Kybernetika

Heinz Stahn
Über die Methode der Entscheidungstabellen

Kybernetika, Vol. 16 (1980), No. 2, (172)--182

Persistent URL: http://dml.cz/dmlcz/125593

KYBERNETIKA — JAHRGANG 16 (1980), HEFT 2

Über die Methode der Entscheidungstabellen[1])

HEINZ STAHN

Die Methode der Entscheidungstabellen wurde zu einem leistungsfähigen algorithmischen System entwickelt. Nach dem Vergleich der Entscheidungstabellentechnik mit anderen algorithmischen Systemen wird ein Beispiel eines Steuerungssystems mit parallelen Operationen diskutiert.

1. ALGORITHMISCHE SYSTEME

Wir wollen hier kurz einige bekannte algorithmische Systeme und die Methode der Entscheidungstabellen charakterisieren. Dabei bezeichnen wir als algorithmisches System jedes allgemeine Verfahren zur Darstellung von Algorithmen.

MARKOWsches Normalalgorithmensystem[2])

MARKOWsche Normalalgorithmen (MNA) sind Substitutionsalgorithmen für Zeichenketten. Ein MNA besteht aus einer geordneten Menge von Regeln (Substitutionsregeln). Jede Regel

$b \to a$

ist zu interpretieren als:

wenn b ein linksoptimales Teilwort der vorgelegten Zeichenkette ist,
so ersetze das Teilwort b durch das Teilwort a und gehe zur 1. Regel über, falls es keine Stop-Regel ist,
anderenfalls gehe zur nächsten Regel.

[1]) Kolloquium an der TH Bratislava, EF — Katedra počitačov, am 8. 12. 1977.

[2]) Wir weichen hier von der üblichen Bezeichnung ab, wir wollen jedoch mit dieser Bezeichnung betonen, daß zwischen dem algorithmischen System und danach dargestellten, konkreten Algorithmen zu unterscheiden ist.

Stop-Regeln werden speziell markiert. In MNA können Zustandsgrößen nur dann eingeführt werden, wenn diese während der Anwendung des Algorithmus in die Operanden eingeführt werden können.

Wesentliche Merkmale von MNA sind also:

- die Elemente sind Regeln
- die Anwendungsbedingung einer Regel ist die linksoptimale Enthaltensrelation für die Zeichenkette a
- die Menge der Regeln ist algorithmisch relevant geordnet
- nach jeder Anwendung einer Regel wird zur ersten Regel zurückgesprungen
- es sind keine Zustandsgrößen möglich.

Algorithmisches System nach McCARTHY [5]

Bedingte Ausdrücke werden als geordnete Menge von Regeln

$b_i \Rightarrow a_i$

geschrieben. Die b_i stehen für BOOLEsche Ausdrücke, die entweder wahr oder falsch sein können. Die a_i bezeichnen Ausdrücke. Eine Regel ist wie folgt zu interpretieren:

wenn das Einsetzen der vorgelegten Operanden für b_i einen wahren Ausdruck ergibt,

so führe den Ausdruck a_i aus und gehe zurück zur 1. Regel, falls es keine Stop-Regel ist,

anderenfalls gehe zur nächsten Regel.

Wesentliche Merkmale dieses algorithmischen Systems sind:

- die Elemente sind Regeln
- die Regeln sind logische Implikationen
- die Menge der Regeln ist algorithmisch relevant geordnet
- nach jeder Anwendung einer Regel wird zur ersten Regel zurückgesprungen
- die letzte Regel ist eine unbedingte Regel, eine „else-Regel“
- nicht nur zum Abschluß eines Algorithmus sind unbedingte Regeln möglich.

TURINGsches Automatenschema[3])

Elemente eines TURING-Programmes, dargestellt in einer TURING-Tafel, sind Befehle. Elemente der Befehle sind die Regeln, auch Befehlszeilen genannt. Eine

[3]) Es gilt das entsprechende wie bei 2.

 Regel

(x, z, z', y, w)

ist wie folgt zu interpretieren:

wenn der anliegende Eingangswert $\xi(t) = x$ und der vorhandene Zustandswert $\zeta(t) = z$ ist,

so setze $\zeta(t + 1) = z'$ und gebe den Ausgangswert $\eta(t) = y$ aus und gehe zum Operandenfeld mit der relativen Adresse w über und verlasse die Tabelle,

anderenfalls gehe zu einer beliebigen anderen Regel.

Wesentliche Merkmale des TURINGschen Automatenschemas sind:

- die Elemente sind Regeln
- die Regeln sind logische Implikationen
- die Menge der Regeln ist nicht algorithmisch relevant geordnet
- nach jeder Anwendung einer Regel wird die Tafel verlassen
- die Verwendung von Zustandsgrößen wird ausdrücklich ermöglicht bzw. gefordert
- die Fortsetzung des Prozesses wird durch Übergang zu einem angegebenen Operandenfeld und nachfolgende Anwendung derselben Tafel realisiert.

Algorithmisches System für Mikroprogramme [2]

Ein Mikroprogramm besteht aus einer geordneten Menge von Anweisungen, jede Anweisung besteht aus einer geordneten Menge von Regeln. Jede Regel

(A, y, j)

ist wie folgt zu interpretieren:

wenn der Eingangswert $\xi(t) \in A$ ist,

so führe die Ausgabeaktion $\eta(t) = y$ aus und gehe über zur ersten Regel der Anweisung j, falls es keine Stop-Regel ist,

anderenfalls gehe zur nächsten Regel.

Wesentliche Merkmale dieses algorithmischen Systems sind:

- die Elemente sind Regeln
- die Regeln sind logische Implikationen
- die Menge der Regeln ist algorithmisch relevant geordnet
- nach Anwendung einer Regel wird zu einer angegebenen Regel übergegangen
- die mengentheoretische Enthaltensrelation wird als Bedingung gefordert.

Im Rahmen dieses Konzeptes werden die Entscheidungsmöglichkeiten vollzählig aufgezählt. Elemente einer Entscheidungstabelle sind die Regeln, die Menge der Regeln ist nicht algorithmisch relevant geordnet.

Eine Regel

(Bedingung i, φ_i, Bedingung j, φ_j, ..., x, Aktion l, x, Aktion m, x, ...)
üblicherweise dargestellt

Bedingung i	φ_i
Bedingung j	φ_j
...	
Aktion l	x
Aktion m	x
...	

ist wie folgt zu interpretieren:

wenn durch das Einsetzen der vorgelegten Operanden die Bedingung i den Wahrheitswert φ_i und die Bedingung j den Wahrheitswert φ_j und ... hat,
so führe die Aktion l und die Aktion m und ... aus und verlasse die Tabelle,
anderenfalls gehe zu einer beliebigen anderen Regel.

Wesentliche Merkmale dieses Entscheidungstabellenkonzeptes sind:

- die Elemente sind Regeln
- die Regeln sind logische Implikationen
- Bedingungen (Prämissen) und Aktionen (Conclusionen) können in beliebiger, auch verbaler Form gegeben sein. Dabei wird zu jeder Bedingung der erforderliche Wahrheitswert explizite vorgegeben
- die Menge der Regeln ist algorithmisch nicht geordnet
- nach jeder Anwendung einer Regel wird die Tabelle verlassen.

Zusammenfassend stellen wir fest, daß alle vorgenannten algorithmischen Systeme für die Beschreibung von parallelen Prozessen nicht geeignet sind.

Das erweiterte Entscheidungstabellenkonzept [7]

Nach einer Reihe von Zwischenstufen wurde das nachfolgende Konzept entwickelt. Da wir uns mit diesem Konzept befassen wollen, diskutieren wir zuerst ein einfaches Beispiel, den abstrakten Koinzidenzautomaten.

koinzidenz	Bedingungen	Aktionen	goto
1	—	$z = z + x$	
2	$z \in 0Z$	$y = 1$	*nt*
3	—	$y = 0$	*nt*

Die Regeln dieser Tabelle sind wie folgt zu interpretieren:

Regel 1: Führe unbedingt die Aktion (= Anweisung) $z = z + x$ aus und gehe zur nächsten Regel.

Regel 2: *wenn* die Bedingung $z \in 0Z$ erfüllt ist,
so führe die Aktion $y = 1$ aus und verlasse die Tabelle,
anderenfalls gehe zur nächsten Regel.

Regel 3: Führe unbedingt die Aktion $y = 0$ aus und verlasse die Tabelle.

Eine Automatentabelle zur Beschreibung desselben Automaten wäre sehr groß.

Allgemein hat eine Regel die folgende Form:

i: Aussageformverbindung // Verbindung von Anweisungen / goto k und ist zu interpretieren:

wenn die Anwendung der Aussageformverbindung auf das vorgelegte Situationstupel eine wahre Aussage ergibt,
so führe die Gesamtheit der Anweisungen aus und gehe zur Regel k,
anderenfalls gehe zur nächsten Regel.

Völlig äquivalent kann man dafür schreiben:

i: Eingangsbedingungen und Zustandsbedingungen // Zustandsaktionen und Ausgangsaktionen / goto k.

Dieses ist zu interpretieren:

wenn durch das Einsetzen der vorgelegten Operanden die Eingangsbedingungen und die Zustandsbedingungen erfüllt sind,
so führe die Zustandsaktionen und die Ausgangsaktionen aus und gehe zur Regel k,
anderenfalls gehe zur nächsten Regel.

Die wesentlichen Merkmale dieses Entscheidungstabellenkonzeptes sind:

- die Elemente sind Regeln
- die Regeln sind logische Implikationen
- die Menge der Regeln ist algorithmisch relevant geordnet
- nach Anwendung einer Regel wird zu einer Regel derselben Tabelle oder zur ersten Regel einer anderen Tabelle übergegangen

	Bedingungen	Aktionen	goto	Kommentar
1	$(he = h, ew_1 = 0, z_1 = 1, ew_2 = 0, z_2 = 1, et = t, zt = 1)$	$(da_1 = 0, da_2 = 0, vt = v)$		Auslösen der Tischbewegung
2	$(he = 0, ew_1 = 0, z_1 = 1, ew_2 = 0, z_2 = 1, et = 0, zt = 1)$	$(zt = 0)$		Änderung der Zustandsgröße zt
3	$(he = 0, ew_1 = 0, z_1 = 1, ew_2 = 0, z_2 = 1, et = 1, zt = 0)$	$(zt = 1, vt = 0, da_1 = 1,$ $vw_1 = v, nw_1 = n,$ $vw_2 = v, nw_2 = n)$	5	Auslösen aller Operatoren und Freigabemeldung ($da_1 = 1$)
4	—		1	Warten auf Eintreten der Ereignisse
5	$(-, ew_1 = 1, z_1 = 1, -, -, et = t, zt = 1)$	$(z_1 = 0, vw_1 = r)$		Umschalten Vorschub Werkzeug 1, Änderung der Zustandsgröße z_1
6	$(-, ew_1 = 0, z_1 = 0, -, -, et = t, zt = 1)$	$(z_1 = 1, vw_1 = 0, nw_1 = 0)$		Ausschalten Vorschub und Antrieb Werkzeug 1, Änderung der Zustandsgröße z_1
7	$(-, -, -, ew_2 = 1, z_2 = 1, et = t, zt = 1)$	$(z_2 = 0, vw_2 = r)$		Umschalten Vorschub Werkzeug 2, Änderung der Zustandsgröße z_2
8	$(-, -, -, ew_2 = 0, z_2 = 0, et = t, zt = 1)$	$(z_2 = 1, vw_2 = 0, nw_2 = 0)$		Ausschalten Vorschub und Antrieb Werkzeug 2, Änderung Zustandsgröße z_2
9	$(he = h, ew_1 = 0, z_1 = 1, ew_2 = 0, z_2 = 1, et = t, zt = 1)$	—	1	Der Zyklus ist abgearbeitet, Rücksprung zum Anfang
10		—	5	Warten auf Eintreten der Ereignisse

→ Steuerungsprozeß

Bild 1. Entscheidungstabelle Steuerungssystem einer Taktstraße mit parallelen Operationen nach dem Taktprinzip.

– die Verwendung von Zustandsgrößen des Algorithmus wird ausdrücklich zugelassen; die Einführung von „Aktivierungsvariablen" für parallele Prozesse wird in Teil 2 vollzogen
– unbedingte Regeln sind an beliebiger Stelle möglich.

2. BEISPIEL ZUR STEUERUNG EINER TAKTSTRASSE

Wir diskutieren dieses Beispiel in zwei Varianten: ohne und mit Verwendung von Aktivierungsvariablen.

Steuerung einer Taktstraße [7]

Das Diagramm des Steuerungsprozesses zeigt die wesentlichen Aspekte. Dabei wurde dieses Diagramm direkt in die Entscheidungstabelle (Bild 1) hineingezeichnet, um die gegenseitige Zuordnung zu verdeutlichen. Die letzte Regel der Tabelle ist eine „else-Regel", die das zyklische Durchmustern der Tabelle im angewendeten Taktprinzip sichert. Das Beispiel zeigt, daß das erweiterte Entscheidungstabellenkonzept auch für die Beschreibung paralleler Prozesse geeignet ist.

Steuerung einer Taktstraße unter Verwendung von Aktivierungsvariablen [1]

Entsprechend der Methodik der PETRI-Netze werden neben den Eingangs-, Zustands- und Ausgangsgrößen auch Aktivierungsvariable eingeführt. Für eine Operation im PETRI-Netz [6] gilt die logische Implikation (Bild 2):

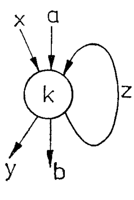

Bild 2. Operation im PETRI-Netz.

wenn alle Eingangsgrößen x und alle Zustandsgrößen z und alle Aktivierungsvariablen a aktiviert sind,
so führe alle Zustandsaktionen z und alle Ausgangsaktionen y und alle Aktivierungsaktionen b aus und gehe zu einer Operation,
anderenfalls gehe zu einer anderen Operation.

	Bedingungen	Aktionen	goto	Kommentar
1	$(a = 1, b = 1, c = 1)$	$(a = 2, b = 0, c = 0, da_2 = 0, vt = 0)$	nt	Auslösen der Tischbewegung
2	$(a = 2, et = 0)$	$(a = 3)$	nt	Tisch ist in Bewegung, Zustandsgrößenänderung
3	$(a = 3, et = 1)$	$(a = 4, b = 4, c = 4, da_1 = 1, vw_1 = v, vw_1 = n, vw_2 = v, vw_2 = n, vt = 0)$	nt	Ausschalten Vorschub Tisch, Auslösen aller Bearbeitungsoperatoren, Freigabe Handoperator
4	$(a = 4, he = h)$	$(a = 1, da_1 = 0, da_2 = 2)$	nt	Handoperation ist vollzogen
5	$(b = 4, ew_1 = 1)$	$(b = 5, vw_1 = r)$	nt	Umschalten Vorschub Werkzeug 1
6	$(b = 5, ew_2 = 0)$	$(b = 1, vw_1 = 0, nw_1 = 0)$	nt	Ausschalten Werkzeug 1, Operation ist vollzogen
7	$(c = 4, ew_2 = 1)$	$(c = 5, vw_2 = r)$	nt	Umschalten Vorschub Werkzeug 2
8	$(c = 5, ew_2 = 0)$	$(c = 1, vw_2 = 0, nw_2 = 0)$	nt	Ausschalten Werkzeug 2, Operation ist vollzogen

Bild 3. Entscheidungstabelle Steuerungssystem einer Taktstraße nach dem Ereignisprizip unter Verwendung von Aktivierungsvariablen.

Dabei ist eine Größe aktiviert, wenn die entsprechende logische Bedingung wahr ist.

Für die Verbundaktionen in der zu diskutierenden Entscheidungstabelle (Bild 1) müssen wir noch eine Vereinbarung über die Arbeitsweise des Systems treffen. Bezeichnen wir jede Änderung einer Eingangsgröße, Zustandsgröße oder Aktivierungsvariablen als Ereignis, so kann das System nach dem Ereignisprinzip arbeiten, und es besitzt dann ein Unterbrechungssystem, das dann den Eingang in die Entscheidungstabelle realisiert. In diesem Fall wird die Entscheidungstabelle nach Anwendung einer Regel wieder verlassen und die Regie and das Unterbrechungssystem zurückgegeben.

Nach dem Taktprinzip würde sich das System in der Entscheidungstabelle in einer Warteschleife befinden, zyklisch alle Regeln auf Anwendbarkeit durchmustern und so im Taktbetrieb auf das Eintreten eines Ereignisses warten.

Die angegebene Tabelle (Bild 3) ist für das Ereignisprinzip formuliert, für das Taktprinzip müßte eine weitere, letzte Regel mit unbedingtem Rücksprung zur ersten Regel angegeben werden.

Abschließend vergegenwärtigen wir uns, daß dieses Steuerungssystem ST nur im Zusammenwirken mit dem Basissystem BS arbeiten kann, weil sonst die Eingangsgrößen keinen Sender und die Ausgangsgrößen keinen Empfänger haben. Im GLUSCHKOWschen Sinne [4] (Bild 4) ist also das BS das Interpretationssystem

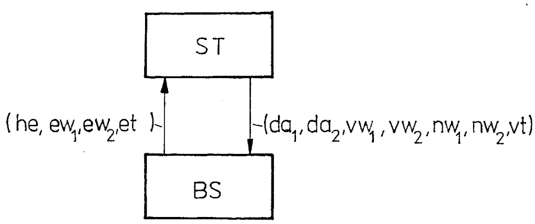

Bild 4. Fundamentalrelation Basissystem BS — Steuerungssystem ST.

für das Steuerungssystem ST. Dabei wird für das BS die folgende (Bild 5) Transformationsmatrix vorausgesetzt.

$y \uparrow$	da_1	da_2	vw_1	nw_1	vw_2	nw_2	vt $\xrightarrow{x}$
he	+	+					
ew_1			+				
ew_2					+		
et							+

wenn $(x + y)$, so ist y von x abhängig

Bild 5. Transformationsmatrix für das Basissystem.

Die klassischen algorithmischen Systeme sind für die Beschreibung von Systemen mit parallelen Prozessen ungeeignet. Derzeitig sind nur wenige moderne Systeme bekannt, so z. B. die PETRI-Netze [6].

Das erweiterte Entscheidungstabellenkonzept [7]

Die vorn genannten wesentlichen Merkmale dieses Konzeptes können wir gemäß Teil 2 wie folgt ergänzen:

- Zur Beschreibung des Steuerungsprozesses sind die Eingangsgrößen vom Basisprozeß, die Ausgangsgrößen zum Basisprozeß, Steuerungszustandsgrößen und eben auch Aktivierungsvariable zulässig. Dadurch sind sowohl parallele Teilprozesse im Basisprozeß als auch parallele Teilprozesse im Steuerungsprozeß zulässig und beschreibbar.

Algorithmisches System für Mikroprogramme [4]

Ein solches Programm besteht aus einer geordneten Menge von Befehlen. Ein Befehl hat die allgemeine Form:

$$\begin{aligned} L_i : A_i/x_1 \,.\, z_1 &: L_j, L_k, \ldots ; \\ x_2 \,.\, z_2 &: L_l, L_m, \ldots ; \\ \ldots &; \end{aligned}$$

und ist zu interpretieren:

wenn die Aktivierungsvariable L_i aktiviert ist,
so führe die Menge A_i von Aktionen aus
und wenn die Bedingungen x_1 des Basisprozesses und die Bedingungen z_1 des Steuerungsprozesses erfüllt sind,
so aktiviere die Aktivierungsvariablen L_j und L_k und ...
und wenn die Bedingungen x_2 des Basisprozesses und die Bedingungen z_2 des Steuerungsprozesses erfüllt sind,
so aktiviere die Aktivierungsvariable L_l und L_m und ... und ...
anderenfalls gehe zum nächsten Befehl.

Es zeigt sich, daß beide genannten Systeme algorithmisch äquivalent sind.

(Eingegangen am 2. Mai 1978.)

[1] H. J. Busch, M. Engelien, H. Stahn: Das algorithmische System Entscheidungstabellentechnik. Akademie-Verlag, Berlin 1980.

[2] N. Frištacký: Sequence Control Systems. University of Salford 1971.

[3] N. Frištacký, M. Kolesar: Modular Parallel Programs and their Implementation in Programmable Logic Controllers. 21. Internationales Kolloquium TH Ilmenau 1976.

[4] V. M. Gluschkov, A. A. Letichevskii: Theory of algorithms and discrete processors. In: Advances in Information Systems Science, Vol.I, J. T. Tou (Ed.). Plenum Press, New York 1969, 1—58.

[5] J. McCarthy: Recursive functions of symbolic expressions and their computation by machines. CACM *3* (1960), 4, 184 ff.

[6] C. A. Petri: Grundsätzliches zur Beschreibung diskreter Prozesse. In: 3. Colloquium über Automatentheorie (Hannover 1965). Birkhäuser-Verlag, Basel und Stuttgart 1967, 121—140.

[7] H. Stahn: Entscheidungstabellen als Darstellungsmittel abstrakter Automaten. Wiss. Zeitschrift TU Dresden *23* (1974), 1, 159—162.

Prof. Dr. sc. techn. Heinz Stahn, Technische Universität Dresden, Sektion Informationsverarbeitung, Mommsenstr. 13, 8027 Dresden. DDR.