

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:

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

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 durchgeführten 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):

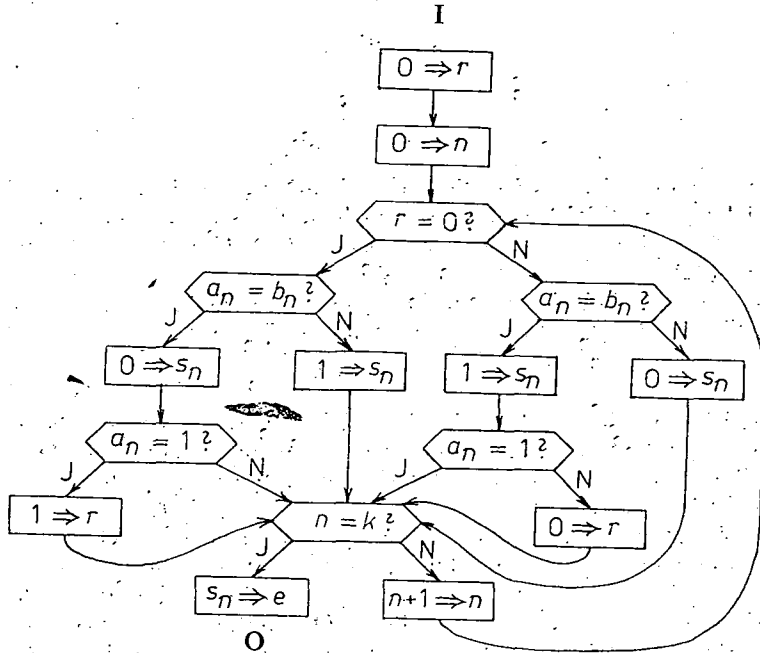


Abb. 1

Diese Einrichtung funktioniert wie folgt: Von I beginnend hat man an Kanten in ihrer Richtung entlangehend 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 gestellte 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 „Entscheidungs-tabellen“ zu ersetzen² (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:

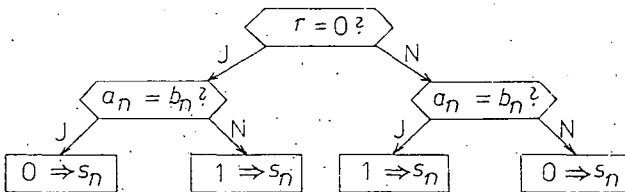


Abb. 2

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

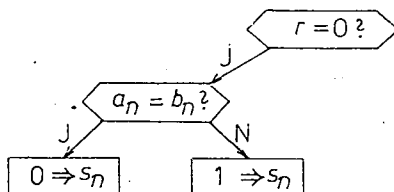


Abb. 3

zurück erhalten. 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:

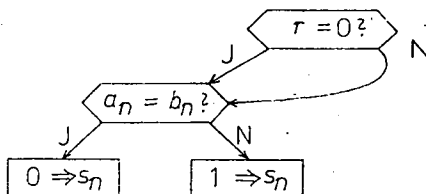


Abb. 4

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:

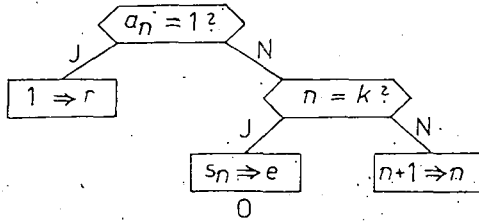


Abb. 5

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:

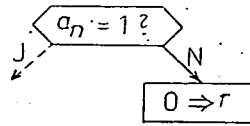


Abb. 6

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 $P_{i,j}$ beginnenden 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 Hinweis 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:

$$T_0: \begin{array}{|l|l|l|} \hline P'_{0,1} & 0 \Rightarrow r & X \\ \hline P'_{0,2} & 0 \Rightarrow n & X \\ \hline & \text{go to } T_1 & X \\ \hline \end{array}$$

(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:

$$T_0: \begin{array}{|l|l|l|} \hline P'_{0,1} & 0 \Rightarrow r & X \\ \hline P'_{0,2} & 0 \Rightarrow n & X \\ \hline & \text{go to } T_1 & X \\ \hline \end{array}$$

$$T_1: \begin{array}{|l|l|l|l|l|} \hline P_{1,1} & r = 0? & J & J & N & N \\ \hline P_{1,2} & a_n = b_n? & J & N & & \\ \hline P_{1,3} & a_n = b_n? & & & J & N \\ \hline P'_{1,1} & 0 \Rightarrow s_n & X & & & \\ \hline P'_{1,2} & 1 \Rightarrow s_n & & X & & \\ \hline P'_{1,3} & 1 \Rightarrow s_n & & & X & \\ \hline P'_{1,4} & 0 \Rightarrow s_n & & & & X \\ \hline & \text{go to } T_2 & X & & & \\ \hline & \text{go to } T_{2,2} & & X & & X \\ \hline & \text{go to } T_3 & & & X & \\ \hline \end{array}$$

$$T_2: \begin{array}{|l|l|l|l|l|} \hline P_{2,1} & a_n = 1? & J & N & N \\ \hline P_{2,2} & n = k? & & J & N \\ \hline P'_{2,1} & 1 \Rightarrow r & X & & \\ \hline P'_{2,2} & s_n \Rightarrow e & & X & \\ \hline P'_{2,3} & n+1 \Rightarrow n & & & X \\ \hline & \text{go to } T_{2,2} & X & & \\ \hline & \text{stop} & & X & \\ \hline & \text{go to } T_1 & & & X \\ \hline \end{array}$$

Dabei ist

$$T_{2,2}: \begin{array}{|l|l|l|l|} \hline P_{2,2} & n = k? & J & N \\ \hline P'_{2,1} & s_n \Rightarrow e & X & \\ \hline P'_{2,2} & n+1 \Rightarrow n & & X \\ \hline & \text{stop} & X & \\ \hline & \text{go to } T_1 & & X \\ \hline \end{array}$$

$$T_3: \begin{array}{|l|l|l|l|} \hline P_{3,1} & a_n = 1? & J & N \\ \hline P'_{3,1} & 0 \Rightarrow r & & X \\ \hline & \text{go to } T_{2,2} & X & X \\ \hline \end{array}$$

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

Ferner wird auch als nützlich geschätzt, 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 Tabelle T_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 dienen, 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. Führt 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

$$P'_{0,1}, P'_{0,2}, \dots, P'_{0,1}$$

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 $P'_{i,j}$ ($i \geq 1$) 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 “.

Würden bereits Linien L_1, L_2, \dots, 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 $P'_{i,j}$ ($i \geq 1$) 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 $i \geq 2$; 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, \dots, T_n ähnlich gebildet wurden, und Q ein Punkt des betrachteten Graphen ist, der in keinem von diesen auftritt. Sei $1 \leq i \leq n$ 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

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

sein. Dann mußte aber auch „go to T_{n+1} “ als Spaltenausgang in einem der Tabellen T_1, \dots, 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, \dots, 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, \dots, 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 Programmiersprachen übersetzt werden³).

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

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

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 ($m \leq n$) können Anweisungen folgender Form vorkommen:

$$\text{stop, go to } T_k, \text{ go to } T_{i,j}, \text{ go to } T_{i,j}$$

(wobei $i \leq m$ 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 Programmiersprache 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 $P_{i,1}, \dots; P'_{i,1}, \dots$ 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 bzw. N , und kam das nächste nicht leere Zeichen (vor dem Ausgang) dieser Spalte in der 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 zu 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 Programmiersprachen 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_i ist. Dann sind (mathematische) Punkte

$$P'_{i,1}, \dots, P'_{i,x_1}, P'_{i,x_1+1}, \dots, P'_{i,x_1+x_2}, \dots, P'_{i,x_1+x_2+\dots+x_i}$$

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 $P'_{i,1}, \dots, P'_{i,x_1}$ 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 $P'_{i,x_1+1}, \dots, P'_{i,x_1+x_2}$ 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.

Betrachten wir z. B. eine zur Betriebsorganisation verwendete Entscheidungstabelle, die auf S. 19 des in Fußnote² zitierten Buches angegeben wird. Diese kann mit den Bezeichnungen

F_1, F_2, F_3, F_4 bzw. 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. Anweisung-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 2⁴ = 16 Spalten, wieviele zu bilden wären, wenn für die 4 verschiedenen Fragen alle Antwort-Variationen aufgenommen würden.

Von dieser Tabelle sind die von P₁ ausgehenden Linien des entsprechenden Vorgraphschemas Spalte für Spalte verfolgend unmittelbar abzulesen. So ergibt sich:

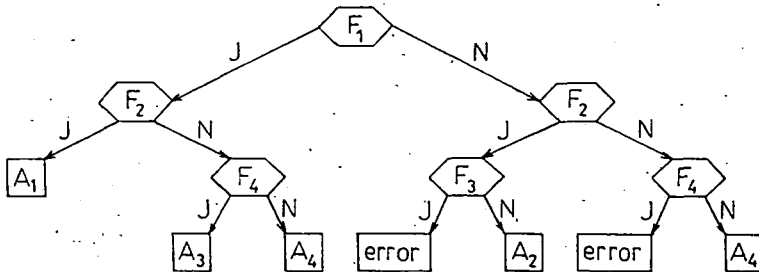


Abb. 7

§ 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

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

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

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

und

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

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

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

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

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

die Folge

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

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 vermeintlich 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:

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

(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

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

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

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

bzw.

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

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

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

zugeordnet.

Doch dem Output O wird folgende mathematische Funktion zugeordnet:

$$f_0(k, a_0, \dots, a_k, b_0, \dots, b_k, r, n, s) = s.$$

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

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

(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:

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

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

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

ist, wo sowohl m_1 als auch m_2 mit einem der n_i ($i=1, 2, \dots, u$) 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 Anfangsfunktionen auf. Solche Tabellensysteme können „Normaltabellensysteme“ genannt werden.

Ich habe bewiesen¹, 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 Computers 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 Computers immer als die Berechnung der Werte einer zahlentheoretischen Funktion betrachtet werden. Wird für die Zelleninhalte keine Schranke gestellt, so kann man beweisen, daß die durch Computer berechenbaren zahlentheoretischen Funktionen mit den partiell-rekursiven Funktionen identisch sind⁴. So sind diese nach den Vorherigen auch mit den durch Normaltabellensysteme berechenbaren Funktionen identisch. Man kann sagen: was die Computer 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 безосторожного определения. Для "flow" диаграмм уже раньше Калзний 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 программный 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 Computer-Theorie*.