und eine Permutation ϱ_w der Zahlen $1, \ldots, \zeta(w)$ zuordnet. Dabei ist $\zeta(wx)$ für die zu definierenden Funktionen der Längenzuwachs des Bildwortes von wx gegenüber dem Bildwort von w. Daher hat das Bildwort von w, wenn (was wir immer fordern wollen) das Bildwort des leeren Wortes wieder das leere Wort ist, die Länge $\lambda_r(w) = \sum_{w_1 \sqsubseteq w} \zeta(w_1)$. Wir definieren die Wortpermutation $s_{l[t,r]}$ durch Angabe der Folge $\{\sigma_w : w \in X^*\}$ der erzeugenden Indexpermutationen:

$$\sigma_x(1) = 1$$

$$\sigma_{wx}(i) = \begin{cases} \sigma_w(i) & \text{für } 1 \leqq i < t(wx) \\ \sigma_w(i - \zeta(wx)) & \text{für } t(wx) + \zeta(wx) \leqq i \leqq \lambda(wx) \\ \lambda_r(w) + \varrho_w(i - t(wx) + 1) & \text{für } t(wx) \leqq i < t(wx) + \zeta(wx). \end{cases}$$

Damit erklären wir nichtlängentreue quasisequentielle Funktionen.

Definition 11. Die Wortfunktion $f : X^* \to Y^*$ heißt $\langle l, [t, r]\rangle$-sequentiell $=_{\mathrm{Df}}$ Es existiert eine sequentielle Funktion $\varphi : X^* \to Y^*$ mit $|\varphi(w)| = \lambda_r(w)$ und

$$f = \varphi * s_{l[t,r]}.$$

Für $\zeta(w) \equiv 1$ ergeben sich als Spezialfall die längentreuen quasisequentiellen Funktionen.

Die Verallgemeinerung, daß der Zuwachs von $f(wx)$ gegenüber $f(w)$ nicht in Form eines einzigen Teilwortes an einer Stelle in $f(w)$ eingeschoben wird, sondern sich auf mehrere Stellen verteilt, wird hier nicht gebraucht.

§ 4 Funktionen, die im zweiten Sinne von treu ausspeichernden PDA berechnet werden

Die sogenannten treu ausspeichernden PDA verdienen deswegen besondere Beachtung, weil sie quasisequentielle Funktionen berechnen.

Definition 12. Der PDA $[Z, X, Y, K, \mu, z_0]$ heißt treu ausspeichernd $=_{\mathrm{Df}}$ Es existiert eine eindeutige Abbildung $\alpha: K \to Y^*$ mit

$$\forall y \forall z \forall z' \forall k \forall \varepsilon \left(\mu(z, k, \#) = (z', e, y, \varepsilon) \to y = \alpha(k)\right)$$

$$\forall y \forall z \forall z' \forall k \left(\mu(z, k, \varrho) = (z', e, y, 0) \to y = \alpha(k)\right).$$

Wir behandeln zu Beginn den Spezialfall der synchronen PDA.

Definition 13. Der PDA $P = [Z, X, Y, K, \mu, z_0]$ heißt synchron $=_{\mathrm{Df}}$ Ist $\mu(z, k, x) = (z', p, q, \varepsilon)$, so ist

(a) falls $x \in X$ ist, $p = k \wedge q \in Y$ (P druckt) oder $p = kk' \wedge q = e$ mit $k' \in K \setminus \{\Delta\}$ (P speichert),

(b) falls $x = \# \wedge k \neq \Delta$ ist, $p = e \wedge q \in Y$ (P speichert aus),

(c) falls $x = \# \wedge k = \Delta$ ist, $p = \Delta \wedge q = e \wedge \varepsilon = 1$,

(d) falls $x = \varrho \wedge k \neq \Delta$ ist, $p = e \wedge q \in Y \wedge \varepsilon = 1$.

Eine im zweiten Sinne von einem synchronen PDA berechnete Funktion ist offenbar längentreu.

Die Analyse des Berechnungsverhaltens im zweiten Sinne eines PDA $P = [Z, X, Y, K, \mu, z_0]$ stützt sich auf eine sequentielle Funktion τ_P, die die Arbeitsweise von P beschreibt. Ausgehend von P führen wir die gleichen Überlegungen durch wie im § 2 und verfügen damit über die Mengen M_i, M_i^j, $N_j^{i'}$ und die Funktionen g_i, d_i, s_i und h_i ($i = 0, \ldots, n$; $j = 1, \ldots, m$; $i' = 1, \ldots, n$). Wenn γ der durch

$$\gamma(u) =_{\mathrm{Df}} \begin{cases} D & \text{für } u \in Y, \\ S & \text{für } u \in K, \end{cases}$$

festgelegte Homomorphismus von $(Y \cup K)^*$ in $\{D, S\}^*$ ist, während η der Homomorphismus von Y^* in $\{A\}^*$ ist, der durch $\eta(y) = A$ für jedes $y \in Y$ bestimmt ist, so setzen wir

$$\gamma_i =_{\mathrm{Df}} g_i \circ \gamma, \quad \eta_j =_{\mathrm{Df}} h_j \circ \eta$$

und definieren τ_P durch $\{M_i, M_i^j, N_j^i, s_i, \gamma_i, \eta_j\}_{n,m}$ gemäß Definition 8.

Die so definierte Funktion τ_P beschreibt insofern die Arbeitsweise von P, als $\tau_P(w)$ genau die Folge der vorzunehmenden Operationen (D = Drucken, S = Speichern, A = Ausspeichern) angibt, die bei Abarbeitung von w im Sinne der Definition 3 auszuführen sind.

Definition 14. θ sei der Homomorphismus von $\{A, D, S\}^*$ in die Halbgruppe $[Nz, +]$ der natürlichen Zahlen, der durch $\theta(A)=\theta(D)=1$, $\theta(S)=0$ definiert wird. w' entstehe aus dem Wort w durch Streichen des letzten Buchstaben. Dann setzen wir $t_P(w) =_{Df} 1+\theta(\tau_P(w')) = 1+$Anzahl der in $\tau_P(w')$ vorkommenden A und D.

Die Bedeutung dieses t_P wird deutlich durch den

Satz 5a. Wird die Wortfunktion f durch einen synchronen treu ausspeichernden PDA P im zweiten Sinne berechnet, so ist $f \in S_{lt_P}$ (vgl. Def. 10c, wobei wie oben $l(w)=|w|$ gesetzt ist.)

Beweis. Nach Definition 10 ist für jedes $w \in X^*$ zu zeigen

$$\pi_{t_P(w)}(f(w))=f(\pi_{l(w)}(w)).$$

Es sei $f(x_1 \ldots x_n)=y_1 \ldots y_n$. Wir betrachten den Zeitpunkt, in dem die Abarbeitung von $x_1 \ldots x_n$ im ersten Sinne gerade beendet ist. Bis zu diesem Moment seien bereits die Buchstaben $y_1 \ldots y_k$ ausgegeben worden. Hieraus folgt:

1. Im Speicher von P befindet sich noch ein Wort der Länge $n-k$, das in den folgenden Takten ausgegeben wird, wobei das Restwort $y_{k+1} \ldots y_n$ entsteht.

2. Nach Definition von t_P ist $t_P(x_1 \ldots x_n x)= k+1$ für jedes $x \in X$.

Es sei $w=x_1 \ldots x_n x_{n+1}$. Wird $f(x_1, \ldots, x_{n+1})$ berechnet und ist $\tau_P(w)= =\tau_P(x_1 \ldots x_n)\sigma_{n+1}$, so sind zwei Fälle zu unterscheiden.

1. *Fall.* $\sigma_{n+1}=SA^c$ $(c \geqq 0)$.

Dann wird x_{n+1} als irgendein k_{n+1} gespeichert und anschließend werden c Buchstaben des Speichers, als erster der zuletzt gespeicherte Buchstabe k_{n+1}, ausgespeichert. Damit ist die Abarbeitung von w im ersten Sinne beendet. Der Rest von $f(w)$ entsteht durch nachfolgende Speicherleerung. Es ergibt sich

$$f(w)=y_1 \ldots y_k y_{n+1} y_{k+1} \ldots y_n \quad \text{mit} \quad y_{n+1}=\alpha(k_{n+1}).$$

2. *Fall.* $\sigma_{n+1}=DA^c$ $(c \geqq 0)$.

Dann wird x_{n+1} als ein y'_{n+1} gedruckt. Alles weitere verläuft wie im ersten Fall. Es entsteht

$$f(w)=y_1 \ldots y_k y'_{n+1} y_{k+1} \ldots y_n.$$

In beiden Fällen gilt

$$\pi_{t_P(w)}(f(w))=y_1 \ldots y_k y_{k+1} \ldots y_n=f(x_1 \ldots x_n)=f(\pi_{l(w)}(w)),$$

was wir beweisen wollten.

Nach Satz 4 gibt es für diese Funktion f eine Darstellung der Form $\varphi * s_{lt_P}$ mit sequentiellem φ, über das der nächste Satz eine Aussage macht.

Satz 5b. Die eben erwähnte Funktion φ ist gemäß Definition 8 durch $\{M_i, M_i^j, N_j^i, s_i, \bar{g}_i, \bar{h}_j\}_{n,m}$ gegeben, wobei alle $\bar{h}_j$ die konstante Funktion e bedeuten, die $\bar{g}_i$ aus g_i (vgl. § 2) dadurch entstehen, daß alle $k \in K$ durch $\alpha(k)$ ersetzt werden (vgl. Def. 12), während alle anderen Mengen bzw. Funktionen genau diejenigen sind, die sich bei der Analyse von P in § 2 ergeben haben.

Beweis. Durch Induktion über die Wortlänge von w zeigen wir

$$f(w)=s^{w}_{lt_p}(\varphi(w)).$$

Der Induktionsbeginn ist klar.

Induktionsannahme. Es gilt $f(w)=s^{w}_{lt_p}(\varphi(w))$ für $w=x_1\dots x_n$.
Induktionsschluß. Sei

$$f(x_1\dots x_n)=y_1\dots y_n,$$

$$\varphi(x_1\dots x_n)=y_{i_1}\dots y_{i_n}$$

und

$$\varphi(x_1\dots x_nx_{n+1})=y_{i_1}\dots y_{i_n}y_{i_{n+1}}.$$

Nach Definition von φ ist $y_{i_{n+1}}$ entweder $\alpha(k_{n+1})$, wobei k_{n+1} der beim Lesen von x_{n+1} gespeicherte Buchstabe ist, oder der beim Lesen von x_{n+1} gedruckte Buchstabe. Aus dem Beweis von Satz 5a wissen wir, daß $y_{i_{n+1}}$ in das Wort $f(x_1\dots x_n)$ an der Stelle $t_P(x_1\dots x_{n+1})$ eingefügt wird

$$f(x_1\dots x_{n+1})=y_1\dots y_k y_{i_{n+1}}y_{k+1}\dots y_n.$$

(Hierbei haben wir $t_P(x_1\dots x_{n+1}) = k+1$ angenommen.) Andererseits wirkt aber $s^{wx}_{lt_p}$, angewendet auf $\varphi(x_1\dots x_{n+1})$ nach Definition (§ 3) folgendermaßen. Der letzte Buchstabe $y_{i_{n+1}}$ wird an der Stelle $t_P(x_1\dots x_{n+1})$ in das Wort $s^{w}_{lt_p}(\varphi(x_1\dots x_n))$, das nach Induktionsannahme mit $y_1\dots y_n$ übereinstimmt, eingefügt. Also gilt

$$s^{wx}_{lt_P}(\varphi(x_1\dots x_{n+1})) = y_1\dots y_k y_{i_{n+1}}y_{k+1}\dots y_n = f(x_1\dots x_{n+1}),$$

womit der Satz bewiesen ist.

Die Funktionen φ aus Satz 5b und τ_p sind in folgendem Sinne miteinander verträglich: Sind φ und τ_p nach Definition 8 durch

$$\{M_i, M^j_i, N^i_j, s_i, \varphi_i, h_j\}_{n,m} \text{ bzw. } \{R_i, R^j_i, S^i_j, r_i, \tau_i, k_j\}_{n',m'}$$

gegeben, so gilt $n=n'$, $m=m'$ und für alle i und j

$$M_i=R_i,\quad M^j_i=R^j_i,\quad N^i_j=S^i_j \quad\text{und}\quad s_i=r_i.$$

Für die Umkehrung der bisherigen Analyseergebnisse benötigen wir folgende Definitionen.

Definition 15

(a) Wir bezeichnen mit T die Menge aller Funktionen aus $P_1(X, \{A, D, S\})$, die nach Definition 8 dargestellt werden können, wobei folgende zusätzliche Bedingungen erfüllt sind

1. $\forall i\forall w\,(|d_i(w)|=|w|)$
2. $d_i:X^*\to\{D, S\}^*$
3. $h_j(w)=A^{|w|}$ für jedes j
4. $\forall i\forall w\forall x(d_i(wx)=d_i(w)D\leftrightarrow s_i(w)=s_i(wx))$
5. $\forall i\forall w\forall x(|s_i(wx)|-|s_i(w)|\leqq 1)$.

(b) Mit $\mathfrak{T}_1$ bezeichnen wir die Menge aller $t\in\mathfrak{T}$, die aus den Funktionen $\tau\in T$ nach Definition 14 hervorgehen.

Bemerkungen

1. Die Funktionen aus $\mathfrak{T}_1$ sind alle mit wachsender Wortlänge monoton nicht fallend.

2. Die Beziehung zwischen den $t \in \mathfrak{T}_1$ und den $\tau \in T$ ist nicht eineindeutig. Gehört nämlich t nach Definition 14 zu τ, so auch zu jedem τ', das aus τ dadurch hervorgeht, daß gewisse D durch SA ersetzt werden.

Definition 16. Wir bezeichnen mit $\Phi(X, Y)$ die Menge aller Funktionen aus $P_1(X, Y)$, bei deren Darstellung nach Definition 8 alle h_j konstant auf das leere Wort abbilden und alle d_i längentreu sind. Ferner soll gelten $\forall i \forall w \forall x(|s_i(wx)| - |s_i(w)| \leqq 1)$.

$\Phi(X, Y)$ enthält nur sequentielle Funktionen, jedoch auch solche mit unendlichem Gewicht.

Beispiel. Wir wählen $X=\{a, b\}$, $K=\{0, 1\}$, $Y=\{a, b\}$,

$$M_0^1 = \{ba^n b : n \geqq 1\}, \quad M_1^1 = \{a\}, \quad N_1^1 = \{0\}, \quad N_1^2 = \{1\},$$

$d_0(ba^n b) = ba^n b$ für jedes n, $d_1(a) = a$, $d_2(a) = b$, $s_0(ba^n) = s_0(ba^n b) = 10^n$, $s_1(a) = e$, und die übrigen Bestimmungsstücke denken wir uns beliebig, aber im Einklang mit Definition 8 fixiert. Die hierdurch definierte Funktion heiße φ. Dann gilt

$$\varphi(ba^n ba^{n+1}) = ba^n ba^n b = ba^n b\varphi_{ba^n b}(a^{n+1})$$

und für $m > n$

$$\varphi(ba^m ba^{n+1}) = ba^m ba^{n+1} = ba^m b\varphi_{ba^m b}(a^{n+1}).$$

Hieraus folgt $\varphi_{ba^n b} \neq \varphi_{ba^m b}$ für $n \neq m$. Also hat φ kein endliches Gewicht.

Definition 17. Ist $t \in \mathfrak{T}_1$ und $\varphi \in \Phi(X, Y)$, so heißen t und φ verträglich $=_{\mathrm{Df}}$ Es gibt ein mit φ verträgliches (im Sinne der Ausführungen im Anschluß an Satz 5b) $\tau \in T$, aus dem t nach Definition 14 hervorgeht.

Satz 6. Eine längentreue Wortfunktion f von X^* in Y^* ist genau dann im zweiten Sinne von einem treu ausspeichernden synchronen einfachen PDA berechenbar, wenn es miteinander verträgliche Funktionen $\varphi \in \Phi(X, Y)$ und $t \in \mathfrak{T}_1$ gibt mit

$$f = \varphi * s_{lt}.$$

Beweis. Die eine Hälfte der Behauptung folgt ganz leicht aus den Sätzen 5a und 5b unter Berücksichtigung der Definitionen 15, 16 und 17. Es ist noch zu zeigen: Wenn $f = \varphi * s_{lt}$ und φ und t sind verträgliche Funktionen aus $\Phi(X, Y)$ bzw. $\mathfrak{T}_1$, so ist f PDA-berechenbar.

Nach Voraussetzung gibt es ein zu t gehöriges τ, das mit φ verträglich ist. τ und φ seien nach Definition 8 gegeben durch $\{M_i, M_i^j, N_j^i, s_i, \tau_i, h_j\}_{n,m}$ bzw. $\{M_i, M_i^j, N_j^i, s_i, \varphi_i, h_j'\}_{n,m}$. Wir definieren die Funktionen ψ_i durch

$$\psi_i(x_1 \ldots x_N) =_{\mathrm{Df}} H_1 y_1 H_2 y_2 \ldots H_N y_N, \text{ falls } \varphi_i(x_1 \ldots x_\nu) = \varphi_i(x_1 \ldots x_{\nu-1}) y_\nu$$

und

$$\tau_i(x_1 \ldots x_\nu) = \tau_i(x_1 \ldots x_{\nu-1}) H_\nu \quad \text{für} \quad \nu = 1, \ldots, N.$$

Wegen der vorausgesetzten Verträglichkeit ist es möglich, gemäß Definition 8 eine Funktion ψ durch $\{M_i, M_i^j, N_j^i, s_i, \psi_i, h_j\}_{n,m}$ zu definieren.

Wie beim Beweis von Satz 1 verschaffen wir uns zu ψ die gekoppelten Automaten Q_1 und Q_2. Mit den gleichen Bezeichnungen wie dort berechnet Q_1 im Zustand u_i $(i=0, \ldots, n)$ die Funktion ψ_i. Die Konstruktion des PDA $P=[Z, X, Y, K, \mu, z_0]$, der f berechnet, geschieht folgendermaßen. Wir setzen

a) für $v \in V \setminus V''$, $f_2(v, k)=v'$

$$\mu(v, k, \#) = \begin{cases} (v', e, k, 0), & \text{falls } v' \notin V'' \\ (v', k, e, 1), & \text{falls } v' \in V'', \end{cases}$$

b) für $u \in U \setminus U''$, $f_1(u, x)=u'$

$$\mu(u, k, x) = \begin{cases} (u', k, y, 0), & \text{falls } u' \in U'' \wedge g(u, k) = Dy, \\ (u', k, y, 1), & \text{falls } u' \notin U'' \wedge g(u, k) = Dy, \\ (u', kk', e, 1), & \text{falls } u' \notin U'' \wedge g(u, k) = Sk', \\ (u', kk', e, 0), & \text{falls } u' \in U'' \wedge g(u, k) = Sk'. \end{cases}$$

Wie leicht zu sehen ist, berechnet P das gegebene f, womit der Satz bewiesen ist.

Satz 6 läßt sich leicht auf nichtsynchrone einfache treu ausspeichernde PDA verallgemeinern. Wie oben bilden wir die Funktion τ_P. Aus ihr ist zu ersehen, welche Stellen in $f(wx)$ gestrichen werden müssen, um zu $f(w)$ zu gelangen. Ist

$$\tau_P(wx) = \tau_P(w) S^{a(wx)} D^{b(wx)} A^{c(wx)},$$

so sind gerade $a(wx)+b(wx)$ Stellen zu streichen, wobei die erste durch $t_P(wx)$ angegeben wird. Weiter ist zu sehen, daß die Länge von $f(w)$ durch $\lambda(w) = \sum_{u \sqsubseteq w} (a(u)+b(u))$ gegeben ist.

Wir definieren nun eine Folge von Permutationen ϱ_w durch

$$\varrho_e(1) = 1$$

$$\varrho_w(i) = \begin{cases} a+i & \text{für } 1 \leqq i \leqq b(w), \\ a+b+1-i & b(w) < i \leqq a(w)+b(w). \end{cases}$$

Da a, b und λ von P abhängen, hängen auch die ϱ von P ab. Wir setzen $r_P(w) =_{\mathrm{Df}} \varrho_w$ und nennen $[t_P, r_P]$ den zu τ_P gehörigen Typ.

Mit $\tilde{\Phi}(X, Y)$ bezeichnen wir die Klasse aller Funktionen aus $P_1(X, Y)$, bei deren Darstellung gemäß Definition 8 alle Funktionen h_j konstant auf das leere Wort abbilden.

Mit $\tilde{T}$ bezeichnen wir die Klasse aller Funktionen $\tau \in P_1(X, \{A, D, S\})$, bei deren Darstellung gemäß Definition 8 folgende Bedingungen erfüllt sind.

1. $d_i: X^* \to \{D, S\}^*$,
2. $\forall i \forall w \forall x \forall a \big(\exists b (d_i(wx) = d_i(w) S^a D^b \leftrightarrow |s_i(wx)| = a + |s_i(w)|\big)$,
3. $\forall j \forall w \big(h_j(w) = A^{|w|}\big)$.

Mit $\tilde{\mathfrak{T}}_1$ wird die Klasse aller Paare $[t, r]$ bezeichnet, die zu den $\tau \in \tilde{T}$ gehören.

Damit können wir die folgende Verallgemeinerung von Satz 6 formulieren, deren Beweis analog zum Beweis von Satz 6 verläuft.

Satz 6′. Eine Wortfunktion f von X^* in Y^* ist genau dann im zweiten Sinne von einem einfachen treu ausspeichernden PDA berechenbar, wenn es miteinander verträgliche Funktionen $\varphi \in \tilde{\Phi}(X, Y)$ und $\tau \in \tilde{T}$ gibt, so daß für den zu τ gehörigen Typ $[t, r]$ gilt

$$f = \varphi * s_{t,[t,r]}.$$

Bemerkung. Um nichtlängentreue Funktionen berechnen zu können, bei denen $f(wx)$ aus $f(w)$ dadurch hervorgeht, daß an mehreren Stellen des Wortes $f(w)$ Teilwörter eingefügt werden, muß man nichtsynchrone PDA betrachten, die über Befehle folgender Form verfügen: Werden im Zustand z die Eingabe x und das Speichersymbol k gelesen, so wird der Zustand z' angenommen, das Eingabeband um ε verschoben, und es werden

$$\begin{cases} k_{11}\ldots k_{1a_1} \text{ gespeichert,} \\ y_{11}\ldots y_{1b_1} \text{ gedruckt,} \\ c_1 \text{ Buchstaben ausgespeichert} \end{cases} \quad \ldots \quad \begin{cases} k_{n1}\ldots k_{na_n} \text{ gespeichert,} \\ y_{n1}\ldots y_{nb_n} \text{ gedruckt,} \\ c_n \text{ Buchstaben ausgespeichert.} \end{cases}$$

Diese Verallgemeinerung bleibt hier außer Betracht.

§ 5 Funktionen, die im zweiten Sinne von beliebigen PDA berechnet werden

Die Sätze 6 und 6′ zeigen, daß die im zweiten Sinne von treu ausspeichernden PDA berechneten Funktionen sequentielle Funktionen mit anschließender Permutation der Buchstaben der Bildwörter sind. Diese Permutationen hängen dabei von der Art und Weise der Speicherbenutzung ab. Die von beliebigen PDA berechneten Funktionen kann man sich so vorstellen, daß zunächst eine sequentielle Funktion berechnet wird, dann setzt eine Permutation s ein, und danach werden eventuell diejenigen Buchstaben geändert, die ausgespeichert worden sind. Denkt man sich diese letzten Änderungen *vor* der Permutation durchgeführt, so kann auch hier eine Gleichung der Form

$$f = \varphi * s \tag{8}$$

aufgeschrieben werden, wobei allerdings φ i.a. nicht mehr sequentiell ist.

Wir wollen das an einem Beispiel verfolgen und betrachten dazu einen PDA, für dessen τ-Funktion gilt

$$\tau(x_1x_2x_3x_4) = DSSD$$
$$\tau(x_1x_2x_3x_4x_5) = DSSDDA.$$

Hieraus folgt für das zugehörige s_{lt}

$$s_{lt}(x_1x_2x_3x_4) = x_1x_4x_3x_2 \tag{9}$$
$$s_{lt}(x_1x_2x_3x_4x_5) = x_1x_4x_5x_3x_2.$$

f sei die berechnete Funktion, und es sei

$$f(x_1x_2x_3x_4) = y_1y_4y_3y_2 \tag{10}$$
$$f(x_1x_2x_3x_4x_5) = y_1y_4y_5\bar{y}_3\bar{y}_2$$

mit $y_3 \neq \bar{y}_3$, $y_2 \neq \bar{y}_2$. Für φ ergibt sich aus (8), (9) und (10)

$$\varphi(x_1x_2x_3x_4) = y_1y_2y_3y_4$$

$$\varphi(x_1x_2x_3x_4x_5) = y_1\bar{y}_2\bar{y}_3y_4y_5.$$

Dieses φ ist nicht sequentiell, weil die zweite und dritte Stelle stören. Blendet man sie aus, so ist der Rest sequentiell. Solche Funktionen könnte man fastsequentiell nennen.

Wir wollen eine genaue Charakterisierung der möglichen fastsequentiellen φ angeben. Dazu gehen wir bei gegebenem (einfachen) PDA P wie im § 2 von den Automaten Q_1 und Q_2 aus und verfügen somit über die regulären Mengen M_i, M_i^j, N_j^i und die in den einzelnen Zuständen aus V'' bzw. U'' berechneten Funktionen g_i bzw. h_j. Nun erweitern wir den Definitionsbereich der h_j unter Beibehaltung der gleichen Bezeichnung auf $(K \cup Y)^*$: Ist $q_0, \ldots, q_r \in Y^*$ und $p_1, \ldots, p_r \in K^*$ und bezeichnet $h_{j,p}$ den von p bestimmten Zustand von h_j, so setzen wir

$$h_j(q_0p_1q_1p_2q_2 \ldots p_rq_r) =_{\mathrm{Df}} q_0h_j(p_1)q_1h_{j,p_1}(p_2)q_2 \ldots h_{j,p_1 \ldots p_{r-1}}(p_r)q_r. \tag{11}$$

Die Funktion φ aus (8) ergibt sich damit so

$$\left.\begin{array}{l} \varphi(p) = [h_{j_k}(\ldots[h_{j_2}([h_{j_1}(g_0(p_1))^{-1}]^{-1}g_i(p_2))^{-1}]^{-1} \ldots g_{i_{k-1}}(p_k))^{-1}]^{-1}g_{i_k}(p_{k+1}), \\ \text{wobei } p = p_1 \ldots p_{k+1} \text{ und} \\ p_1 \in M_0^{j_1} \wedge s(p_1)^{-1}\alpha N_{j_1}^{i_1} \wedge p_2 \in M_{i_1}^{j_2} \wedge s(p_1p_2)^{-1}\alpha N_{j_2}^{i_2} \wedge \ldots \wedge p_{k+1} \in M_{i_k}. \\ \text{Für } k=0 \text{ ist die Gleichung } \varphi(p) = g_0(p) \text{ gemeint.} \end{array}\right\} \tag{12}$$

Die Ergebnisse des vorigen Paragraphen sind hierin als Spezialfall enthalten. Dazu braucht man nur alle h_j als identische Abbildungen zu wählen, wodurch ein $\varphi \in \Phi(X, Y)$ entsteht.

Um das Ergebnis der Analyse formulieren und umkehren zu können, definieren wir eine Klasse $P_2(X, Y)$ fastsequentieller Wortfunktionen von X^* in Y^*.

Definition 18. $f \in P_2(X, Y) =_{\mathrm{Df}}$.

1. Es gibt eine endliche Menge K, und es existieren reguläre Mengen $M_i, M_i^j \subseteq X^*$ $(i=0, \ldots, n;\ j=1, \ldots, m)$ und reguläre Mengen $N_j^i \subseteq K^*$ $(i=1, \ldots, n;\ j=1, \ldots, m)$ mit den Eigenschaften (a)—(f) aus Definition 8.

2. Es gibt sequentielle Funktionen endlichen Gewichts

$$s_0, \ldots, s_n : X^* \to K^*,$$

$$g_0, \ldots, g_n : X^* \to (K \cup Y)^*,$$

$$h_1, \ldots, h_m : K^* \to Y^*,$$

so daß f durch (12) gegeben ist, wobei s durch (6) definiert ist und alle h_j durch (11) auf $(K \cup Y)^*$ fortgesetzt sind.

Wir wissen bereits aus dem eingangs dieses Paragraphen angegebenen Beispiel, daß $P_2(X, Y)$ auch nichtsequentielle Funktionen enthält.

Der folgende Satz ist eine Verallgemeinerung von Satz 6′. Der Beweis verläuft analog zum Beweis von Satz 6.

Satz 6″. Eine Wortfunktion f von X^* in Y^* ist genau dann im zweiten Sinne von einem einfachen PDA berechenbar, wenn es miteinander verträgliche Funktionen $\varphi \in P_2(X, Y)$ und $\tau \in \tilde{T}$ gibt, so daß für den zu τ gehörigen Typ $[t, r]$ gilt

$$f = \varphi * s_{l,[t,r]}.$$

Функции вычисляемые автоматами с магазинной памятью

Рассматриваются два вида вычисления функций автоматами с магазинной памятью. Описываются классы вычисляемых таким образом функций независимо от понятия автомата. Вычисляемые в первом смысле функции являются последовательностными функциями состоящими в некотором смысле но кускам из последовательностных функций конечного веса. Вычисляемые во втором смысле функции являются квази-последовательностными функциями точно описанного вида.

SEKTION MATHEMATIK DER
FRIEDRICH SCHILLER UNIVERSITÄT
69 JENA, DDR
UNIVERSITÄTSHOCHHAUS

Literatur

[1] EVEY, R. J., The theory and application of pushdown store maschines, mathematical linguistics and automatic translation, *Comput. Lab. Rept.*, Harvard University, v. NSF-10, May, 1963.
[2] GINSBURG, S., *The mathematical theory of contextfree languages,* Mc Graw-Hill Book Comp., New York, 1966.
[3] GINSBURG, S., G. F. ROSE, Preservation of languages by transducers, *Information and Control.*, v. 9, 1966, pp. 153—176.
[4] GLUSCHKOW, W. M., *Theorie der abstrakten Automaten,* VEB Verlag der Wissenschaften, Berlin 1963.
[5] RANEY, G. N., Sequential functions, *J. Assoc. Comput. Mach.*, v. 5, 1958, pp. 177—180.
[6] WECHSUNG, G., Quasisequentielle Funktionen, *Acta Cybernet.*, c. 2, 1973, pp. 23—33.

(Eingegangen am 27. Juli 1972)

Algorithm for constructing of university timetables and criterion of consistency of requirements

by A. KRECZMAR

1. Introduction

The construction of timetable by means of a computer is the subject of numerous publications. In all these papers two similar problems are investigated:

(1) constructing a school timetable,

(2) constructing a timetable for university department.

In the first case there are given three sets: a set of classes, a set of teachers and a set of time periods. One lesson can be interpreted as a meeting of a teacher and a class for one period. The problem is to schedule all lessons so that no teacher and no class has two or more different lessons at the same hour. Moreover, we must also take into consideration the problem of the so-called preassignments, it means that lessons are not available at every period of time.

The second case is more complicated. We shall indicate below three requirements which will be the subject of further investigations.

(a) University department consists of years, sections groups etc. which can have certain common jobs.

(b) One lecture can last more than one time period.

(c) Every lecture must take place in a given room; therefore apart from sets just defined there is given a fourth set, a set of rooms.

In the present paper we shall give a condition necessary and sufficient for existence of university timetable and an algorithm of constructing of it. We shall use some basic notions of the theory of graphs such as; an independent set, a chromatic number or a colouring of a graph whose definitions the reader can find in [1].

2. Two definitions of timetable

For the first time the timetable problem was defined by Gotlieb [2] as follows.

Let $T=\{t_i\}$ $(i\leqq n)$ be the set of teachers, $C=\{c_j\}$ $(j\leqq n)$ the set of classes and $H=\{h_k\}$ $(k\leqq p)$ the set of time periods.

Let us consider two matrices: $A=\{a_{ij}\}$ $(i\leqq m, j\leqq n)$ where a_{ij} is an integer pointing out how many times a teacher t_i must meet with a class c_j and $B=\{b_{ijk}\}$ $(i\leqq m, j\leqq n, k\leqq p)$ where element b_{ijk} is 1 if teacher t_i can meet class c_j at hour h_k and 0 in the opposite case. A pair $\langle A, B\rangle$ defines the set of all requirements.

Definition 1. The matrix $S=\{s_{ijk}\}$ $(i\leqq m, j\leqq n, k\leqq p)$ fulfilling the conditions:

$$\sum_{i=1}^{m} s_{ijk} \leqq 1 \qquad (1) \qquad\qquad \sum_{j=1}^{n} s_{ijk} \leqq 1 \qquad (2)$$

$$\sum_{k=1}^{p} s_{ijk} = a_{ij} \qquad (3) \qquad\qquad \text{If } s_{ijk}=1 \text{ then } b_{ijk}=1 \qquad (4)$$

for arbitrary $i\leqq m$, $j\leqq n$, $k\leqq p$ is called a timetable for the requirements $\langle A, B\rangle$.

Gotlieb describes in his paper an algorithm of constructing the timetable S for given requirements $\langle A, B\rangle$. The method used by him is based on theorem of P. Hall [3] on distinct representatives of subsets. Unfortunatly this algorithm does not answer the questions whether timetable exists and whether solutions attained are all which satisfy conditions (1)—(4).

In order to introduce our method of reducing the timetable problem to the colouring of graph we must change a little the definition of timetable. In 2. 1 we shall show that this new definition is an extension of the first one.

Now, let $L=\{l_i\}$ $(i\leqq q)$ be the set of all lessons. With every l_i $(i\leqq q)$ we associate the set $g_i\subset H$, of time periods at which lesson l_i is admissible. The interference condition between lessons is described by the relation $\varrho\subset L\times L$ fulfilled if the lessons can not be scheduled at the same our.

Definition 2. A sequence $x=\langle h^1, \ldots, h^q\rangle$ of elements of H will be called a timetable for the family $G=\{g_i\}$ $(i\leqq q)$ and the relation ϱ iff

$$h^i\in g_i \qquad i=1, \ldots, q \qquad (5)$$

$$\text{if } l_i\varrho l_j \text{ then } h^i\neq h^j \quad i, j\leqq q. \qquad (6)$$

In fact, these conditions say that if lesson l_i is scheduled at hour h^i then from (5) l_i is admissible at h^i and from (6) lessons never interfere.

Now we shall show that definition 1 can be replaced by the other.

2. 1. For arbitrary requirements $\langle A, B\rangle$ there exist set L, family G and relation ϱ so that there is a one-to-one correspondence between timetables S and x.

Proof. For the given matrix A we can easily define L as a set of corresponding pairs $\langle t_i, c_j\rangle$. The relation ϱ is given by the following equivalence:

$$\langle t_i, c_j\rangle\varrho\langle t_u, c_w\rangle\equiv(i=u)\wedge(j=w).$$

Next

$$g_{ij}=\{h_k: b_{ijk}=1\}$$

is a set of time periods admissible for $\langle t_i, c_j\rangle$. By a direct verification we see that equivalence

$$s_{ijk}=1\equiv h_k\in g_{ij}$$

determines demanded correspondence.

Let us observe that definition 2 is an essential extension of the first one. In this definition we can take into account the condition of type (a) and many others not mentioned here, by appropriate determination of ϱ. So, if two lessons l_i, l_j for

some reason or other cannot be scheduled at the same hour we put $l_i \varrho l_j$, and $\neg\, l_i \varrho l_j$ if this is not the case.

To compare requirements $\langle A, B\rangle$ to these described by G and ϱ we shall consider an example due to Cisma and Gotlieb [4].

In their example $n=m=p=3$, $A=\{a_{ij}\}$ $(i\leqq 3,\ j\leqq 3)$ where $a_{ij}=1$ and the matrix B is following:

$$b_{1jk}=\begin{matrix}1&1&0\\0&1&1\\1&0&1\end{matrix}\qquad b_{2jk}=\begin{matrix}1&0&1\\1&1&1\\1&1&0\end{matrix}\qquad b_{3jk}=\begin{matrix}0&1&1\\1&1&0\\1&1&0\end{matrix}$$

For these requirements Hall's conditions are fulfilled but a timetable S does not exist.

In the new definition the set L contains all pairs $\langle t_i, c_j\rangle$ $i\leqq 3$, $j\leqq 3$. Subsets g_{ij} are following:

$$g_{11}=\{h_1, h_2\}\ g_{12}=\{h_2, h_3\}\ g_{13}=\{h_1, h_3\}$$

$$g_{21}=\{h_1, h_3\}\ g_{22}=\{h_1, h_2, h_3\}\ g_{23}=\{h_1, h_3\}$$

$$g_{31}=\{h_2, h_3\}\ g_{32}=\{h_1, h_2\}\ g_{33}=\{h_1, h_2, h_3\}$$

then $G=\{g_{11}, g_{12}, g_{13}, g_{21}, g_{22}, g_{23}, g_{31}, g_{32}, g_{33}\}$. The relation ϱ can be displayed as a matrix:

$$\varrho=\begin{array}{c|ccccccccc} & l_1&l_2&l_3&l_4&l_5&l_6&l_7&l_8&l_9\\ \hline l_1&0&1&1&1&0&0&1&0&0\\ l_2&1&0&1&0&1&0&0&1&0\\ l_3&1&1&0&0&0&1&0&0&1\\ l_4&1&0&0&0&1&1&1&0&0\\ l_5&0&1&0&1&0&1&0&1&0\\ l_6&0&0&1&1&1&0&0&0&1\\ l_7&1&0&0&1&0&0&0&1&1\\ l_8&0&1&0&0&1&0&1&0&1\\ l_9&0&0&1&0&0&1&1&1&0\end{array}$$

where $\varrho_{ij}=1\equiv l_i \varrho l_j$ (see also figures 1 and 2).

3. Graph of a timetable

We denote by F the set $\{l_1, \ldots, l_q, h_1, \ldots, h_p\}$ and by $\pi\subset F\times F$ the binary relation defined as follows:

$$l_i \pi l_j \equiv l_i \varrho l_j \tag{7}$$

$$l_i \pi h_j \equiv h_j \notin g_i \qquad h_j \pi l_i \equiv l_i \pi h_j \tag{8}$$

$$h_i \pi h_j \equiv i \neq j \tag{9}$$

The graph $E=\langle F, \pi\rangle$ where F is a set of vertices and π a set of edges will be called the graph of a timetable. Since a relation π is symmetric and antireflexive then there exists the unique chromatic number of graph E.

Now we can establish the main result of the present paragraph.

3. 1. A timetable $x=\langle h^1, \ldots, h^q\rangle$ exists iff a chromatic number of graph $E=\langle F,\pi\rangle$ is equal to the number of elements of H(E is p-chromatic).

Proof. Let $x=\langle h^1, \ldots, h^q\rangle$ be a timetable fulfilling (5) and (6) and let $D_k=\{h_k\}\cup\{l_i: h_k=h^i\}$ $(k=1, \ldots, p)$. We shall show that the sets $D_1, \ldots, D_p$ form a family of independent sets which covers the graph E.

Really, if $l_i, l_j\in D_k$ then $h^i=h^j=h_k$ and from (6) $\neg(l_i\varrho l_j)$. Next if $l_i\in D_k$ then $h_k=h^i$ and from (5) $h_k\in g_i$. By (7) $\neg(l_i\pi l_j)$ and by (8) $\neg(l_i\pi h_k)$ so D_k are independent. Since for every l_i exists D_k such that $l_i\in D_k$, sets $D_1, \ldots, D_p$ cover the graph E, it means E is at least p-chromatic. On the other hand the chromatic number of E cannot be less than p, because there is a complete subgraph of the order p containing all vertices h_k $(k=1, \ldots, p)$.

Thus necessity is proved.

Now, let the family $D_1, \ldots, D_p$ denote a covering of graph E. As all D_k $(k=1; \ldots, p)$ are independent and every h_k must belong to some D_k we can associate with every D_k one element h_k.

Now for every l_i $(i=1, \ldots, q)$ we choose an arbitrary h_k such that $l_i\in D_k$. If h^i stands for this h_k then a sequence $x=\langle h^1, \ldots, h^q\rangle$ is a timetable.

In fact, $l_i, h^i\in D^i$ so $\neg(h^i\pi l_i)$ and by (8) $h^i\in g_i$. If for some l_i, l_j $(i\neq j)$ $h^i=h^j$ then l_i, l_j belong to the same D_k, it means $\neg(l_i\pi l_j)$ and by (7) $\neg(l_i\varrho l_j)$.

It ends the proof of sufficiency.

Immediatly from 3. 1. we have

3. 2. There is an effective procedure of constructing for arbitrary p-colouring of graph E a timetable x if it exists.

The constructing procedure was given in the proof of sufficiency in 3. 1.

So far as can be seen 3. 1 establishes the condition necessary and sufficient for the existence of timetable. In 4. it will be shown how to obtain all p-colourings of graph E and due to 3. 2 we shall be able to obtain all sequences satisfying (5) and (6).

4. Algorithm 1

Efficient methods for graph colouring were investigated by many authors ([5], [6]) and any of them may be used here.

In this paragraph we shall present a simple idea of J. Wiessman [6] who applied boolean transformations to this problem.

Let us consider a graph $E=\langle F, \pi\rangle$ for requirements given in 2. (see figure 1). We treat an ordered set of all vertices as a set of boolean variables. A boolean polynomial:

$$f_1 = (l_2+l_1)\ (l_3+l_1l_2)\ (l_4+l_1)\ (l_5+l_2l_4)$$
$$(l_6+l_3l_4l_5)\ (l_7+l_1l_4)\ (l_8+l_2l_5l_7)\ (l_9+l_3l_6l_7l_8)$$
$$(h_1+l_2l_7)\ (h_2+l_3l_4h_1)\ (h_3+l_1l_6l_8h_1h_2)$$

where every disjunction contains a negation of successive vertex and conjunction of negations of all precedent coincident vertices with this one, is transformed into the disjunctive-conjunctive normal form $DC(f_1)$.

Complements of the set of vertices which occur in successive conjunctions of $DC(f_1)$ are maximal independent sets ([6]). Thus

$$D_1=\{l_3, l_5, h_1\} \quad D_2=\{l_3, l_4, l_8, h_1\} \quad D_3=\{l_1, l_6, l_8, h_1\}$$
$$D_4=\{l_4, l_9, h_1\} \quad D_5=\{l_1, l_5, l_9, h_1\} \quad D_6=\{l_5, l_7, h_2\}$$
$$D_7=\{l_2, l_6, l_7, h_2\} \quad D_8=\{l_2, l_9, h_2\} \quad D_9=\{l_1, l_6, l_8, h_2\}$$
$$D_{10}=\{l_1, l_5, l_9, h_2\} \quad D_{11}=\{l_3, l_5, l_7, h_3\} \quad D_{12}=\{l_2, l_4, l_9, h_3\}$$
$$D_{13}=\{l_3, l_4, h_3\} \quad D_{14}=\{l_2, l_7, h_3\} \quad D_{15}=\{l_5, l_9, h_3\}.$$

In order to obtain all p-colourings of E let us observe that,

$$l_1\in D_3 \text{ or } l_1\in D_5 \text{ or } l_1\in D_9 \text{ or } l_1\in D_{10},$$
$$l_2\in D_7 \text{ or } l_2\in D_8 \text{ or } l_2\in D_{12} \text{ or } l_2\in D_{14} \text{ etc.}$$

Then a boolean polynomial

$$f_2 = (D_3+D_5+D_9+D_{10})\ (D_7+D_8+D_{12}+D_{14})\ (D_1+D_2+D_{11}+D_{13})$$
$$(D_2+D_4+D_{12}+D_{13})\ (D_1+D_5+D_6+D_{10}+D_{11}+D_{15})$$
$$(D_3+D_7+D_9)\ (D_6+D_7+D_{11}+D_{14})\ (D_2+D_3+D_9)$$
$$(D_4+D_5+D_8+D_{10}+D_{12}+D_{15})\ (D_1+D_2+D_3+D_4+D_5)$$
$$(D_6+D_7+D_8+D_9+D_{10})\ (D_{11}+D_{12}+D_{13}+D_{14}+D_{15})$$

transformed into the disjunctive-conjunctive normal form $DC(f_2)$ determines all coverings of graph E. In fact, if a conjunctive $D_{i_1}, \ldots, D_{i_k}$ occurs in $DC(f_2)$ then every vertex must belong to a certain D_{ij} $(j\leqq k)$. Since we search only p-colourings in every step of transformation those conjunctions which have more than p elements must be removed. In our example there is no conjunction in $DC(f_2)$ which has 3 elements then in virtue of 3. 1 a timetable x for these requirements does not exist.

But if the number of the edges of E is reduced by deleting an edge between l_6 and h_3, in the polynomial f_1 we obtain $h_3+l_1l_8h_1h_2$ instead of $h_3+l_1l_6l_8h_1h_2$, then $D_{14}=\{l_2, l_6, l_7, h_3\}$ and next in f_2 there is $(D_3+D_7+D_9+D_{14})$ instead of $(D_3+D_7+D_9)$. Thus in $DC(f_2)$ occurs the conjunction $D_2\ D_{10}\ D_{14}$ which gives a unique timetable $x=\langle h_2, h_3, h_1, h_1, h_2, h_3, h_3, h_1, h_2\rangle$.

An interesting problem arises in the case of inconsistency of requirements: What is a minimal number of edges whose removing decreases a chromatic number of graph E?

This problem is strictly connected with the notion of the critical graph which was investigated by G. A. Dirac ([7], [8]).

5. Multiperiod jobs

In the case of condition (b) apart from sets L, H, G and relation ϱ there is given a function $n: L \to N$ (set of integers) the value of which $n(l_i)=n_i$ defines how many consecutive time periods l_i must last. So, $n_i=1$ defines a single period, $n_i=2$ a double period etc.

We denote by $\langle h_k, n\rangle$ a time interval beginning at h_k and lasting n time periods. It means that

$$\langle h, n\rangle = \{h_k, h_{k+1}, \ldots, h_{k+n-1}\}$$

provided that h_k, h_{k+1} $k=1, \ldots, p-1$ are consecutive periods.

Now, for requirements with function n we must introduce a new definition of timetable.

Definition 3. A sequence $x=\langle h^1, \ldots, h^q\rangle$ will be called a timetable for requirements with function n iff

$$\langle h^i, n_i\rangle \subset g_i \quad i=1, \ldots, q \tag{10}$$

$$\text{If } l_i \varrho l_j \text{ then } \langle h^i, n_i\rangle \cap \langle h^j, n_j\rangle = \varnothing \text{ (empty set).} \tag{11}$$

These two condition correspond with (5) and (6) where one time period h_i is changed by a whole interval $\langle h^i, n_i\rangle$.

5. 1. A timetable $x=\langle h^1, \ldots, h^q\rangle$ exists iff there is a covering $D=\{D_1, \ldots, D_p\}$ of graph $E=\langle F, \pi\rangle$ such that D_k, $k=1, \ldots, p$ are independent sets and

$$h_k \in D_k \quad k=1, \ldots, p \tag{12}$$

$$\text{for every } i=1, \ldots, q \text{ exists } k_i \leqq p-n_i+1 \text{ such that} \tag{13}$$

$$l_i \in \bigcap_{j=k_i}^{k_i+n_i-1} D_j \quad (l_i \text{ belongs to the successive } n_i \text{ independent sets})$$

Proof. Let $x=\langle h^1, \ldots, h^q\rangle$ be a timetable and let $D_k=\{h_k\} \cup \{l_i: h_k \in \langle h^i, n_i\rangle\}$. The proof of independence of D_k is analogous as in 3. 1. The condition (12) is immediate. Let k_i stand for an index of h^i in the set H. Thus $l_i \in D_{k_i} \cap D_{k_i+1} \cap \ldots \ldots \cap D_{k_i+n_i-1}$ which proves the condition (13).

Let us assume that independent sets $D_1, \ldots, D_p$ satisfy (12) and (13). We can define a timetable x as a sequence $\langle h_{k_1}, h_{k_2}, \ldots, h_{k_q}\rangle$. For $h_k \in \langle h_{k_i}, n_i\rangle$ by (12) $h_k \in D_k$ and by (13) $l_i \in D_k$ which is equivalent $\neg(h_k \pi l_i)$. From (8) $\neg(h_k \pi l_i)$ iff $h_k \in g_i$ thus $\langle h_k, n_i\rangle \subset g_i$. In order to prove (11) let us assume that $\langle h_{k_i}, n_i\rangle \cap \langle h_{k_j}, n_j\rangle \neq \varnothing$. It means that for $h_k \in \langle h_{k_1}, n_i\rangle \cap \langle h_{k_j}, n_j\rangle$ in virtue of (9) and (13) $l_i \in D_k$, $l_j \in D_k$. Thus l_i, l_j belong to the same D_k which implies $\neg(l_i \varrho l_j)$.

6. Algorithm 2

The theorem 5. 1 establishes the condition necessary and sufficient for the existence of a timetable with multiperiod jobs. First, so as in algorithm 1 all maximal independent sets $D=\{D_j\}$ of graph E must be achieved.

The second part of procedure we exemplify by colouring the graph from figure 2. This graph we obtain from the graph displayed on figure 1 by adding one vertex h_4,

five edges h_4h_1, h_4h_2, h_4h_3, h_4l_9 and removing one edge h_3l_6. The function n is determined in this example as follows: $n_2=n_6=n_7=2$, $n_1=n_3=n_4=n_5=n_8=n_9=1$.

The family of maximal independent sets for this graph is increased by five sets

$$D_{16}=\{l_1, l_5, h_4\} \quad D_{17}=\{l_1, l_6, l_8, h_4\} \quad D_{18}=\{l_2, l_6, l_7, h_4\}$$
$$D_{19}=\{l_3, l_5, l_7, h_4\} \quad D_{20}=\{l_3, l_8, h_4\}.$$

Since $n_1=1$ the vertex l_1 satisfies condition $l_1\in D_3\cup D_5\cup D_9\cup D_{11}\cup D_{16}\cup D_{17}$. Next, for the vertex l_2 $n=2$, so

$$l_2\in(D_7\cap D_{12})\cup(D_7\cap D_{14})\cup(D_8\cap D_{12})\cup(D_8\cap D_{14})\cup(D_{12}\cap D_{18})\cup(D_{14}\cap D_{18})$$

Similarly for l_6 and l_7. In the analogous way as in algorithm 1 we verify that a boolean polynomial:

$$f_3=(D_3+D_5+D_9+D_{10}+D_{16}+D_{17})\ (D_7D_{12}+D_7D_{14}+D_8D_{12}+D_8D_{14}+D_{12}D_{18}+D_{14}D_{18})$$
$$(D_1+D_2+D_{11}+D_{13}+D_{19}+D_{20})\ (D_2+D_4+D_{12}+D_{13})\ (D_1+D_5+D_6+$$
$$+D_{10}+D_{11}+D_{15}+D_{16}+D_{19})\ (D_3D_7+D_3D_9+D_7D_{14}+D_9D_{14}+D_{14}D_{17}+D_{14}D_{18})$$
$$(D_6D_{11}+D_6D_{14}+D_7D_{11}+D_7D_{14}+D_{11}D_{18}+D_{11}D_{19}+D_{14}D_{18}+D_{14}D_{19})\ (D_2+D_3+$$
$$+D_9+D_{17}+D_{20})\ (D_4+D_5+D_8+D_{10}+D_{12}+D_{15})\ (D_1+D_2+D_3+D_4+D_5)$$
$$(D_6+D_7+D_8+D_9+D_{10})\ (D_{11}+D_{12}+D_{13}+D_{14}+D_{15})\ (D_{16}+D_{17}+D_{18}+D_{19}+D_{20})$$

transformed into the disjunctive-conjunctive normal form gives all coverings which satisfy (12), (13). In this case we obtain only one covering containing 4 elements: $D_2D_{10}D_{14}D_{18}$ and $x=\langle h_2, h_3, h_1, h_1, h_2, h_3, h_3, h_1, h_2\rangle$.

If for some l_i $n_i>2$ a correspondent boolean expression consists of all conjunctions which have n elements $D_{k_1}, D_{k_2}, \ldots, D_{k_q}$, such that l_i belongs to every D_{k_j} and $h_{k_1}, h_{k_2}, \ldots, h_{k_q}$ are consecutive time periods.

Obviously, in this expression conjunctions in which time periods belong to two different days or contain a lunch break must be omited.

7. Room problem

In the extension of timetable problem taking into account the condition (c) there is given *a* set $R=\{r_j\}$ $j\leqq s$ of rooms. As in the case of lectures with every r_j we associate a set $f_j\subset H$, time periods at which room r_j is available. Moreover, there are rooms not fitting to every lecture. This condition is described by a relation $\sigma\subset L\times R$ fulfilled if lecture l_i can take place in room r_j.

Definition 4. A pair $\langle x, y\rangle$ where x is a timetable for the set L and y is a sequence $\langle r^1, r^2, \ldots, r^q\rangle$ rooms will be called a timetable for sets L and R iff

$$l_i\sigma r^i \quad i=1, \ldots, q \tag{14}$$
$$\langle h^i, n_i\rangle\subset f_i \quad i=1, \ldots, q \tag{15}$$
$$\text{if } r^i=r^j \text{ then } \langle h^i, n_i\rangle\cap\langle h^j, n_j\rangle=\varnothing. \tag{16}$$

The condition (14) says that a lecture l_i can take place in a room r^i, (15) that this room is available at hours $\langle h^i, n_i\rangle$ and finally (16) assures that no room is used simulteneously for two lectures.

Now, if there is given a timetable $x=\langle h^1, \ldots, h^q\rangle$ we can define a new graph $E=\langle I, \pi_x\rangle$ where a set of vertices $I=\{l_1, \ldots, l_q, r_1, \ldots, r_s\}$ and the relation π_x is following:

$$r_i\pi_x r_j \equiv i\neq j \tag{17}$$

$$l_i\pi_x r_j \equiv \neg(l_i\sigma r_j)\vee\neg(\langle h^i, n_i\rangle\subset f_j) \tag{18}$$

$$l_i\pi_x l_j \equiv \langle h^i, n_i\rangle\cap\langle h^j, n_j\rangle\neq\varnothing. \tag{19}$$

Of course, $r_j\pi_x l_i \equiv l_i\pi_x r_j$.

7. 1. If x is a timetable for L then a timetable $\langle x, y\rangle$ exists iff graph E_x is s-chromatic.

Proof. If $y=\langle r^1, \ldots, r^q\rangle$ fulfills (14)—(16) then sets $D_j=\{r_j\}\cup\{l_i: r^i=r_j\}$ are independent. In fact for $l_i\in D_j$ from (14) $l_i\sigma r_j$ and from (15) $\langle h^i, n_i\rangle\subset f_j$ thus by (18) $\neg l_i\pi_x r_j$. On the other hand if $l_i, l_k\in D_j$ then $r^i=r^k=r_j$ and by (16) $\langle h^i, n_i\rangle\cap\langle h^k, n_k\rangle=\varnothing$ which gives in virtue of (19) that $\neg l_i\pi_x l_k$.

Since sets D_j $j=1, \ldots, s$ are independent and cover the graph E_x, its chromatic number is equal s.

Now, let a family $D_1, \ldots, D_s$ denotes a covering of E. By (17) we can assume that $r_j\in D_j$ $j=1, \ldots, s$. Let us define $y=\langle r^1, \ldots, r^q\rangle$ where r^i is an arbitrary room belonging to the same set D_j as l_i. So, $\neg l_i\pi_x r^i$ gives by (18) that $l_i\sigma r^i$ and $\langle h^i, n_i\rangle\subset f^i$. If $\langle h,^i n_i\rangle\cap\langle h^j, n_j\rangle\neq\varnothing$ then by (19) $l_i\pi_x l_j$ and l_i, l_j cannot belong to the same D_k. This proves that $r^i\neq r^j$.

8. Algorithm 3

The algorithm consists of two phases. First, all timetables x by the help of algorithm 2 are generated. The second phase is concerned with assignment of rooms. In the analogous way as in 4. the problem is reduced to the colouring of the graph. Since two timetables $\langle x, y\rangle$ and $\langle x, z\rangle$ where $y\neq z$ may be treated as equivalent we break the realization of Wiessman's method after an achievement of first colouring. If a graph E is not s-chromatic a timetable $\langle x, y\rangle$ for the given sequence x does not exist (theorem 7. 1).

We must investigate the next sequence x. A choice of this sequence can depend on desirable features of timetable such as the distribution of lectures over the days and the week, the maximal possibility of choice in the case of facultative jobs etc.

Let us end the presentation of methods hitherto described by an example considered in 6 with following room requirements:

$$R=\{r_1, r_2, r_3, r_4\}$$

$$f_1=\{h_1, h_3, h_4\}\quad f_2=\{h_1, h_2, h_3, h_4\}$$

$$f_3=\{h_1, h_2\}\quad f_4=\{h_1, h_2, h_3, h_4\}$$

$$\sigma = \begin{array}{c|ccccccccc} & l_1 & l_2 & l_3 & l_4 & l_5 & l_6 & l_7 & l_8 & l_9 \\ r_1 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 1 \\ r_2 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 1 \\ r_3 & 1 & 1 & 1 & 1 & 0 & 1 & 1 & 0 & 0 \\ r_4 & 1 & 1 & 0 & 1 & 1 & 1 & 0 & 1 & 0 \end{array}$$

For the sequence $x=\langle h_2, h_3, h_1, h_1, h_2, h_3, h_3, h_1, h_2\rangle$ the graph $E_x=\langle I, \pi_x\rangle$ (see figure 3) has eleven maximal independent sets:

$D_1=\{l_2, l_3, r_1\}$ $D_2=\{l_3, l_7, r_1\}$ $D_3=\{l_2, l_4, r_1\}$ $D_4=\{l_4, l_7, r_1\}$

$D_5=\{l_1, l_6, l_8, r_2\}$ $D_6=\{l_5, l_6, l_8, r_2\}$ $D_7=\{l_1, l_7, l_8, r_2\}$

$D_8=\{l_5, l_7, l_8, r_2\}$ $D_9=\{l_5, l_8, r_3\}$ $D_{10}=\{l_8, l_9, r_3\}$

$D_{11}=\{l_3, l_7, l_9, r_4\}$.

Two 4-colourings are determined by the conjunction $D_3D_5D_9D_{11}$, thus there are two equivalent timetables

$$\langle x,\langle r_2, r_1, r_4, r_1, r_3. r_2, r_4, r_2, r_4\rangle\rangle \quad \text{and} \quad \langle x, \langle r_2, r_1, r_4, r_1, r_3, r_2, r_4, r_3, r_4\rangle\rangle.$$

9. Acknowlegments

The author acknowledges with pleasure and thanks professor Stanislaw Turski for his patronage, dr Andrzej Salwicki for his constant and unlimited support and Mr Waclaw Pankiewicz who has read the paper and helped to make corrections.

$$\pi = \begin{array}{c|ccccccccc|ccc} & l_1 & l_2 & l_3 & l_4 & l_5 & l_6 & l_7 & l_8 & l_9 & h_1 & h_2 & h_3 \\ l_1 & & & & & & & & & & 0 & 0 & 1 \\ l_2 & & & & & & & & & & 1 & 0 & 0 \\ l_3 & & & & & & & & & & 0 & 1 & 0 \\ l_4 & & & & & & & & & & 0 & 1 & 0 \\ l_5 & & & & & \varrho & & & & & 0 & 0 & 0 \\ l_6 & & & & & & & & & & 0 & 0 & 1 \\ l_7 & & & & & & & & & & 1 & 0 & 0 \\ l_8 & & & & & & & & & & 0 & 0 & 1 \\ l_9 & & & & & & & & & & 0 & 0 & 0 \\ h_1 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 \\ h_2 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 \\ h_3 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 1 & 0 \end{array}$$

$$F = \{l_1, l_2, l_3, l_4, l_5, l_6, l_7, l_8, l_9, h_1, h_2, h_3\}$$

Figure 1

	l_1	l_2	l_3	l_4	l_5	l_6	l_7	l_8	l_9	h_1	h_2	h_3	h_4
l_1										0	0	1	0
l_2										1	0	0	0
l_3										0	1	0	0
l_4										0	1	0	0
l_5					ϱ					0	0	0	0
l_6										0	0	0	0
$\pi = l_7$										1	0	0	0
l_8										0	0	1	0
l_9										0	0	0	1
h_1	0	1	0	0	0	0	1	0	0	0	1	1	1
h_2	0	0	1	1	0	0	0	0	0	1	0	1	1
h_3	1	0	0	0	0	0	0	1	0	1	1	0	1
h_4	0	0	0	0	0	0	0	0	1	1	1	1	0

$$F = \{l_1, l_2, l_3, l_4, l_5, l_6, l_7, l_8, l_9, h_1, h_2, h_3, h_4\}$$

Figure 2

	l_1	l_2	l_3	l_4	l_5	l_6	l_7	l_8	l_9	r_1	r_2	r_3	r_4
l_1	0	0	0	0	1	0	0	0	1	1	0	1	1
l_2	0	0	0	0	0	1	1	0	0	0	1	1	1
l_3	0	0	0	1	0	0	0	1	0	0	1	1	0
l_4	0	0	1	0	0	0	0	1	0	0	1	1	1
l_5	1	0	0	0	0	0	0	0	1	1	0	0	1
l_6	0	1	0	0	0	0	1	0	0	1	0	1	1
$\pi_x = l_7$	0	1	0	0	0	1	0	0	0	0	0	1	0
l_8	0	0	1	1	0	0	0	0	0	1	0	0	1
l_9	1	0	0	0	1	0	0	0	0	1	1	0	0
r_1	1	0	0	0	1	1	0	1	1	0	1	1	1
r_2	0	1	1	1	0	0	0	0	1	1	0	1	1
r_3	1	1	1	1	0	1	1	0	0	1	1	0	1
r_4	1	1	0	1	1	1	0	1	0	1	1	1	0

$$I = \{l_1, l_2, l_3, l_4, l_5, l_6, l_7, l_8, l_9, r_1, r_2, r_3, r_4\}$$

Figure 3

Алгоритм для получения расписания университета и критерий согласования с требованиями

В первой части приводим формальное определение расписания учебных занятий, в котором появляется только очень простая модель [2]. Эквивалентное определение в терминах раскраски графов позволяет сформулировать необходимые и достаточные условия существования расписания занятий. Предлагается алгоритм построения расписания и приводится пример, который неразрешим комбинаторными методами (взят из [4]).

Далее приводятся более сложные модели с учетом неравнодлительных занятий и проблемой залов. Все они записаны терминами проблемы раскраски графов. Приводятся соответствующие критерии существования и алгоритмы построения расписания учебных занятий.

INSTITUTE OF COMPUTER SCIENCE
WARSAW UNIVERSITY

References

[1] Ore, O., Theory of graphs, *Amer. Math. Soc. Transl.*, v. 38.
[2] Gotlieb, C. C., The construction of class-teacher timetables, *Proc. IFIP Congres 1962*, Amsterdam, 1963, p. 73.
[3] Hall, P., On representatives of subsets, *J. London Math. Soc.*, v. 10, 1935, pp. 26—30.
[4] Cisma, J., C. C. Gotlieb, Test on a computer method for constructing school timetables, *Comput. Surveys. ACM*, v. 7, 1964, p. 160.
[5] Welsh, D., Upperbounds for chromatic number of graphs, *Comput. J.*, 1967, pp. 85—86.
[6] Wiessman, J., Boolean algebra map colouring and interconnections, *Amer. Math. Monthly*, v. 69, 1962, p. 608.
[7] Dirac, G. A., Circuits in critical graphs, *Monatsh. Math.* v. 59, 1955, pp. 178—187.
[8] Dirac, G. A., The structure of k-chromatic graphs, *Fund. Math.* v. 40, 1953, pp. 42—55.
[9] Kreczmar, A., Kryterium istnienia i algorytm ukladania rozkladu zajec szkolnych, (in Polish) *Wydawnictwa U. W.* (Warsaw University) Institute of Computer Science, 1970, p. 23.

(Received August 26, 1972)

A man-machine principle applied to human behavioural tests

By Z. HANTOS and I. MADARÁSZ

The subject of our paper is to draw up a principle of a special man-machine relation and the examination of conclusions originating in this principle, as well as of consequences emerging during the practical realization.

In this system there is an automaton with a relatively great degree of autonomy, a capacity of decision making, interposed by the cognitive subject into the process of cognition, and in this manner a connection is established between the automaton and certain mechanisms of the system to be studied which is a human being. This automatic cognitive system intervenes in the activity of the subject investigated, makes a diagnosis of the state of the system by analysing the responses received.

The special nature of a connection of this type between man and machine has appeared in a few psychological testing methods, and in certain systems consisting of a human operator and a machine suitable to realize some kind of adaptation. We outline the application of this principle by one of the most frequently used psychophysiological investigations, the reaction-time (*RT*) measurements. One of the peculiarities to be stressed here is: since the person is given a light stimulus and his response is only to press a button, the man-machine conversation occurs through a very *narrow* information channel. In spite of this we are trying to make use out of this connection as much as possible.

Here we are aimed essentially at giving a review of experiences gained in the first phase of the realization of such a cognitive system. Dealing with systems of this type experimentally a twofold benefit is expected:

1. It accelerates the process of recognition of certain biological (in our case psychophysiological) processes.

2. It makes possible to study the partial laws of certain human cognitive processes themselves — due to the cybernetical approach applied — independently of the system just studied.

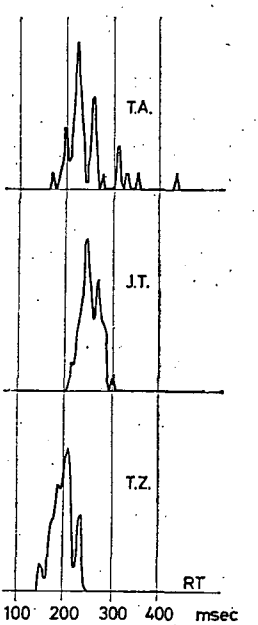

Fig. 1. Distribution of reaction-time values of three persons on the same stimulus structure

The system, to be investigated by applying the principle mentioned above was a kind of human behaviour, in connection of which we have had direct experimental results, performing psychophysiological research of numerous behavioural forms. We suggest to use the term *RT*-behaviour, and we are going to illustrate the practical application of this particular man-machine principle outlined above with the help of a cybernetical system suitable for analysing this behavioural form.

* * *

First of all we are going to explain what is to be called reaction-time behaviour, or more precisely what is the reason to speak of it as a physiological category.

The human optomotoric RT is generally regarded as a product of a simple delay unit, and the sequence of responses evoked by stimulus sequences is characterized by statistical parameters (mean, standard deviation etc.). In our experiments we tried to go further by applying non-traditional evaluation methods.

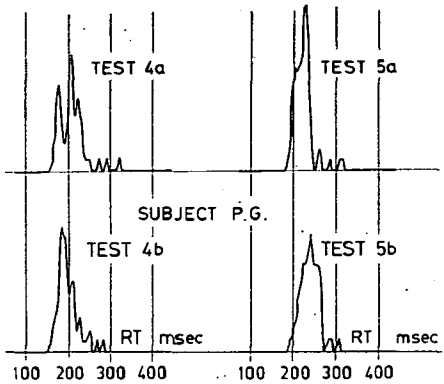

Fig. 2. Distribution of reaction-time values obtained from the same person on the same day. In the test versions *a* and *b* the interstimulus intervals were produced by a random generator of the same parameters.

The rhythmical stimulus structures applied previously — in spite of being easily produced — turned out to have some obvious misadvantages. Due to the recognition and habitude of rhythm, a phenomenon of rhythm-following appears, consequently one response cannot be considered to belong to the preceding stimulus any more.

This failure is eliminated by applying pseudo-random stimulus sequences presented by the experimenter himself. In the most up-to-date method at present the basic structure of stimuli is produced by the random generator of a computer. In certain cases, we modified the basic random structure by interposing short sequences of fixed intervals.

Various stimulus sequences consisting of 64 or 128 interstimulus intervals were available on punch tapes, and were fed into a special-purpose computer (NTA 512), which gave red light flash stimuli to the subject. (The light stimulus ceased only when the subject reacted with pushing a button. Hence, he could estimate his performance more or less precisely.) The reaction-time values were measured, stored and then displayed, or registrated in punch tape for further analysis.

A small but psychophysiologically well examined group was investigated for a long time using various stimulus structures.

Our aim in evaluating the results has been from the beginning to be able to classify the individuals according to their *psychophysiological types* and *actual states*. In connection with this it soon turned out that if only traditionally accepted statistical methods were applied, the types and states could be separated only by comparing with the results of other psychophysiological tests — if it was possible

at all. Therefore we examined the distribution of RT values in which significant various shapes were found indicating the individual differences (Fig. 1).

Naturally the statistical parameters (mean, deviation, greatest occurence etc.) can be illustrated by these histograms.

It can be seen, however, that the shapes of histograms greatly depend on the interval structure applied in the tests.

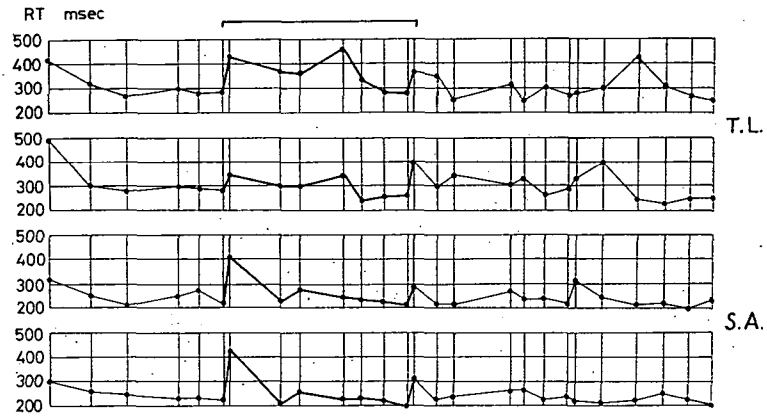

Fig. 3. Reaction-time sequences on the same test structure. The first and second curves were obtained from the same person in the morning and in the evening, respectively. The third and the fourth ones are from an other person and are paired similarly. In the marked phase the permanence of individual response patterns is well shown.

It was surprising to find strongly different histograms (Fig. 2) registrated from the same person on the same day, in response to two stimulus sequences of a random generator with *same* parameters. This phenomenon turned our attention to the fact that the shapes of histograms depends less on the statistical characteristics of stimulus structures, but more on the *context* of preceding stimuli. Naturally, it means at the same time that when examining the context-dependence of each RT value, it is not sufficient to take into account the dependence on one preceding interstimulus interval only, since the interval distribution of random stimulus structures with the same parameters can be considered to be equal (in the case of sufficiently great number of samples). Consequently in the two test versions the same response-distribution should be received.

We examined the dependence of RT values on the *one* preceding interval, and we found slight correlation in the case of very short (200—500 msec) intervals only.

After this there was only one way left to analyse the reaction-time sequences: *to analyse them as sequences*, i.e. time processes. During this examination a particular attention was paid to responses evoked by certain stimulus patterns consisting of 2—10 stimuli. These responses, because of their peculiarity, can be called as "response patterns". It was observed that the same persons on different days — occasionally in different physiological or psychical state — produced characteristic individual response patterns.

In the Fig. 3 the first and the second curves represent the response of the same

person on the same test structure in the morning and in the evening, respectively, the third and fourth curve show the similarly connected results of another person.

Analysing the patterns it turned out that their individual-determination is of greater degree than that of their distribution not reflecting time processes. An analysis on the basis of time processes — though it is more difficult — provides also more information.

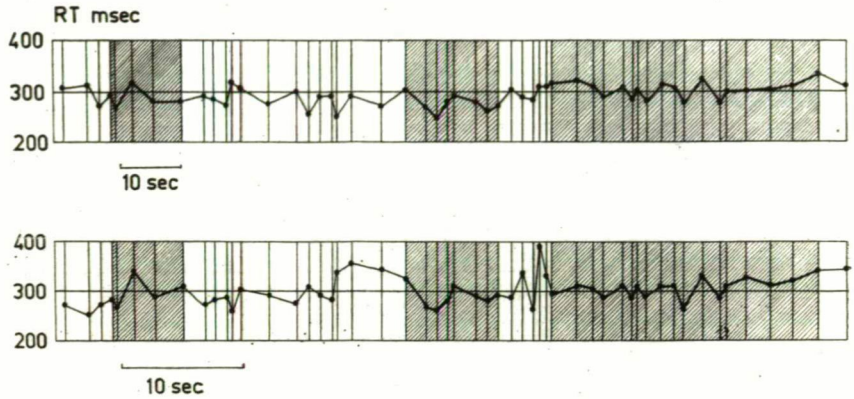

Fig. 4. Reaction-time sequences obtained from the same person. In the second case each interstimulus interval of the test structure was restricted to the half. Note the similarity of the RT sequences in the lined sections.

It was reasonable to try to give an explanation of the phenomenon in the following way: during the test the sequences of stimuli of the same intensity, or on the other hand the sequences of intervals between them constitute the environmental changes for the person examined. The person or more precisely his complex of mechanisms taking part in producing responses — while his state is being changed— will produce his response. The response can be conceived as context-dependent if the context is constituted by *the complex of stimuli and the preceding responses.* On the other hand, the response patterns can be regarded *behavioural patterns* to the environmental changes. It is reasonable to state that human beings show a particular *RT-behaviour* under these experimental conditions. Furthermore, we supposed that the RT-behaviour explained in this way is a component of the human individual behaviour as a whole, consequently it is characteristic for the structure of his nervous system.

This behavioural conception of RT-patterns being one of the starting points of our work, is strongly supported by our observation illustrated in Fig. 4.

The curves are from the same person, in the second case each interstimulus interval (ISI) was restricted to the half. (For an easier comparison the scales are different). In spite of this in the two curves we can find three sections with almost the same shape, amounting approximately 50% of the test. This finding supports, on the other hand, our previous statement that a RT-value cannot be explained only on the basis of the *one* previous ISI value.

A question arises: how is it possible to recognize the typological class and

actual state of the person examined from the RT-behavioural investigation. To determine the behavioural signs mentioned above a great number of stimulus patterns should be employed.

The application of certain stimulus patterns to a given person can be fruitful if the person reacts to them in a sensitive way, but at the same time an other person — just because of his different reaction type — eventually provides non-informative,

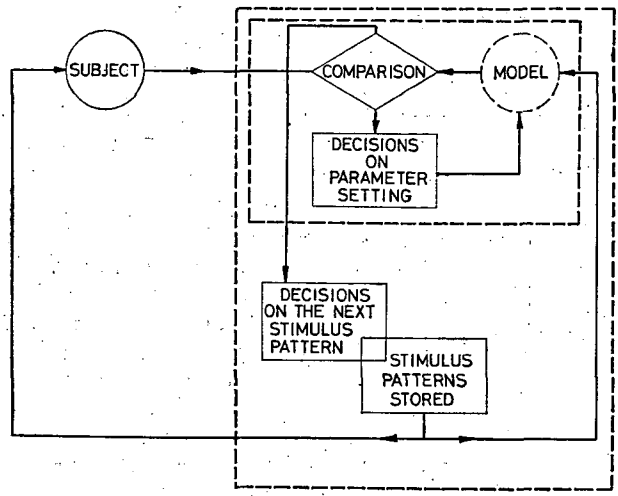

Fig. 5. Simplified block diagram of a man-machine system for psychophysiological testing. For explanation see text.

non-characteristic responses only. It is easy to see that during further examinations with two persons, stimulus patterns of different types should be used. Hence, in the cognition of RT-behaviour — like every other cognitive process — we should pass through the stages of *preliminary hypothesis — experiments — modified hypothesis — control, etc.* periodically repeated. This process takes a lot of time, the cognitive subject's judgement and decision is slow (in comparison with the speed of information concerning behavioural signs) and finally, the duration of an experiment is limited.

These considerations constitute the basis of the experimental setup in which a particular man-machine relation mentioned in the introduction can be realized. One of the peculiarities of this system is — as it performs the mechanization of a phase of the human cognitive process — to include by all means a functional model of the object examined.

* * *

Fig. 5 illustrates a schematic, simplified block diagram of the system consisting of a person investigated and a computer. The major functions of the subsystem to be realized by the computer are:

1. Comparison of response patterns of the subject and of the functional model to the same stimulus pattern, on the basis of similarity criteria.

2. Suppose that after examining the possible model variations i.e. possible parameter-constellations the similarity criterion is satisfied by several sets of parameter values. Therefore, they are to be stored representing diagnostical alternatives at the present stage of investigation.

Then, the corresponding parameters of the same value in the different sets satisfying the similarity criteria must be searched in order to leave them out of consideration i.e. to fix their value during the further investigation. Hence, the necessary number of model versions to be examined at the next stimulus patterns can be reduced step by step.

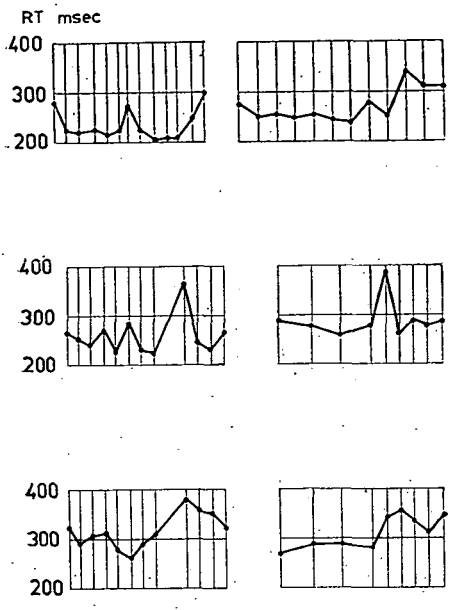

Fig. 6. Behavioural patterns in reaction-time sequences. 1) slow fluctuations in RT performance indicating energetic processes, 2) "surprise effect": a mistake followed by successful corrections presumably owing to high motivation and good energetic economy, 3) "surprise effect" followed by slow corrections.

3. After establishing the "proper" sets of parameter values the next operation is to choose one out of the stimulus patterns stored, depending on the previous responses i.e. on the model parameters of *greatest uncertainty*. This requires to classify patterns from the point of view of several parameters to be determined. Consequently, an extensive, off-line model investigation is to be done in order to compile a parameter-oriented set of stimulus patterns.

The steps sketched above form a cycle of cognition, which is likely the same in some human cognitive processes. Each cycle produces sets of parameter values, in other terms, they are points in an n-dimensional space of parameters. It is expected that after some cycle these points will be placed in the space of parameters within a domain well representing the person's psychophysiological type and actual state. Also, if the existing psychological and physiological categories for types and states can be expressed by the language of model parameters i.e. boundaries can be formed in the parameter space, then, using suitable classification methods the diagnosis will be made.

According to this concept, the realization of such a system started with constructing the model of human RT-behaviour. The model experiments — accompanied by further human experiments, of course — are expected to suggest principles of formulization of the cognition strategy, of similarity criteria, and of selecting stimulus patterns into an appropriate parameter-oriented pattern system.

* * *

The major part of physiological mechanisms integrated in the human RT-behaviour is supposed to be of neuronal (nervous) character. A minor contribution is thought to be added by biochemical, hormonal, biophysical (muscular)

factors. In spite of this, a neuronal-network type model based upon formal neurons, or upon their recently developed, more sophisticated forms, behaving like randomly connected, moving-threshold elements — could not be applied. This was because — as usually occurs in biological systems — neither the number of the elementary functional elements, nor the exact form and nature of relations connecting them, as well as the transfer functions realized by the subsystems were correctly known. What remained was the way of constructing working hypotheses about the most probable physiological and psychological mechanisms that interact according to general psychophysiological laws more or less proved, thus forming a unique system of the human behavioural mode to be studied.

In Fig. 6 several physiological examples of behavioural patterns are shown in attempt to illustrate the way of our thinking we followed in constructing functional analogies of the rules presumably manifested.

The first response patterns on the figure was a very common finding in numerous tests produced by almost all persons if a randomly distributed but nearly isorhythmical stimulation is presented. The wave-like shape of the contour line suggested the existence of at least two basic mechanisms to be considered: a more or less correct *estimation* of the distribution of interstimulus intervals and the *energy-dependence* of some psychological activities not correctly known at yet.

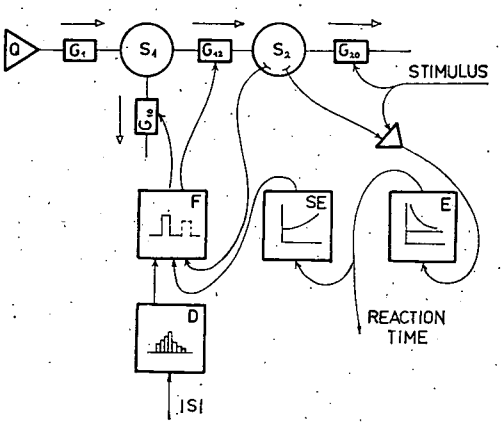

Fig. 7. Block diagram of the model of RT-behaviour. For explanation see text.

The second pattern is a particular type of response, occuring with a considerable variability, but preserving its general character in all the tests, where a firmly settled signal distribution is suddenly followed by an interval surprisingly different from the preceding ones. Considering its peculiar shape, the idea about an estimator that functions according to some kind of probabilistic logic, seems to be more validated, and further on, this estimating-mechanism presents itself to be governed by a *correcting* mechanism too, which must be activated whenever the person deviates from a mean performance, i.e. an *error detecting* function can also be supposed. If the executive part of the system, (the energy-dependent subsystems) is well fitted to the task and the estimators are working properly too, then a very successful corrective step will be performed, as seen in the 2. patterns. But if the person is more tired, or the economy of the energetic chain is disturbed by other reasons, the correcting step induced by a mistake cannot be so successful. (See patterns No. 3.) A further assumption emerging from the facts and relations mentioned above was the following: the psychophysiological activity requiring some kind of energy would be the *focusing of attention* to a specific task, or in neurophysiological terms the concentrating of excitation upon a central nervous system area surrounded by some specific sort of lateral inhibition. This area seems more likely to be the sensorimotor cerebral cortex; thus the specific excitation-inhibition

pattern impinged upon it will determine the synchronity of nervous cell's firing, which, in turn results in the absolute value of the reaction-time. This latter assumption is strenghtened also by introspective observations, according to which voluntary focusing of the attention to one's *manual activity* rather than to that of stimulus perceiving — considerably shortens the reaction time. It is also an introspection that plays a considerable role in detecting such *feedbacks* which for example, are activated by a sudden or even continuous involuntary fall of the *specific* attention level.

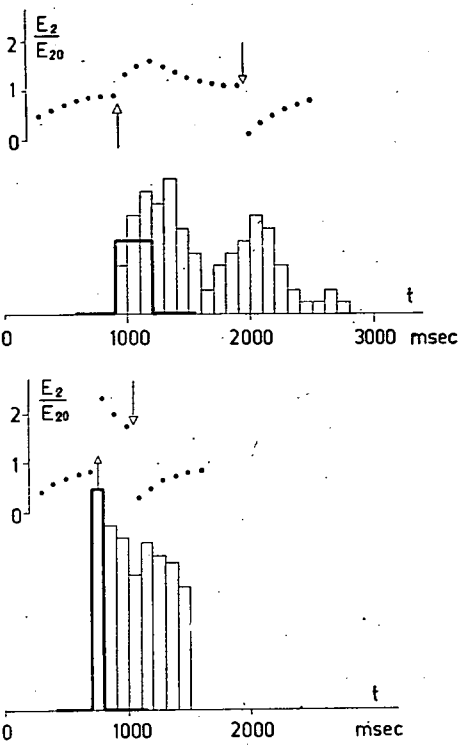

Fig. 8. Generation of "focusing" impulses based on wide (above) and narrow (below) distribution of the preceding interstimulus intervals. Thick lines indicate the position, width and height of the focusing impulse. Dotted lines represent the specific attention level (E_2) related to its resting (unfocused) value (E_{20}). Due to the focusing, E_2 suddenly rises, and then falls almost to zero, when the stimulus appears.

After having analysed a great variety of such RT-patterns it became clear that all mechanisms supposed to take part in generating them, can hardly be kept in mind at the same time. Therefore, we tried to describe these mechanisms, i.e. to formulise a model of the mechanisms supposingly involved in RT-pattern generation. Every mechanism seemed to have two aspects of its functioning: one is the information processing, the other is the operating as an energetical system of finite capacity. As a matter of fact, these functions are closely connected to each other, still, we separated the whole model of mechanisms into two parts. A subsystem of energy distribution is responsible for maintaining an energy level necessary to perform the task. This subsystem is controlled by the other part of the model in many ways.

Obviously, the present model does not include all considerations we make when trying to explain a pattern qualitatively. Its task is now only to decide whether this concept of energy distribution and information processing is sufficient or not for presenting basic properties of human RT-behaviour.

Fig. 7 shows the simplified block diagram of the model. The energy distribution system consisting of the energy source Q, energy stores S_1 and S_2, energy channels G_1, G_{10}, G_{12}, G_{20} can well be illustrated by electrical or hydrodynamical analogies. The energy level of S_1 corresponds to an unspecific attention level supplied by Q and consumed through the channels G_{10} and G_{12}. S_2 stores *specific* energy of attention, its content is then regarded to be available for actions in RT-tests, and is consumed through the channel G_{20} almost entirely when a response occurs. The increase of the transfer capacity of G_{12} (which is occasionally accompanied by the simultaneous

restriction of the channel G_{10} representing the *aspecific* drainage of energy) causes a transient increase of the specific attention level in S_2. This process occurs when a "focusing" instruction comes from the other part of the model.

This instruction delivered by the focusing unit F can be produced in two different ways: *1.* on the basis of the distribution of the preceding interstimulus intervals stored in D, and *2.* by observing that the level of S_2 falls to a certain threshold. This feedback-evoked focusing occurs when the stimulus did not arrive by the time expected on the basis of the ISI distribution and the level of specific attention slowly decreases.

In the focusing process the subjectively estimated value of the preceding response plays a very important role. Here the motivation level of the subject examined is thoroughly involved. The "self-estimation" SE realizes in the present model yet a simple relationship between the preceding response and the weight factor produced for the focusing impulses.

The characteristic of the "executive" unit E shows our assumption concerning an inverse relationship between the reaction-time value and the actual specific attention level.

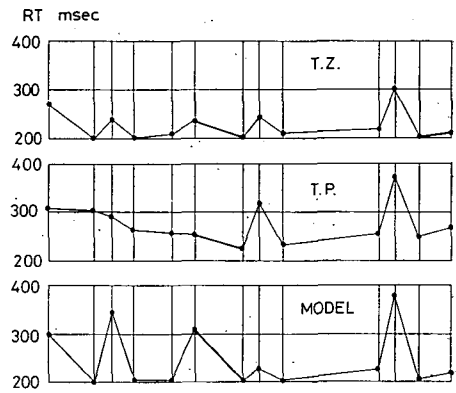

Fig. 9. Reaction time sequences obtained from two persons and from the model on the same stimulus structure.

As an example, Fig. 8 shows qualitatively how focusing impulses are generated and furthermore, how the specific attention level is affected by the distribution of preceding ISI values. Nevertheless, it is to be stressed here that the weight factor for a newcoming ISI depends not only on its relation to the former distribution but also on the fact that shorter ISI values are naturally of greater importance than the longer ones (namely in case of unexpected long intervals one can rely upon feedback mechanisms maintaining a high, but fluctuating attention level for a reasonable time. Naturally, since the most intellectual part of the model is concerned here, beside the processes learning and forgetting, habituation and dehabituation, certain emotional weighting functions indicated probably by the reaction-times themselves should be taken into consideration.

In this initial stage of work we were seeking for those approximative ranges of model parameters within which the RT-behaviour of some individuals in our experimental group could be simulated. Other than stated a good similarity between model-generated and human patterns, the analysis of time processes in the corresponding patterns supported our basic hypothesis according to which there are two characteristic functions closely coordinated in reaction-time generation: an activity like *information processing* and its *energetic conditions.*

In Fig. 9 the upper two curves were registrated from two different persons, while the third is a model-generated pattern. In the initial phase a considerable difference can be seen between each pattern, showing the better energetic economy and estimating functions of the first subject. In the second phase, however, the courses of the reaction-time values are of uniform character.

Fig. 10 illustrates response patterns of two persons and of the corresponding model versions, to an other stimulus pattern. Both versions have estimating and focusing parameters of same values, and in the second case the energy levels are only slightly higher than that in the first one. In the second case, in turn, there were higher focusing impulses produced, as a result of a more intensive self-estimation i.e. of a greater effort to shorten the reaction-time after a long one. Consequently, the only difference between the two person's RT-behaviour was — according to the model experiments — the higher motivation level of the second person.

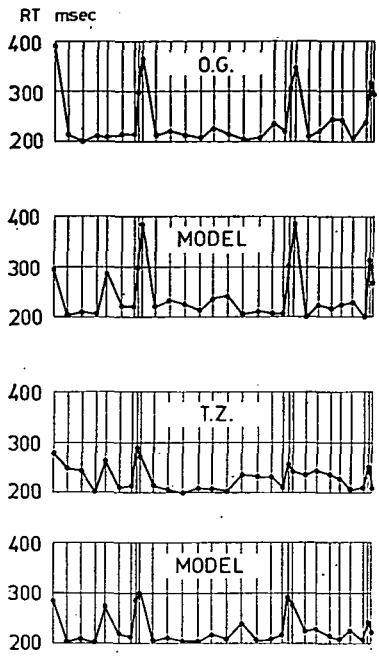

Fig. 10. Reaction-time sequences from two persons and from their model versions.

According to our final task the model of RT-behaviour, being a subsystem of the complex cognitive system, needs a further development in a rather unusual direction. It is due to the fact that among the subjects of our experimental group there was a separated RT-behavioural class of the persons whose patterns could be simulated by the present structure with best fidelity.

This class consisted of two persons, a professional pilot and a racing driver.

The conclusion is that in developing the model an energetic chain of *worse economy* governed by a *less precise information system* probably of stochastic nature as well, is required.

Принципы человеко—машиной системы применительно к тестам поведения

Авторами предлагается методика проведения психофизиологических процессов обследования, обеспечивающая минимальную потерю информации. Для увеличения эффективности обследований составляется индивидуально для каждого пациента структура раздражений в зависимости от его предыдущих ответов. При этом требуется высокая скорость обработки информации о реакции обследуемого пациента и, следовательно, включение в разрабатываемую систему быстродействующей ЦВМ.

Возможность и целесообразность реализации такой системы иллюстрируется на примере исследования оптико — моторного времени реакции. На первом этапе необходимо разработать модель реакции поведения. В рассматриваемой человеко — машинной системе человек является объектом управления (познания), а управляющим звеном — вычислительная машина, определяющая виды раздражений и оценивающая ответы.

Проведенные экспериментальные исследования показали возможность предложенного подхода и адекватность полученных результатов сформулированной гипотезе.

LABORATORY OF CYBERNETICS
JÓZSEF ATTILA UNIVERSITY
SZEGED, HUNGARY

(Received February 14, 1973)

Human fatigue as a phenomenon modulated by environmental and typological factors: an approach based upon a complex test structure

By I. MADARÁSZ and P. HUNYA

Between the numerous psychophysiological testing methods currently used in ergonomical, psychological laboratories for detecting fatigue there seems to be an apparent lack of coherence. This is due to the fact that each testing method has been developed on purely empirical grounds because of the unsatisfactory degree of general validity of the theories concerning fatigue-dynamics. The early concepts about the nature of fatigue are founded on contemporary physiological theories of simple *muscular* tiring. These concepts are characterized by making use of this rather primitive, mechanistic analogy in elucidating a phenomenon much more complicated than simple muscular contraction. One cannot escape the suspicion that the main cause of the survival of such a "per analogiam" theory, in spite of numerous experimental facts indicating its inaccuracy, has to be found in the essentially conservative nature of the so called "common", as well as "scientific" sense. The main subject of this paper is the outlining of a more coherent picture of the different mechanisms (physiological and psychological as well) which presumably take part in the fatigue in general, and later on the outlining of a method of investigation and evaluation which has been used by us in our attempt to measure this phenomenon more objectively.

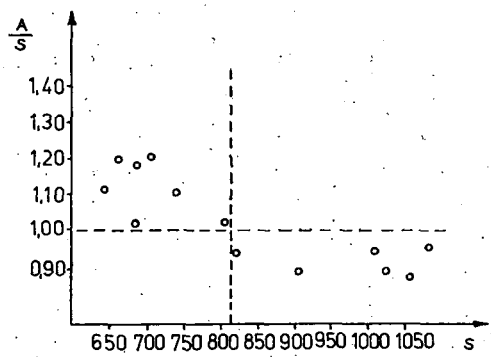

Fig. 1. Changing of control-goodness measured in tremor tests (aiming and static) caused by equal work-load in a group of car drivers. Abscissa: control-goodness at the start of the work-loading; ordinate: arrival start control-goodness ratio.

In our opinion the phenomenon called fatigue is in its essence a psychophysiological *reaction* and therefore cannot be characterized by such purely quantitative parameters as the height of muscular contraction or the number of errors made during a simple test of routine psychological practice; e.g. the crossing out of all vocals in a printed text, etc. In other words: the fatigue, being essentially a special human *behavioral manifestagion* is *not* necessarily equal or even proportional

to any of the output quantities measurable *separatedly* in common laboratory testing procedures.

What is the main consideration in saying that the fatigue cannot be a simple decreasing in a stretching force as that of a spring moving a clockwork? This is due to the fact that man behaves in *real* work-situations like an adaptive automaton, i.e. he changes continuously the *quality* of his adaptive reactions, in other words, the characteristics of the transfer functions of his respective controlling subsystems. The terms both "adaption" and "real work-situation" need to be clarified.

Speaking about the adaptive side of the fatigue as complex behavioral reaction means, that we are bearing in mind the essentially *purposeful* nature of any human working activity. Purposefulness is meant as an objectfunction which has to be optimized. During the performing of a work either of physical, or of intellectual type, *fatigue-causing factors* are acting upon the organism as input quantities, which may take their origin either from the outer, of from the internal environment. Because of the homeostatic nature of the biological systems, *adaptation* tends to eliminate the "disturbing" fatigue-causing inputs, and it is easy to see that this may occur either by *habituation* (decreasing the sensitivity of the input peripheric organs), or by "*active*" *counteraction* against the fatigue-causing inputs, i.e. a) decreasing the rate of work, b) ceasing with working for a while, if it is possible under the circumstances given.

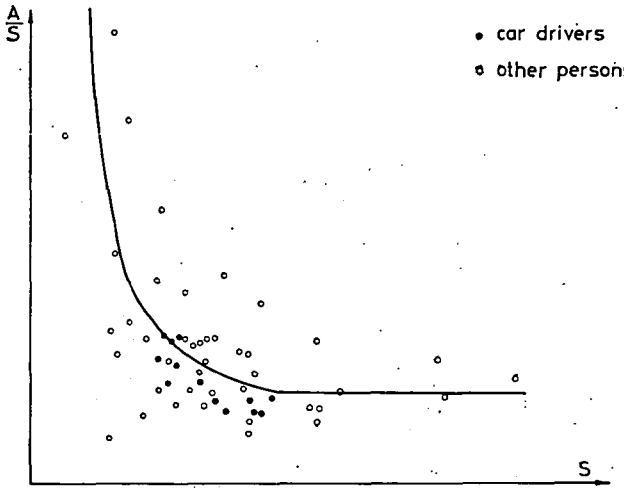

Fig. 2. Changing of control-goodness in tremor tests (aiming and static) in a mixed group.

What *real* work-situation means will be different according to the type of work to be done. In common clerical work, for instance, activity can be suspended for shorter or longer intervals allowing thus the reequilibration of the disturbed balance between fatigue-causing factors and adaptive mechanisms. But if the elimination of all the fatigue-causing factors should be incomplete during a long period of work, the subjective feeling of tiredness and a net *loss in general adaptation capacity* will result. This statement is supported — as to the adaptation capacity loss — by physiological experiments on laboratory animals conducted by us, where some neuro-hormonal adaptative mechanisms were found decreased after a difficult-task and highly motivated learning session.

The degree of difficulty of a car driver's work will again depend in many aspects on the nature of the task to be fulfilled, but it remains more of psychological than that of muscular character. The car driver can most properly be considered as a human operator in a man-machine system. The practical consequences of this view are rather far-reaching if we consider that the quality of performance of a given

man-machine system depends in most cases on the *controlling* characteristics of the human operator's control system. Turning again to the point of the so called real work situation it can be established that the peculiarity of a car driver's work is in the fact that *he has to keep on controlling* his vehicle independently of the actual state of his energy-balance; he has continuously to mobilize spare energies from *other control systems* in order to fulfill the obligatory task, i.e. the keeping of the vehicle on road according to the rules of the correct driving behavior.

The experimental work, the first results of which we are going to report, has been based on a hypothesis according to that at the present level of technological civilization fatigue can be expressed, first of all, not by the decreasing of physical output of the work done, but much better by that of its qualitative indexes. In other words it means that *fatigue is expressed in the restriction of adaptation capacity* to the external and internal conditions, in the deterioration of the complex control function as a whole, which results in the decrease of work-efficiency after consuming the reorganization reserves at disposal.

It is supposed that the elements of the complex control system realizing human adaptation are subsystems whose output quantities can be measured as concrete physiological manifestations. In the case of experimental investigation of numerous and selected subsystems it is expected that we can get a measure about the goodness (correctness) of the control activity of the complex system too.

To find a minimally acceptable compromise with the requirements of theoretical considerations, methodological possibilities, as well as mathematical formulization — often contradicting to one another — we have chosen for experiments a set of subsystems, on the basis of our preliminary investigations. We supposed, that each of them represents a more or less independent control activity of physiological nature. They are the following:

1. The constancy of opto-motor reaction time and its dependence from the interstimulis interval,

2. the so called error-time, (test earlier published by us) which essentially reflects the estimation errors in the time and space coordinates of a moving target,

3. the physiological variant of the intention tremor called by us aiming tremor,

4. the so called static tremor reflecting the physiological postural lability of an arm in resting condition,

5. data of blood pressure measurements.

Each of the physiological mechanisms was the subject of investigation in a testing procedure, separetedly. The testing procedures were carried out on healthy adult persons.

One of the peculiarities of physiological measurements in humans is that the uncontrollable effects are greater in number than in animal experiments. The equipment for measuring the selected functions has been constructed therefore with the aim to reach the greatest measuring accuracy possible. We have made it suitable to record and store automatically hundreds of data per person. The nucleus of the experimental setup consists of a multichannel analyser of 1024 words (16 bit each) capacity. By means of a few modifications in its original hardware it has been made capable of performing operations in a stored-program mode too, enabling thus the automatic control of the whole running of the tests, including the storage and partial processing of the data received from the experimental subjects, as well as the punching of the results on tape for further processing.

The series of light flashes in the reaction-time behavior test characterized by a structured, pseudo-random distribution of the interstimulus-intervals was produced by the central control unit of the analyser. The appropriate programs realizing this stimulus-structure were written with the purpose of producing a modelled sequence of the different, adaptation-evoking input patterns acting normally upon the driver under real working conditions. It has been the stored-program operating mode that made it possible too, applying differently shaped generative functions for the moving-target test. The adaptive capacity required from the subject was tested by changing the speed, as well as the contours outlined by the moving lightspot on the screen. The functioning of the measuring device used in the detection of the so called aiming and static termor was governed also by a central control unit. The design and operating principle of the apparatus was published earlier.

As results of the tests we have received essentially functions that expressed the deviation of the examined "elementary" control activities from the ideal ones. We have not examined the whole course of these functions; for the moment we have watched only 3 control-theoretical characteristics, namely

1. the mean value of deviation from the ideal control,
2. the mean value of squared deviation,
3. the maximal deviation.

The results of individual tests are expressed by triplet numbers. These numbers define, as coordinates, a point in an Euclideal space. The distance of this point from the origo will be a measure of the goodness of the control in a given test. As the triplet numbers of all the tests are essentially *general distances* in themselves, it is justified to consider all the components of different tests to be the coordinates of a 3. N. dimensional space whose certain three dimensional subspaces characterize a test by each. N equals the number of the tests applied. If necessary, each arbitrarily selected subspace can be drawn together. A single generalized distance will then result measured from the origo, without losing significant information, because we have kept the distances as quantities characteristic for the goodness of control activity.

After choosing the experimental equipment and mathematical system of the evaluation of data, our aim was to investigate the control characteristics received from the tests according to their sensitivity. We tried to determine experimentally those subspaces in which the changes of work-productivity — i.e. the control-goodness deterioration supposed by us — appear in proportion with the degree of fatigue.

Our first observations were done with professional drivers exposed to measurably equal work loads. The theory and praxis of the objective measurement of loading factors in car-driving are explained in detail in the paper of D. Muszka.

With this group our aim was set to find experimentally a *measure* characteristic first of all to the degree of fatigue. As there was no possibility to prove that the individual tests applied represent independent control activities, we chose heuristically subspaces supposedly of great informative value and the generalized distances obtained in them were analysed. It has been found that certain phenomena can well be illustrated in a coordinate system with axes as starting values, respectively quotients of arrival/start. It will be perhaps of interest to note that the subsystem choosen was that of the different tremor phenomena (aiming and static) (Fig. 1).

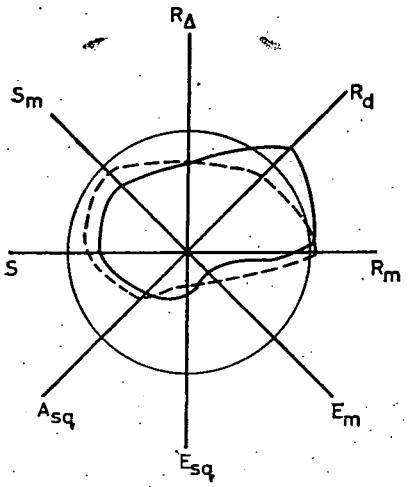

Fig. 3

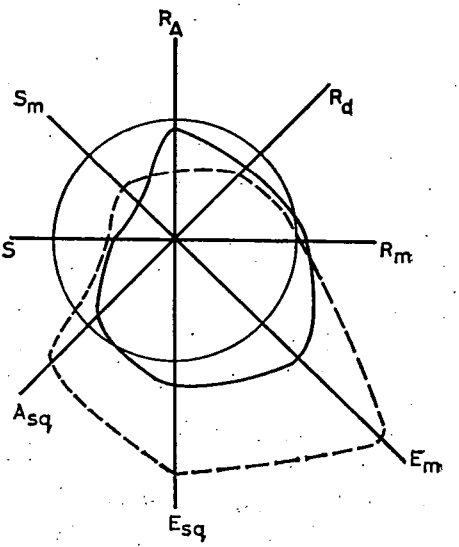

Fig. 4

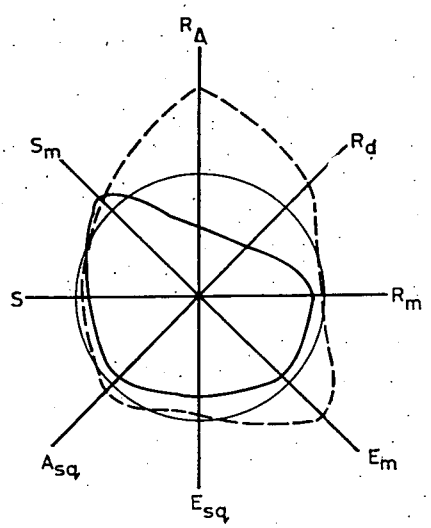

Fig. 5

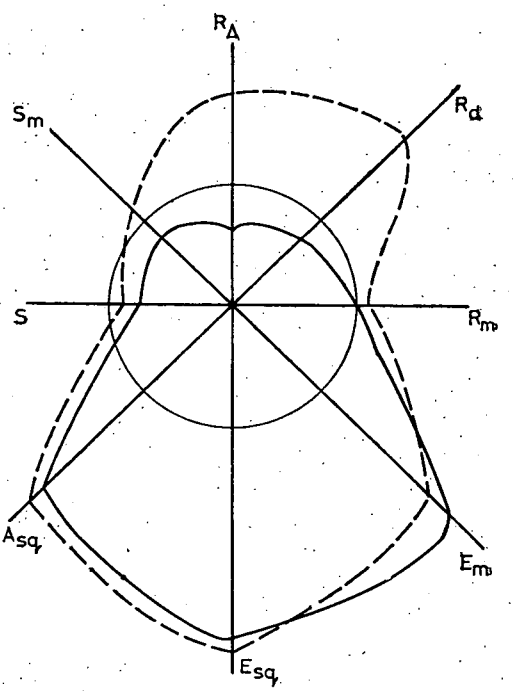

Fig. 6

Two-dimensional illustration of the individual control-efficiency change in the multi-dimensional space. Explanation in text.

It was found that the characteristics of persons starting with good levels of control-capacity (left upper compartment) showed deterioration to the work-load, while others starting with moderate, or even great deviations from the ideal were improving under the same amount of load (right bottom compartment). The conclusion drawn from the data was that the mechanistic theory of the fatigue considering it as a simple decreasing of energy at the output side — is obviously untenable.

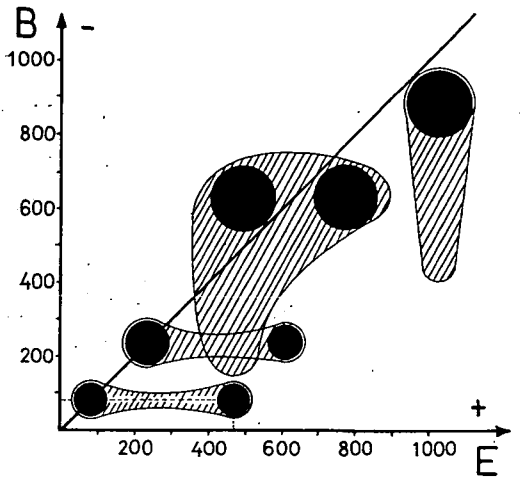

Fig. 7

In the following period a greater and more heterogenous group of persons was investigated in multiphasic testing experiments. The quantitative value of work-load was not measured in this group, but it could be well apperoximated saying that it had the usual measure of their normal workday. Fig. 2 shows the results obtained.

Investigating the same subspace as before, we could state that professional drivers fit well into the whole distribution forming a relatively stable and uniform central part of it.

To examine and illustrate — rather qualitatively than quantitatively some principally important features of the individual control-efficiency change under work-load, the figures 3. 4. 5. 6. can be used. Certain characteristics of each test are mapped as distances along the 8 axes in the figures namely, R_m mean value of reaction time; R_d standard deviation of reaction time; R_A variancy range of reaction time; S_m mean deviation of static tremor; S generalized distance in the static tremor's subspace; A_{sq} mean squared deviation of aiming tremor; E_m mean deviation of error-time; E_{sq} mean squared deviation of error-time.

The solid line represents the initial state, the broken one the state of arrival, while the circle stands for the mean values of the initial state for the whole group.

These specially-shaped, spherical patterns reflect properties of controlling activities in different aspects of a single individual. The following features are to be stressed:

1. The persons as individuals produce particular, similar patterns by repetition.

2. Under the effect of work-load almost every person shows a reaction of particular dynamics as reflected in the independent novement of each element characteristic for his pattern.

3. Numerical measures can be derived, proportional to the area of the spherical pattern, but these give seemingly less information than the pattern as a whole.

From the very beginning our working-hypothesis was that particularities of the dynamics of human fatigue can be understood only on the basis of the cognition of the persons' typological characteristics, i.e. of the cognition of their inherited central nervous mechanisms. That is why we strived to classificate our experimental

persons under work-load into classes. Theoretically it can be done if we try to consider in mind the characteristics of all the tests at the same time. The efficiency of this method, however, is rather doubtful because the data-processing capacity of the human observer's mind is limited and because of the danger of subjectivity. Our investigations aimed therefore at constructing an *automatic classification system* — described in the paper of P. HUNYA — which made it possible to apply automatized classification procedure in this field. The results received as outputs of this classification system are illustrated in Fig. 7.

The perpendicular axe corresponds to the starting overall control-performance, the horizontal one to the value at arrival. The areas bordered by continuous contour-lines mark the limits of groups (classes) pointed by the automatic classification system, their darker nuclei mark the greater frequency of individual cases and their lighter parts the less frequent occurences. We were going to make it clear by this type of illustration that the independent existende of one group (type) is determined, first of all by the position of nuclei, so it may occur that peripherical parts of recognized classes may differ in some extent of overlapping.

Комплексное исследование влияния внешних и внутренних факторов на процесс усталости человека

В работе рассматривается вопрос описания нейрокибернетического механизма процесса усталости. За оценку усталости принимается степень изменения качества функционирования сложных адаптивных механизмов человека, в частности, снижение психомоторных регулирующих действий организма.

На основе сформулированной гипотезы авторы разработали комплексную психофизиологическую испытательную систему.

Полученные экспериментальные данные принимаются за координаты многомерного пространства. Вводятся расстояния, пропорциональные мере усталости испытуемого лица при воздействии эталонной нагрузки.

На основе анализа динамики изменения степени усталости авторами рассматривается возможность классификации исследуемой группы людей (принцип предложен П. Хуня) с использованием самообучающейся автоматической системы.

LABORATORY OF CYBERNETICS
JÓZSEF ATTILA UNIVERSITY
SZEGED, HUNGARY

(Received February 14, 1973)

Reduction of traffic risks applying cybernetic methods

By D. MUSZKA

The immense progress of the automobilism, in addition to conventional problems such as motor vehicle design, road construction, traffic control etc., poses a series of new questions to the traffic science. Among these the oppressive problem of road accidents can be named as one of the most important ones. The transport is a social phenomenon to-day, its risks are imminent for any member of the society. Thus, it is not surprising that the need of studying and introducing new effective methods to prevent road accidents presents itself on a social level in our days. The mortality of road accidents hits such a level to-day and the future predicted by estimations based upon extrapolation is so dark that the experts are becoming more and more aware of the fact that the efficiency of the conventional methods in preventing road accidents is unsatisfactory. The appearence and the spread of a new science, the cybernetics have yielded the objective basis for this recognition. A lot of results of cybernetical research, and primarily the computers have already found widespread applications in the solution of conventional problems of automobilism (motor vehicle design, road construction, traffic control etc.). This is not the case, however, regarding the new problem just said above. Research aimed at the prevention of road accidents and based upon cybernetical methods is carried on only at a few places isolatedly from one another. The results of the research usually do not get published. In spite of all these, we are firmly convinced that the defending reflex of the society will soon make this work more intensive, more efficient and better organized in order to bring down the road transport risks to a tolerable level.

One of the aspects of problems of road traffic studied from the viewpoint of cybernetics consists of the fact that in its focus the running man, the driver of the motor vehicle stands as the most important subsystem of the complicated man-machine system. In view of the operating reliability with respect to the road traffic, this system is of essentially smaller output that the other subsystems of the whole and so, like other man-machine systems, it represents the operating reliability of the whole system. The major part of the road accidents is caused by the subjective mistakes of the driver and only a few of them by the failures due to the vehicles and roads. The transport risks can be interpreted as the probability of erroneous functioning of the optimal algorithm belonging to the man-machine-road system. The research aiming at reducing the value of this probability has to take the following factors into consideration:

1. The possibility of constructing high ways for the overland transport is limited.

2. The number of vehicles taking part in the traffic is not limited.

3. The capacity of the driver functioning in the traffic on the basis of a right algorithm is limited.

These viewpoints are strongly connected, but it is the third of them that promises results for the most part.

We have formulated two questions as the basis and starting point of our research. The first one concerned the possibility of increasing the capacity of the driver by mechanization of certain activities of the driving process, the second one the possibility of a dynamic measurement of the capacity i.e. the change of capacity of the driver. We are aware of the great importance of the first question and we do not consider the research on automata and automaton system acting in the driver's capacity at certain points of the driving process as of second rank. However, we are going to study the second question here in detail.

One of the consequences of the enormous progress of the automobilism is the fact that the number of non-professional drivers is rapidly increasing. Consequently, among the drivers there appear people of most various types whose responses to the stress due to driving a motor vehicle get going on a very wide scale. Furthermore, howsoever rapid is the improvement in the road conditions, the increase of the number of vehicles and that of the traffic speed result a traffic density of such a measure that, together with the decrease of the time intervals disposable to certain actions, makes demands on the fatigue tolerance of the driver to a greater extent and in a more sensitive way. Thus, it seems to be important to develop such an apparatus (both theoretical methods and technical installation) with the aid of which different specific factors involved in the fatigue of the driver could be measured and evaluated.

Some questions of the psyhophysiology of fatigue and questions of measurement of specific factors causing fatigue

First of all, we have to point out that the definition of the fatigue as condition comes up against a difficulty. According to present modern conception, the fatigue is primarily a psychic phenomenon. Certain authors mind that it is nothing else but a subjective experience of the fatigue which is not necessarily accompanied by physiological symptoms referring to a decrease of output. This should mean that it is not possible to characterize the tiredness as conditioned by any of its symptoms. The most known and significant fatigue symptoms such as lengthening of the opto-motoric reaction time, regression observable in a test of a longlasting manual skill, the increase of the amplitude and frequency of the vegetative tremor neither individually nor together give a good agreement with the subjective feeling of fatigue and do not yield satisfactory correlation with the objective output.

Without making an effort to give a generalized definition of the fatigue, in accordance with the majority opinion of the experimental psychology, we will consider the fatigue as an internal change of condition being reflected in the mind, the consequences of which affecting the work are determined by the general psychosomatic condition of the whole organism (physiological fitness, motivation level)

and the dynamical equilibrium of factors causing fatigue during the work. In this sense, the fatigue is a kind of self-adaptation of the human being which is aimed at reestablishing the inner equilibrium formerly overbalanced by factors being cause of the fatigue. On the one side of this fatigue-accomodation complex, one can find the decrease of the just beginning fatigue feeling performed by means of an automatic increase of the motivation level, on the other side, in turn, the work interrupted by an unbearable feeling of fatigue stands.

We ask now the following question: can the fatigue be measured? On the present level of physiological experiences, there is a great probability of the fact that the inner changes of conditions happening in the central nervous system and in the other parts of the body, which we know in fact quite imperfectly, cannot be measured at all or they can be measured only by means of their non-significant projections. The subjective feeling of fatigue, though objectively not measurable, can be however studied. More precisely, one can study the measure of it given by people's self-estimation which, under appropriate experimental conditions, linearly increases with the time devoted to the work in question.

The consequences of the fatigue as condition affecting the work have been attempted to track down in many ways. Based upon the physiologically well-known process of the simple muscle tiring, the designers of the measurements had been starting above all with the assumption that, according to the analogy of the tiring muscle, the quantity of the work done would decrease. But, this view, as it turned out after many decades, did not proved to be fruitful. It did call, however, the attention to the decisive role of the factors acting against the fatigue, since it became plausible that one of the most significant properties of the tiring process was the compensation. It is obvious that at the final phase of the tiring process, even in case of an inexplicit activity of muscle work character, the output can be decreased. It would be, however a mistake to identify the measure of this decrease with that of the fatigue by overemphasizing the quantity side of the output. In fact, it can happen, and not only in separate cases, that the output can increase in the state of an ultimate effort. From the comprehension of the fatigue as a complex accomodation reaction just explained, it follows that research methods of measuring the fatigue are to follow not only the quantitative but the qualitative aspects of the work, too. This requirement means, of course, that always a concrete work situation is able to answer the question asked about the qualitative aspect of the work and to tell the method of studying them under an acceptable objectivity. In the literature of experimental psychology studies of this kind are very rare, a fact probably due to the uncleared notion of the fatigue and difficulties appearing in the measuring technique.

In connection with the research aimed at an objective approach of the fatigue there is an other aspect of the problem which can be also measured. This is the reveal of the quantitative terms of the fatigue-causing factors during the work and the experimental determination of their correlation with the change of output and subjective feeling of fatigue. In this work we try to approach the problem from this direction.

The driving of a motor vehicle is an occupation of sitting-work character mostly with psychic demands in which, from the viewpoint of the fatigue, the sui generis muscle work has only a very subordinate role. Since in this activity a higher degree working of the nervous system is prevalent, it would be difficult to measure the quantitative side of the work simply by the output. Still, it can be seen that its

usual approximating estimate is represented by the formula: average speed × time (if accident-free driving is supposed). The description of the actual situation given by these factors is, however, very inaccurate. The following gives a good account of it. In virtue of our formula, the output of the driver while running a route of low traffic at an average speed and the output of the driver on a route of heavy traffic have to be considered as equal, a fact which is, of course, not true. It is apparent from the example that the special aspects of the work done by the driver are not at all or just from very far away reflected by the so-called "output characteristics" having their roots in the muscle work analogies.

Does it really mean that these quantitative characteristics as those characteristics of the driver's output which play a role in the reasoning well approximating the reality should be turned away? In our opinion, this is not the case. We will arrive to measuring quantitative characteristics much more exact than those taken into consideration up to now, the data of which, under suitable transformations, will lead to values being in a good accord with the experience.

We repeat again that, when characterizing the work of the driver, not only the quantitative but the qualitative sides of the activity have to be studied. It is clear that driving a motor vehicle is a dynamic control-like activity in the characterization of which the qualitative sides cannot be left out of consideration. The ultimate purpose is the objective evaluation of the "goodness" of this activity. We think that this can be carried out in the following way: the actual driving activity is compared with an activity ideal in every possible travelling situation (i.e., which satisfies all the rules and results in an optimal speed) and the difference obtained is determined. The deterioration of the activity caused by fatigue could be measured by this difference. (We have to point out, however, that by means of the present apparatus of the control theory the description of the activity of the ideal driving encounters immense difficulties. These difficulties can be prevented if it is succeeded in determining experimentally the characteristic features of a driving style close to the ideal one. Then, in fact, the has-to-be value of the control activity could be replaced by the style determined experimentally.)

The facts said above imply that to the approximately objective measurement of the fatigue caused by driving we can arrive in several steps:

1. We experimentally determine the qualitative characteristics of the algorithm of the ideal driving.

2. We examine the factors causing fatigue during the trip, the measurable proportion of them, as well as the level they have to accumulate up to the ensuing of the impairment of activity.

3. Concerning the persons examined, by using a psychological test model reflecting the aspect of the control activity of driving we determine qualitative parameters characteristic for the level of fatigue.

4. These parameters are used as correcting factors for the computation of the factors causing fatigue.

A possible cybernetical model of the problem

We are going to describe the main characteristics of the driving work by means of cybernetical notions in order to make the setting up of the model of the problem possible.

We start with the fact that, from the viewpoint of participating in the traffic the driver and the vehicle constitute one complex system, i.e., we do not consider this couple separately as directing and directed system. The central figure of this system is the human being who is in a mutual effect, in a mutual regulation with the vehicle driven by him. At the same time, there is also a mutual effect between the man-vehicle system and the outside world. The effecting elements of the outside world are the traffic, road and meteorological conditions etc. The effect on the outside world of the system is performed by the conscious and unconscious activity of the driver through the vehicle as an intermediating unit. (See Fig. 1.)

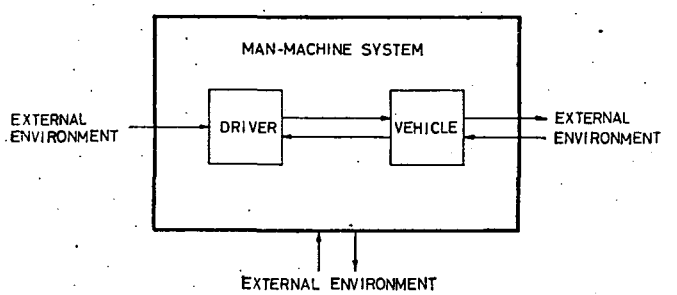

Fig. 1

The sequence of the moments, i.e., the behavior of the system can be characterized by certain parameters of time. These parameters are the elements of the effect on the outside world of the man-machine system. Their mutual effect reflects, in turn, the style of driving. In the cybernetical characterization of the system a very important element is the fact that the parameters of moments as physical quantities are of output character, and in the same time they are fed back through the driver's sense organs as information characteristics for the style of driving. (See Fig. 2.)

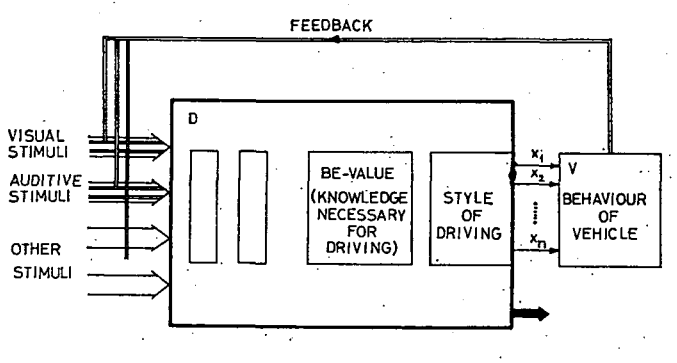

Fig. 2

In our opinion, the feedback-information (together with other informations coming on the driver) effects as a factor causing fatigue on the physique of the driver and it has a decisive role of both the instantaneous and long-lasting change.

We note that other factors causing fatigue of the driver such as the noise, the temperature of the cabin of the car, its humidity and ion-concentration, the driver's activity of other kind (conversation, other mental activity etc.) can be characterized in a typical travelling situation by means of a monotoneously increasing function. Thus, in our study aiming at the measurement of fatigue they can be considered as constants or weight-factors. The most visual information produced by the traffic and road conditions, such as the motion of the vehicle to be left behind, will appear yet in the feedback-information deduced from the parameters of moments, hence their actual values can be determined by not only an experimental estimation but by an actual measurement.

From these we can conclude that from the point of view of fatigue the examined persons can be considered in the driving activity as adaptive automata of black-box character which reestablish their inner equilibrium destroyed by factors causing fatigue in such a way that in favour of the stabilization of the output structure they alter the parameters of their subsystems.

Under a given level of fatigue (which is of course determined by specific peculiarities), the change mentioned above first appears often in an improvement of the efficiency degree (the output is approaching the ideal style of driving) which, later on, may happen at the cost of other adaptive functions of the organism and as a result of this the output deviates more and more from the ideal one. It is important to note here that at this stage of fatigue the driving still can be good for a long time (lower average speed, less risk etc.), and it is even very likely that the professional drivers stay on the quasi-optimal level and they leave this level just in a very few cases. But if the factors causing fatigue do not come into an equilibrium with the output though the quasi-optimal style has already entered (it can be seen, in fact, that the quasi-optimal style puts the brake on the increase of the factors of fatigue), the style of driving begins to grow worse steeply (the steepness of the deviation from the ideal case also changes) and then comes the state when a minimal additional claim for an output pushes the adaptive automaton into an instable stage.

The facts said above imply that the (regulating) scale of the driver's nervous system has to be of different width on the different level of fatigue. Our task is now to encounter experimentally this scale-width by psychophysiological measurements.

The possibility of describing the driving activity by an algorithm

Our primary studies, starting with the model above, have been based upon the hypotheses according to which the driver's work and every subactivity of it aim at a certain purpose well traceable; every action performed to this purpose are encountered by well-describable rules. Thus, the algorithm of every travelling can be set up at least theoretically. In other words, to every travelling there can be given such a sequence of instructions which, supposing all drivers obey, results that the driver leaves the road interval behind in the shortest time under the given travelling conditions.

In the most suitable way, the set up of the algorithm can be done in some formal language of the computer science. We choose the operator method of Liapunov which will turn out to be excellent for our purpose. In the frame of this method, i.e. the formal language corresponding to it, it is enough actually to use two signs (composed in general), so-called operators:

a) The sign of the so-called physical operators which represents a certain part of the driver's activity requiring mechanical work (e.g., turning the steering-wheel, operating the gas, brake and clutch pedal, etc.). These correspond to the arithmetic (valueing) operators used in the programming-theoretical application of Liapunov operators.

b) The sign of the so-called logical operators. These represent the driver's activity in deciding whether some condition important from the viewpoint of a partial activity of driving is satisfied or not (e.g., is there any traffic sign in front of the vehicle, is the vehicle going in front of him overtakeable? etc.) and depending on the fact that they are satisfied or not, he decides about the further operator (meaning physical activity or a newer decision) on the basis of which the driving activity should be continued. These latter signs correspond to logical operators used in the theory of programming.

Besides these we are going to use also another auxiliary sign, the so-called label in the setting-up of the algorithm of driving. The labels serve for marking the operators, more precisely for marking each occuring place of them in the algorithm.

Other operators appearing in the application of the Liapunov method trace to the above two kinds. The use of these serve merely for shortening the setting-up. In the algorithm of driving, among the latter ones only the so-called cycle-operators have to be taken into consideration, which refer to the driver's activity repeating itself (possibly under different values of certain parameters) several times again.

We note that, differently from numerous programming-theoretical applications, not only discrete but continuous parameters can also occur here.

As an example we present the algorithm of the overtaking in case of a two-way traffic road

$$p \overset{1}{\uparrow} I_b L_b q \overset{2}{\uparrow}_{t=t_0} \{G^+\}^S G^+ I_j L_j G^- I_0 L_0 \underset{2}{\downarrow} I_j G^- B L_j I_0 \underset{1}{\downarrow},$$

where p is a logical condition saying that with my vehicle denoted by W_0 I am in the position to prepare to overtake the vehicle W_1 going just in front of me, i.e.

$$p = \exists W_1((x(W_1) > x(W)) \wedge (v(W_1) > 0) \wedge$$

$$\wedge (v_{\max}(W_0) > v(W_1)) > (x(W_1) > (x(W_0) + 3b(v(W_0))))),$$

where $x(W)$ is the distance of the vehicle W from the start of the way; $v(W)$ is the speed of the vehicle W; $v_{\max}(W)$ is the top speed of the vehicle W; $b(v)$ is braking distance at a speed v; I_b is the operator of the left-turn indication; L_b is the operator of a turn to the left.

p says that on the road, in front of my W_0 vehicle going in the same direction as W_1 such that its top speed is higher than the speed of W_1 and its distance from W_0 three times longer than the braking distance belonging to the speed.

L_b itself also is a complicated operator: the vehicle W_0 has to be steered to the left to an extent which makes it possible for the driver to see both tracks of the road (the angle between the steered wheels and the linear axis of the car depends on the speed in this case), and then the vehicle has to be brought again in its original direction by means of the steering-wheel again. q is the logical condition saying that there is no obstacle in the overtaking.

$$q = \exists W_1[(x(W_1) > x(W_0)) \wedge (v(W_1) > 0) \wedge (v_{\max}(W_0) > v(W_1)) \wedge$$
$$\wedge \forall (W_2((x(W_2) > x(W_1)) \wedge (v(W_2) > 0)) \rightarrow ((x(W_2) > x(W_1) + 3b(v_{\max}(W_0)) \wedge$$
$$\wedge (v(W_2) < 15\ \text{km/h}) \rightarrow (x(W_2) > x(W_1) + 6b(v_{\max}(W_0)))) \wedge$$
$$\wedge \forall (W'(x(W') > x(W_0)) \wedge v(W') < 0) \rightarrow (x(W_1) + v(W_1) \frac{x(W_1) - x(W_0)}{v_{\max}(W_0) - v(W_1)} <$$
$$< (x(W') - v(W') \frac{x(W_1) - x(W_0)}{v_{\max}(W_0) - v(W_1)} + 3b(v_{\max}(W_0)))) \wedge \ldots]$$

i.e. in front of the vehicle W_1 to be overtaken, for all vehicle W_2 going in the same direction $(v(W_2) > 0)$ the condition says that

1. the distance from W_1 is three times longer than the braking distance belonging to the maximal overtaking speed $(v_{\max}(W_0))$.

2. in case the speed W_2 is less than 15 km per hours, a sixfold of the braking distance belonging to W_0 is needed and then after overtaking it is possible to range behind W_2 safely.

As far as the vehicle W' passing in the opposite direction is concerned, the condition says that during the time

$$\frac{x(W_1) - x(W_0)}{v_{\max}(W_0) - v(W_1)}$$

needed for the overtaking (at top speed), the vehicle W_1 running a way

$$v(W_1) \frac{x(W_1) - x(W_0)}{v_{\max}(W_0) - v(W_1)}$$

does not get closer to the vehicle W' running, during the same time, a way

$$v(W') \frac{x(W_1) - x(W_0)}{v_{\max}(W_0) - v(W_1)}$$

than three times the braking distance belonging to the maximal overtaking speed.

To the place ... such a logical condition has to be written which says that there is no further obstacle in the overtaking (e.g., there is no traffic sign forbidding overtaking or speed-limit etc.).

t_0 is the moment of beginning of the overtaking, s is a logical condition saying that the overtaking has happened in such a way that I can begin to go back to my lane without forcing the W_1 vehicle overtaken to diminish its speed, hence

$$x(W_0) > x(W_1) + 3b(v(W_1)).$$

G^+ is the operator of accelerating by gas; I_j is the operator of the right-turn indication; L_j is the operator of steering to the right; G^- is the operator of slowing down by taking the gas away; I_0 is the operator of the turn indication of zero-setting; B is the operator of braking; $\downarrow_1$ is turning back to the main program (usual way of travelling).

All the partial activities of driving can be set up in an analogous way and from these it is possible to synthetise the complete algorithm.

The above method of writing algorithms requires an analysis of such a deepness which fully takes into consideration the micro- and macro-situations of the traffic, the established and regulated traffic order, and which closely reflects all of these, meanwhile it fulfils the fundamental rules of the construction of algorithm. It was especially important to strive to a high-level precisity, since, it was doubtful that among the reasons causing the driver's fatigue, the repeated functioning of the operator occuring in the algorithm is of central importance. This means that in principle we should somehow (we think here, of course, of artificial receptors) experience the functioning of the different operators and we should lay down the extent to which the functioning of each operator contributes to making the specific factors of fatigue active, and in addition we should know how many times the operators have been functioning.

The functioning of physical operators are obviously indicateable, but the logical decision as a product of the functioning of the driver's mind cannot be perceived by our present means. Thus, in the man-machine system we have to look for such connections with the aid of which one can conclude to the functioning of a logical operator without trying to follow the psychological process leading to decision with our conventional instruments. Our related studies have shown that between the physical and logical operators there exists a connection such that the system sketched above will also work in case the logical operators i.e. their functioning are traced back to physical operators. In connection with this our argument is as follows: the functioning of the driver's mind effects to each directing organ of the vehicle through a sequence of mechanical reactions (physical operators). To processes of consciousness important from the point of view of the driving there always corresponds a certain state of the directing organ of the vehicle i.e. change of condition which is represented as consequences of mechanical reactions.

The study of any subalgorithm leads to similar results, if we want to indicate the functioning of the operators, it is enough to be confined to physical operators. They fully reflect the functioning of the logical operators. This is one of the most important results of the construction and study of the algorithm of driving by means of Liapunov operators.

It is easy to see that the driving activity represented by physical operators consists of three parts: 1. advancing with constant speed, 2. acceleration and slowing down, 3. change of direction. (All these are to be taken in a certain state of speed.)

Any activity necessary for the solution of a problem raised by a subalgorithm consists of these or of their combination. There exists no partial activity which could not be decomposed into the parts just mentioned.

The characteristic parameters of any of the three parts as well as the speed are quantities easily measurable: a is the acceleration, φ is the angle of steering, v is the speed. For example: on Fig. 2, let $n=3$, thus let $x_1 \approx v$, $x_2 \approx a$ and $x_3 \approx \varphi$.

We have to observe that there is a possibility of giving each subalgorithm with the parameters in question i.e. their quantitized values and numerically determining a connection between the fulfilment of tasks imposed to the driver and their burdening effects on him.

Any of these three quantities varies in a well-defined domain. These are

$$v = 0\text{—}150 \text{ km/h},$$
$$a = +2\text{—}-7 \text{ m/sec}^2,$$
$$\varphi = 0 \pm 35°,$$

when considering a vehicle of an average make and output.

Let us consider a partition of these domains into parts as small as required

$$v_i \quad (i=0, 15, 30, \ldots, 150),$$
$$a_i \quad (i=+2, +1, 0, -1, \ldots, -7),$$
$$\varphi_i \quad (i=\pm 0, \pm 0{,}5, \pm 1, \ldots, \pm 35).$$

We get sets of elements of partition as parameters. $(v, \mathfrak{A}, \mathfrak{B})$. Take the Cartesian product of these sets

$$\beta = v \times \mathfrak{A} \times \mathfrak{B}.$$

Each element of the product represents an activity of driving and the product space contains all the activities theoretically possible. It is obvious that during a travelling we get such a subset of the set of activities which can be studied by means of statistical methods and, in the same time, the time series of each element gives micro- and macro-samples.

As an example let

$$v\{15, 30, 45, 60, 75, 90, 105, 120, 135, 150\};$$
$$\mathfrak{A}\{+2, +1, 0, -1, -2, -3, -4, -5, -6, -7\};$$
$$\mathfrak{B}\{0, 1, 2, 3, 4, 5, 10, 15, 20, 25, 30\}.$$

As we have already said S_i $(i=1, 2, \ldots 1200)$ $(\in \beta)$ each represents a partial activity of driving.

Let $N_1, N_2, \ldots, N_{1200}$ the number of these ensuing during a way run.

At the end of the way let us perform a physiological measurement of fatigue. Let M the result of it.

Suppose that each of the activities $S_1, S_2, \ldots, S_{1200}$ on every occasion whenever it comes to its turn, contributes to the fatigue of the driver with quantities depending on the activity, that M can be solved as a linear function of the variables, i.e.

$$M = p_1 N_1 + p_2 N_2 + \ldots + p_{1200} N_{1200},$$

where $p_1, p_2, \ldots, p_{1200}$ are constant values. These quantities (weights) are to be empirically determined by means of a special-purpose computer built in the vehicle. We store the samples by means of an appropriate built-in store. These informations, characteristic for the dynamics of fatigue, compared with modelling processes and experiences of direct psychophysiological measurements are used for individualizing the weights obtained when using statistical methods.

Снижение аварий в дорожном транспорте с помощью применения кибернетических методов

Опасность несчастных случаев в дорожном транспорте растет весьма быстрыми темпами. Старые методы защиты оказываются недостаточно эффективными. Центральным звеном исследования защиты нового типа, направленной на снижение аварийных ситуаций, является человек, управляющий транспортом. Предлагается новый аспект этой проблемы: водитель и автомобиль рассматриваются как типичный пример человеко—машинной системы, действие которой может быть описано с помощью основных понятий кибернетики.

Теоретически можно задать алгоритм действия шофера. Исследование этого алгоритма показывает, что значительная часть факторов, определяющих усталость водителя может быть задана по объективно измеряемым параметрам с помощью математико—статистических методов. Таким образом, появляется возможность квазиобъективной оценки последствий и изменений, вызванных в операторе.

LABORATORY OF CYBERNETICS
JÓZSEF ATTILA UNIVERSITY
SZEGED, HUNGARY

(Received February 14, 1973)

INDEX—TARTALOM

Felelős szerkesztő és kiadó: Kalmár László
kézirat a nyomdába érkezett: 1972. március hó
Megjelenés: 1973. november hó
Példányszám: 1000. Terjedelem: 7,7 (A/5) ív
Készült monószedéssel, íves magasnyomással
az MSZ 5601-90 és ez MSZ 5602-55 szabvány szerint
73-1595—Szegedi Nyomda