Elektronische Informationsverarbeitung und Kybernetik
EIK 9 (1973) 10, 635–650

Feedback-orientierte Kodierung endlicher Automaten

Von Karl-Adolf Zech[1])

0. Einleitung

Diese Note gibt ein Verfahren für die Kodierung der inneren Zustände endlicher Automaten an, bei dem die Realisierung (digitale Schaltung) einem bestimmten Schema genügt. Der Algorithmus erhebt nicht den Anspruch, streng optimale Schaltungen zu liefern, jedoch wird erwartet, daß im Rahmen der vorgegebenen Schaltungsstruktur „gute" Ergebnisse erzielt werden können.

Bis heute sind eine ganze Reihe von Arbeiten über die Sekundärkodierung endlicher Automaten erschienen. Sie beschreiben entweder auf heuristischen [3–5], statistischen [6] oder anderen [7, 8, 15–17] Grundlagen basierende Kodierungsalgorithmen, oder sie machen die in der Theorie der Partitionen und Überdeckungen als „Informationsflußungleichungen" formulierbaren internen Abhängigkeiten des Automaten für eine Kodierung nutzbar [1, 9–12].

In Hartmanis/Stearns [1] wird ein Formalismus für die Beherrschung des Informationsrückflusses angegeben, der für eine bestimmte Anordnung einer digitalen Schaltung bzw. eines Automatennetzes auftritt. Die rückgekoppelte Information über den gegenwärtigen Zustand, hier kurz Feedback genannt, entspricht einer Partition bzw. Überdeckung der Zustandsmenge. Davon ausgehend lassen sich Kodierungspartitionen recht einfach ermitteln. Diese sind jedoch nicht eindeutig bestimmt, ihre Fixierung hat einen wesentlichen Einfluß auf den entstehenden Realisierungsaufwand.

Folgende Einschränkungen sind zu berücksichtigen: 1) Es werden nur getaktete Verzögerungsspeicher (Delay-Flip-Flops) verwendet, und 2) die Ausgabefunktion des Automaten wird nicht mit betrachtet.

Zur Notation von Aussagen und Relationen wird die Sprache des Prädikaten-Funktionen-Kalküls herangezogen (vgl. Starke [2]).

1. Partitionen und Überdeckungen der Zustandsmenge

Die in diesem Abschnitt nur kurz vorgestellte Theorie der Automatenstrukturen wurde zu Beginn der Sechziger Jahre von den Mathematikern J. Hartmanis und R. E. Stearns entwickelt und in [1] ausführlich dargestellt. Es ist für das Verständnis der folgenden Definitionen und Zusammenhänge vorteilhaft, mit ihren Grundlagen vertraut zu sein.

Wir gehen in folgendem von dem endlichen Halbautomaten $A = [X, Z, f]$ aus. Dabei ist X die Menge der *Eingabesignale* und Z die der *inneren Zustände*.

[1]) Institut für Nachrichtentechnik Berlin, Rechenzentrum, Abteilung Technische Probleme.

Beide Mengen sind endlich und nicht leer. Die *Überführungsfunktion* $f : Z \times X \to Z$ ordnet jedem gegenwärtigen Zustand z aus Z und Eingabesignal x aus X eindeutig einen Folgezustand $z' = f(z, x)$ zu.

Das Verhalten einer digitalen Schaltung S kann durch einen endlichen Automaten A beschrieben werden. Die Speicherbelegungen von S (BOOLEsche Vektoren) sind dann eindeutig den Zuständen von A zugeordnet. Offensichtlich werden jedem einzelnen Speicher i von S durch jede seiner beiden möglichen Belegungen 0 und 1 zwei Mengen $N_i(0)$ und $N_i(1)$ von Zuständen in A zugewiesen mit $N_i(0) \cup N_i(1) = Z$. Ist die Zuordnung der Speicherbelegungen zu den Zuständen eineindeutig, so sind die Überdeckungen $\{N_i(0), N_i(1)\}$ Zerlegungen von Z, in folgendem als *Partitionen* bezeichnet.

Die algebraische Strukturtheorie untersucht umgekehrt den Zusammenhang zwischen den Eigenschaften von Partitionen/Überdeckungen der Zustandsmenge eines Automaten und der Struktur der Realisierung, die diesen entsprechen.

1.1. Definitionen und Aussagen

Sei Z eine endliche, nichtleere Menge.

(i) Eine *Überdeckung* π von Z ist eine Teilmenge der Menge $\mathfrak{P}^*(Z)$ aller nichtleeren Teilmengen von Z mit $\bigcup \pi = Z$ und $\forall B \forall B' (B, B' \in \pi \wedge B \subseteqq B' \to B = B')$. Die Elemente von π heißen *Blöcke*, ihre Anzahl wird mit $\#(\pi)$ bezeichnet.

(ii) Eine *Zerlegung* oder *Partition* π von Z ist eine Überdeckung mit $\forall B \forall B' (B, B' \in \pi \wedge B \neq B' \to B \cap B' = \emptyset)$.

(iii) Sei π eine Überdeckung von Z. $\underset{\pi}{\sim}$ heißt die durch π induzierte Verträglichkeitsrelation über Z mit $z \underset{\pi}{\sim} z' \underset{\mathrm{Df}}{\leftrightarrow} \exists B (B \in \pi \wedge \{z, z'\} \subseteqq B))$ für z, z' aus Z. Ist π eine Partition, so ist $\underset{\pi}{\sim}$ eine Äquivalenzrelation. Umgekehrt induziert jede Verträglichkeitsrelation R über Z eine Überdeckung $[Z/R]$ mit $[Z/\underset{\pi}{\sim}] \supseteqq \pi$ und $\underset{[Z/R]}{\sim} = R$.

(iv) Seien π und τ Überdeckungen von Z. π heißt *höchstens feiner (kleiner)* als τ $(\pi \leqq \tau)$, wenn $\forall B \exists B' \; (B \in \pi \wedge B' \in \tau \wedge B \subseteqq B')$. π heißt (echt) *feiner (kleiner)* als $\tau (\pi < \tau)$, wenn $\pi \leqq \tau$ und $\pi \neq \tau$.

(v) Da $\leqq$ eine antisymmetrische Relation ist, sind das Supremum $\sup(\pi, \tau)$ und das Infimum $\inf(\pi, \tau)$ für zwei Überdeckungen π und τ eindeutig bestimmt. Daher können die Operationen $+$ und $\cdot$ durch $\pi + \tau =_{\mathrm{Df}} \sup(\pi, \tau)$ und $\pi \cdot \tau =_{\mathrm{Df}} \inf(\pi, \tau)$ definiert werden. Bezeichnet $\mathfrak{M}$ die Menge aller Partitionen bzw. Überdeckungen von Z, so ist $\inf(\mathfrak{M}) = 0$ die Zerlegung von Z in Einermengen und $1 = \sup(\mathfrak{M}) = \{Z\}$. 0 ist die Einheit der Addition und 1 die der Multiplikation. $\langle \mathfrak{M}, +, \cdot \rangle$ ist ein i. a. nicht-distributiver, jedoch komplementärer Verband. Man beachte, daß zwischen dem Verband der *Überdeckungen* und dem der *Partitionen* von Z wesentliche Unterschiede bestehen. So läßt sich beweisen, daß für Partitionen π und τ die Beziehung $\inf(\pi, \tau) = \{B \cap B' | B \in \pi \wedge B' \in \tau \wedge B \cap B' \neq \emptyset\}$ und $\sup(\pi, \tau) = \{B | \{z, z'\} \subseteqq B \leftrightarrow \exists m \exists z_0 \exists \cdots \exists z_m \, (m \geqq 2 \wedge \{z_0, \ldots, z_m\} \subseteqq Z \wedge z = z_0 \wedge z' = z_m \wedge \forall j \, (j \in \{0, \ldots, m-1\} \to z_j \underset{\pi}{\sim} z_{j+1} \vee z_j \underset{\tau}{\sim} z_{j+1}))$ gelten, während bei Überdeckungen $\inf(\pi, \tau) = \max(\{B \cap B' | B \in \pi \wedge B' \in \tau\})$ und $\sup(\pi, \tau) + = \max(\pi \cup \tau)$ ist.

Überdeckungen notieren wir vereinfachend mit Hilfe einer durch Schrägstriche getrennten Angabe aller Blöcke. Beispielsweise schreiben wir für $\pi = \{\{1, 2, 3\}, \{4, 5, 6\}, \{6, 7\}\}$ kurz $\pi = /1, 2, 3 \,/\, 4, 5, 6 \,/\, 6, 7/$.

Beispiele. $\pi = /1, 2, 3 / 4, 5 / 6/$ und $\tau = /1, 2 / 3, 6 / 4, 5/$ sind Partitionen mit $\inf(\pi, \tau) = \pi \cdot \tau = /1, 2 / 3 / 4, 5 / 6/$ und $\sup(\pi, \tau) = \pi + \tau = /1, 2, 3, 6 / 4,5/$. Im Verband der Überdeckungen aber ist $\sup(\pi, \tau) = /1, 2, 3 / 3, 6 / 4, 5/$.

Man kann den Begriff der Überdeckung mit dem der *Information* verbinden: Die Information über einen Zustand bei Kenntnis des ihn enthaltenden Blockes ist um so größer, je feiner die Überdeckung ist.

Im folgenden seien π, τ und ϱ stets Partitionen bzw. Überdeckungen von der Zustandsmenge Z des Automaten A.

1.2. Definition. $[\pi, \tau]$ heißt *Partitionspaar* bzw. *Überdeckungspaar* für A (kurz: PP bzw. ÜP), wenn $\forall x \forall B \exists B'$ $(x \in X \wedge B \in \pi \wedge B' \in \tau \wedge f(B, x) \subseteq B')$, wobei unter $f(B, x)$ die Vereinigung $\bigcup_{z \in B} \{f(z, x)\}$ zu verstehen ist.

Beispiel. Für $A = [\{0, 1, 2\}, \{1, 2, 3, 4, 5\}, f]$ mit

f	1	2	3	4	5
0	3	4	5	5	4
1	1	3	4	1	3
2	5	2	3	5	2

ist bei $\pi = /1, 4 / 3 / 2, 5/$ und $\tau = /1, 4/2, 3, 5 /$ $[\pi, \tau]$ ein PP.

1.3. Definition. (i) $m(\pi) =_{\mathrm{Df}} \inf(\{\tau | [\pi, \tau] \text{ ist PP}\}$ entspricht der *maximalen Information*, die man über die Folgezustände von A erhält, wenn man als Ausgangsinformation nur den Block von π kennt, in dem sich der gegenwärtige Zustand befindet $([\pi, m(\pi)]$ ist PP$)$.

(ii) $M(\tau) =_{\mathrm{Df}} \sup(\{\pi | [\pi, \tau] \text{ ist PP}\})$ ist die *minimale Information*, die man benötigt, um die Folgezustände in A bis auf $\underset{\tau}{\sim}$ zu bestimmen, d. h. die gröbste aller derjenigen Überdeckungen/Partitionen π, für die $[\pi, \tau]$ PP ist.

Beispiel. Für den unter 1.2 angegebenen Automaten ist mit $\pi = /1, 3, 4 / 2,5/$ und $\tau = /1, 4 / 2, 3, 5/$ $[\pi, \tau]$ ein PP im Verband der Partitionen. Auch mit $\tau > \tau' = /1, 4 / 3, 5 / 2/$ ist $[\pi, \tau']$ PP, wobei $\tau' = m(\pi)$. Umgekehrt ist $M(\tau) = M(\tau') = \pi$.

1.4. Satz. *$\langle \mathcal{L}_U, +, \cdot \rangle$ und $\langle \mathcal{L}_P, +, \cdot \rangle$ bilden mit $\mathcal{L}_U = \{[\pi, \tau] | \pi$ und τ sind Überdeckungen und $[\pi, \tau]$ ist ÜP für $A\}$, $\mathcal{L}_P = \{[\pi, \tau] | \pi$ und τ sind Partitionen und $[\pi, \tau]$ ist PP für $A\}$ sowie*

$$[\pi, \tau] + [\pi', \tau'] =_{\mathrm{Df}} [\pi + \pi', \tau + \tau'] \quad \textit{und}$$

$$[\pi, \tau] \cdot [\pi', \tau'] =_{\mathrm{Df}} [\pi \cdot \pi', \tau \cdot \tau']$$

einen Verband. ▮

1.6. Satz.[1] (i) *Wenn $\pi \geqq \tau$, so $m(\pi) \geqq m(\tau)$ sowie $M(\pi) \geqq M(\tau)$.*

(ii) $m(\pi \cdot \tau) \leqq m(\pi) \cdot m(\tau)$.

(iii) $M(\pi \cdot \tau) = M(\pi) \cdot M(\tau)$.

(iv) $M(m(\pi)) \geqq \pi$. ▮

[1]) Vgl. [1], S. 64 f.

1.7. Definition. $\overline{m}(\pi) =_{\mathrm{Df}} \inf(\{\tau \mid \forall B \exists B' (B \in \pi \wedge B' \in \tau \wedge \forall x (x \in X \to$ $\to f(B, x) \subseteqq B'))\})$. $[\pi, \overline{m}(\pi)]$ ist ein PP, bei dem für jeden Block B aus π der Block B' von τ, in den B übergeht, nicht vom Eingabesignal abhängt, sondern nur von B. Offensichtlich ist $\overline{m}(\pi) \geqq m(\pi)$.

Beispiel. Für den unter 1.2 angegebenen Automaten ist mit $\pi = /1, 3, 4 / 2, 5/$ und $\tau = /1, 3, 4, 5 / 2, 3, 4/$ $\overline{m}(\pi) = \tau$ im Verband der Überdeckungen.

2. Die Theorie der rückgekoppelten Information

Es soll nun eine Einführung in die in [1] dargestellte Feedbacktheorie gegeben werden. Dazu betrachten wir das Schema von Abb. 1, dem jede digitale (synchrone) Schaltung genügt.

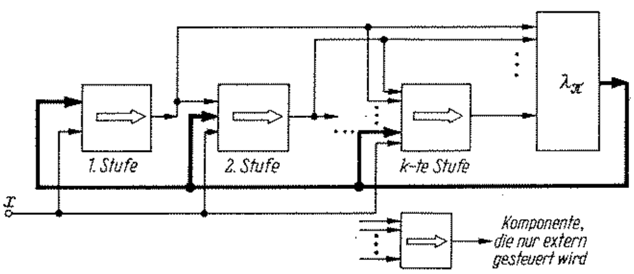

Abb. 1. Feedback-Schema eines Automaten

Der mit λ_π bezeichnete Block realisiert eine Funktion von Z in eine Menge S und induziert somit eine Partition π über Z durch $z \underset{\pi}{\sim} z' \underset{\mathrm{Df}}{\leftrightarrow} \lambda_\pi(z) = \lambda_\pi(z')$. Ist λ_π mehrdeutig, so ist π eine Überdeckung. π heißt Feedback-Partition bzw. -Überdeckung. Die „rückgeführten" Signale s aus S geben an, in welchem Block $B = \lambda_\pi^{-1}(s)$ von π sich der gegenwärtige Zustand der Schaltung befindet.

2.1. Definition. Sei π eine Partition bzw. Überdeckung von Z. Wir definieren den Operator A^i ($i \geqq 1$) rekursiv wie folgt:

$$A^1(\pi) = \pi ;$$

$$A^{i+1}(\pi) = \pi \cdot m(A^i(\pi)) .$$

Sei n die Zahl der Zustände des zugrundegelegten Automaten A. Unter „Partition" wollen wir — falls nicht anders vermerkt — auch Überdeckungen verstehen.

2.2. Satz.[1]) (i) *Wenn* $i \geqq j$, *so* $A^i(\pi) \leqq A^j(\pi)$;

(ii) *Wenn* $j \geqq n$, *so* $A^j(\pi) = A^n(\pi)$;

(iii) *Wenn* $i > j$ *und* $A^j(\pi) = A^{j+1}(\pi)$, *so* $A^j(\pi) = A^i(\pi)$. ▮

2.3. Satz. *π ist genau dann als Feedback-Partition gemäß Abb.* 1 *verwendbar, wenn* $A^n(\pi) = 0$. ▮

Abb. 2 veranschaulicht die Bedeutung des Operators A^i. $A^i(\pi)$ ist die am Eingang der i-ten „Stufe" zur Verfügung stehende Information über den gegen-

[1]) Vgl. [1], S. 148 ff.

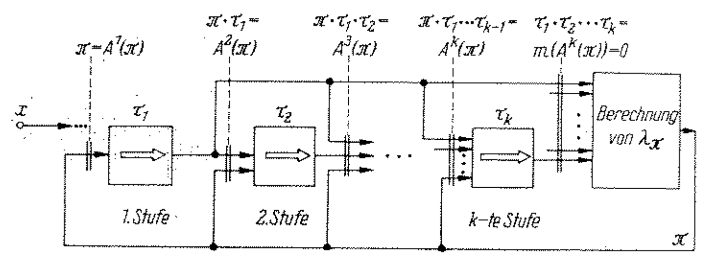

Abb. 2. Veranschaulichung der Bedeutung von $A^i(\pi)$

wärtigen Zustand, der also bis auf $\widetilde{A^i(\pi)}$ bekannt ist. Daraus läßt sich der Folgezustand bis auf $\widetilde{m(A^i(\pi))}$ berechnen, da $[A^i(\pi), m(A^i(\pi))]$ PP ist. Daher kann man der ersten Stufe die *Kodierungspartition* $\tau_1 = m(\pi) = m(A^1(\pi))$ zuordnen. Für $i > 1$ wird $m(A^i(\pi))$ realisiert durch die bereits fixierten Stufen $1, \dots, i-1$ mit den Kodierungspartitionen $\tau_1, \dots, \tau_{i-1}$ und einer neu zu bestimmenden Partition τ_i derart, daß die bisher realisierte Information $m(A^{i-1}(\pi)) = \tau_1 \cdots \tau_{i-1}$ zusammen mit τ_i die realisierbare Information $m(A^i(\pi)) = m(A^{i-1}(\pi)) \cdot \tau_i$ ergibt. π ist dann als rückgekoppelte Information ausreichend, wenn für ein $i < n$ gilt $m(A^i(\pi)) = 0$, d.h. $\prod_{j=1}^{i} \tau_j = 0$. Aus der Bestimmung der τ_j geht hervor, daß mit $[A^j(\pi), m(A^j(\pi))]$ bzw. $[A^j(\pi), \tau_1 \cdots \tau_j]$ stets $[\pi \cdot \tau_1 \cdot \dots \cdot \tau_{j-1}, \tau_j]$ $(1 < j < i)$ PP ist.

Die $j$te Stufe hängt also nur von Eingabesignal x, der rückgekoppelten Information π sowie den Stufen $1, \dots, j-1$ ab.

Bereits in [1], S. 152 ff. wurde durch ein Beispiel darauf hingewiesen, daß die τ_i im allgemeinen noch gröber wählbar sind, so daß also $m(A^{i-1}(\pi)) \cdot \tau_i > m(A^i(\pi))$ gelten kann. In den folgenden Kapiteln werden die Bedingungen ermittelt, die jedes τ_i erfüllen muß, um als Kodierungspartition einen geringen Speicher- und (kombinatorischen) Ansteueraufwand zu ermöglichen, ohne jedoch die Stufenzahl zu vergrößern.

Das folgende Beispiel soll die eben beschriebene Konstruktion der Realisierung verdeutlichen.

Beispiel. $A = [\{0, 1\}, \{1, 2, 3\}, f]$ mit

f	1	2	3
0	2	2	1
1	2	3	3

Die Feedback-Partition sei $\pi = /1 / 2, 3/$.

$m(\pi) = /1, 2 / 3/$ und damit $A^2(\pi) = \pi \cdot m(\pi) = /1 / 2 / 3/ = 0$; $\tau_1 =_{\mathrm{Df}} m(\pi) = /1, 2 / 3/$, $\tau_2 =_{\mathrm{Df}} /1, 3 / 2, 3/$. Die ersten Blöcke von τ_1 und τ_2 werden jeweils mit 0, die zweiten mit 1 kodiert. Bezeichnen wir die Zustandsvariablen ebenso wie die dazugehörigen Partitionen, so ergibt sich:

$$\tau_1 = \bar{x} \vee \bar{\pi}; \qquad \tau_2 = \bar{\tau}_1; \qquad \pi = \overline{\bar{\tau}_1 \cdot \bar{\tau}_2} = \tau_1 \vee \tau_2 .$$

Da $\tau_2 = \overline{m}(\tau_1)$ ist, hängt τ_2 nicht vom Eingabesignal ab. Die Schaltung gibt Abb. 3 wieder.

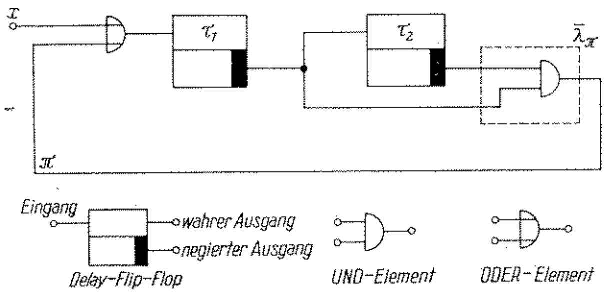

Abb. 3. Beispiel für die Kodierung eines endlichen Automaten mit Hilfe des Feedback-Formalismus

3. Aufbereitung der Feedback-Theorie für die Kodierung der Zustände

3.1. Der Begriff des Quotienten zweier Überdeckungen

Die Definition des Quotienten führt in den Verband der Überdeckungen. Wir gehen daher von vornherein von solchen aus. Sind π und τ zwei Überdeckungen einer Menge Z mit $\tau \geqq \pi$, so besitzt die Gleichung $\tau \cdot \varrho = \pi$ i. a. mehrere Lösungen. Mit ϱ_1 und ϱ_2 sind auch $\varrho_1 \cdot \varrho_2$ und $\varrho_1 + \varrho_2$ Lösungen (Operationen für Überdeckungen!). Die Menge $Q(\pi/\tau) =_{\mathrm{Df}} \{\varrho | \tau \cdot \varrho = \pi\}$ für $\tau \geqq \pi$ ist deshalb abgeschlossen in bezug auf die Multiplikation $\cdot$ und Addition $+$. Folglich existieren für jede nichtleere Teilmenge $\mathfrak{D}$ von $Q(\pi/\tau)$ das Supremum $\sup(\mathfrak{D})$ und Infimum $\inf(\mathfrak{D})$. $Q(\pi/\tau)$ ist ein Unterverband des Verbandes aller Überdeckungen von Z mit $\sup(Q(\pi/\tau)) = \sum_{\varrho \in Q(\pi/\tau)} \varrho$ und $\inf(Q(\pi/\tau)) = \pi$. Wir nennen $\pi/\tau =_{\mathrm{Df}} \sup(Q(\pi/\tau))$ *den Quotienten* von π und τ.[1])

Wir wollen in diesem Abschnitt untersuchen, welche Eigenschaften der Quotient zweier *beliebiger* Überdeckungen π und τ besitzt, nachdem dieser Begriff unter 3.2 eingeführt wurde. Z sei eine endliche Menge.

3.1. Definition. Sei $R \subseteqq Z \times Z$ eine binäre symmetrische und reflexive Relation (Verträglichkeitsrelation) über Z. Dann ist

$$[Z/R] =_{\mathrm{Df}} \max(\{M | z, z' \in M \rightarrow [z, z'] \in R\})$$

die durch R induzierte Überdeckung von Z.

3.2. Definition. Seien π und τ beliebige Überdeckungen von Z. Dann heißt

$$\frac{\pi}{\tau} =_{\mathrm{Df}} [Z/\{[z, z'] | z, z' \in Z \wedge z \underset{\pi}{\sim} z' \vee z \underset{\tau}{\not\sim} z'\}]$$

der Quotient von π und τ.

3.3. Satz. *Für beliebige π, τ ist $(\pi/\tau) \cdot \tau = \pi \cdot \tau \leqq \pi$.*

Beweis. Mit $\varrho =_{\mathrm{Df}} \pi/\tau$ gilt

$$\forall z \forall z' \left(z, z' \in Z \rightarrow z \underset{\varrho \cdot \tau}{\sim} z' \leftrightarrow \left(\left(z \underset{\pi}{\sim} z' \vee z \underset{\tau}{\not\sim} z'\right) \wedge z \underset{\tau}{\sim} z'\right) \leftrightarrow z \underset{\pi \cdot \tau}{\sim} z'\right) . \blacksquare$$

3.4. Folgerung. *$\tau \geqq \pi$ genau dann, wenn $(\pi/\tau) \cdot \tau = \pi$.* $\blacksquare$

---

[1]) Aus drucktechnischen Gründen wird oft π/τ statt $\frac{\pi}{\tau}$ gesetzt.

3.5. Satz. *$\varrho \cdot \tau = \pi$ genau dann, wenn $\pi \leqq \tau$ und $\varrho \in Q(\pi/\tau)$.*

Beweis. Aus $\pi \overset{3.4.}{=} (\pi/\tau) \cdot \tau \geqq \varrho \cdot \tau \geqq \pi \cdot \tau = \pi$ folgt die eine Richtung, die andere ebenso einfach aus 3.2. ■

3.6. Folgerung.

(i) $\dfrac{\pi \cdot \tau}{\tau} = \dfrac{\pi}{\tau}$.

(ii) $\dfrac{\pi}{\tau} \cdot \tau \leqq \pi \leqq \dfrac{\pi \cdot \tau}{\tau}$.

(iii) *Wenn* $\pi \leqq \tau$, *so* $\dfrac{\pi \cdot \tau}{\tau} = \dfrac{\pi}{\tau} \cdot \tau = \pi$.

Der Beweis ergibt sich aus 3.2, 3.3 und 3.4. ■

3.7. Satz. *Für alle π, τ, ϱ gilt*

$$\frac{\frac{\pi}{\tau}}{\varrho} = \frac{\tau}{\varrho \cdot \tau}.$$

Beweis. $\forall z \forall z' \big(z, z' \in Z \to z \underset{(\pi/\tau)/\varrho}{\sim} z' \leftrightarrow (z \underset{\pi/\tau}{\sim} z' \vee z \underset{\varrho}{\not\sim} z') \leftrightarrow$

$$\leftrightarrow (z \underset{\pi}{\sim} z' \vee z \underset{\tau}{\not\sim} z' \vee z \underset{\varrho}{\not\sim} z') \leftrightarrow (z \underset{\pi}{\sim} z' \vee z \underset{\varrho \cdot \tau}{\not\sim} z') \to z \underset{\pi/(\varrho \cdot \tau)}{\sim} z'\big). \ ■$$

3.8. Satz. *$\varrho \cdot \tau \leqq \pi$ genau dann, wenn $\varrho \leqq \pi/\tau$.*

Beweis. klar. ■

3.9. Beispiel. $\pi = /1 \,/\, 2 \,/\, 3 \,/\, 4, 6 \,/\, 5 \,/\, 7/, \tau = /1, 2, 3 \,/\, 4, 6 \,/\, 5, 7/$.
Wegen

$$\underset{\pi/\tau}{\sim} = \{[4, 6], [1, 4], [1, 5], [1, 6], [1, 7], [2, 4], [2, 5], [2, 6], [2, 7], [3, 4], [3, 5], [3, 6], [3, 7], [4, 5], [4, 7], [5, 6], [6, 7]\}$$

ist $\pi/\tau = /2, 4, 5, 6 \,/\, 1, 4, 5, 6 \,/\, 1, 4, 6, 7 \,/\, 2, 4, 6, 7 \,/\, 3, 4, 5, 6 \,/\, 3, 4, 6, 7/$;

$$(\pi/\tau) \cdot \tau = \max(\{\{2\}, \{1\}, \{1\}, \{2\}, \{3\}, \{3\}, \{4, 6\}, \{4, 6\}, \ldots, \cdots, \{4, 6\}, \{5\}, \{5\}, \{7\}, \{7\}, \{5\}, \{7\}\}) = \pi.$$

Mit $\varrho_1 = /1, 4, 5, 6 \,/\, 2, 4, 5, 6 \,/\, 3, 4, 6, 7/ \subseteqq \pi/\tau$ ist $\pi \leqq \varrho_1 \leqq \pi/\tau$ und $\varrho_1 \cdot \tau = \pi$. Das Gleiche gilt für $\varrho_2 = /2, 4, 6 \,/\, 1, 5 \,/\, 3, 7/ \leqq \pi/\tau$.

3.2. Anwendung des Quotientenbegriffs auf den Feedback-Formalismus

3.2.1. Abschwächung der Bedingungen für die Kodierungspartitionen

Die Ermittlung der Partitionen/Überdeckungen $A^i(\pi)$ $(1 \leqq i \leqq n-1)$ geschieht für einen Automaten A durch sukzessive Anwendung des Operators m und der Multiplikation mit π. In jeder Stufe i wird die bisher realisierte Information $\tau_1 \cdot \cdots \cdot \tau_{i-1}$ vervollständigt durch Hinzunahme eines neuen τ_i derart, daß $\tau_1 \cdots \tau_{i-1} \cdot \tau_i = m(A^i(\pi))$ ist. Nun ist für jede Überdeckung ϱ $M(m(\varrho)) \geqq \varrho$, d. h., zur Berechnung von $m(\varrho)$ genügt $M(m(\varrho))$, da $m(M(m(\varrho))) = m(\varrho)$. Das bedeutet für die Wahl der τ_i, daß die Partition/Überdeckung $\pi \cdot \tau_1 \cdot \cdots \cdot \tau_{i-1}$ für die Ermittlung der Information $m(A^i(\pi)) = m(\pi \cdot \tau_1 \cdot \dots \cdot \tau_{i-1})$ möglicherweise ersetzt werden kann durch eine gröbere mit $M(m(A^i(\pi))) \geqq \pi \cdot \tau_1 \cdot \dots \cdot \tau_{i-1}$. Daraus folgt aber, daß die τ_j ($j \leqq i-1$) evtl. gröber wählbar sind, d. h., der Speicheraufwand der j-ten Stufe kann herabgesetzt werden. Darüberhinaus hat man in der Wahl der τ_j durch die obige Ungleichung einen größeren Spielraum, der für eine solche Festlegung der τ_j ausgenutzt werden kann, die eine minimale Abhängigkeit von den vorigen Stufen, von π und vom externen Eingabesignal x garantiert. Das Schema von Abb. 4 soll verdeutlichen, daß

$$
\begin{tikzcd}
\pi = A^1(\pi) \arrow[d, Rightarrow] & & & & & & \\
m(\pi) \arrow[d, Rightarrow] & \leqq & \frac{Mm(\pi \cdot m(\pi))}{\pi} & \leqq & \frac{M\left(\frac{Mm(A^3(\pi))}{\pi}\right)}{\pi} & = & N^2(m(A^3(\pi))) \\
A^2(\pi) \arrow[d, Rightarrow] & \leqq & Mm(\pi \cdot m(\pi)) \arrow[u, Rightarrow] & & M\left(\frac{Mm(A^3(\pi))}{\pi}\right) \arrow[u, Rightarrow] & & \\
m(A^2(\pi)) \arrow[d, Rightarrow] \arrow[urr, Rightarrow] & \leqq & \frac{Mm(A^3(\pi))}{\pi} \arrow[urr, Rightarrow] & = & N^1(m(A^3(\pi))) & & \\
A^3(\pi) \arrow[d, Rightarrow] & \leqq & Mm(A^3(\pi)) \arrow[u, Rightarrow] & & & & \\
m(A^3(\pi)) \arrow[urr, Rightarrow] & & & & & &
\end{tikzcd}
$$

Abb. 4. Veranschaulichung der Ausnutzung des M-Operators und der Divisionsoperation für die Herabsetzung des notwendigen Speicheraufwandes pro Stufe

durch die Wahl der τ_i innerhalb der angegebenen Grenzen unter Beibehaltung der Stufenzahl kein Informationsverlust auftritt, und in welchem Zusammenhang die im vorigen Abschnitt definierte Divisionsoperation Verwendung findet. Der Begriff „Partition" soll auch hier Überdeckungen mit einschließen.

3.10. Definition. Seien π eine fixierte und ϱ eine beliebige Partition sowie i eine ganze Zahl. Wir definieren den Operator N^i für π rekursiv wie folgt

$$
N^i(\varrho) = \begin{cases} \varrho\,, & \text{falls } i = 0\,, \\ \dfrac{M(N^{i-1}(\varrho))}{\pi}\,, & \text{falls } i > 0\,, \\ m(\pi \cdot N^{i+1}(\varrho))\,, & \text{falls } i < 0\,. \end{cases}
$$

3.11. Bemerkung. Für $i > 0$ ist stets $N^i(\varrho) = N(N^{i-1}(\varrho)) = N^{i-1}(N(\varrho))$. ∎

3.12. Notation. Sind $\tau_1, \dots, \tau_{i-1}$ die bei der stufenweisen Konstruktion der Schaltung bereits fixierten Kodierungspartitionen für die 1. bis $i-1$. Stufe, so setzen wir für $i \geqq 1$

$$
B^i(\pi) = \pi \cdot \prod_{j=1}^{i-1} \tau_j\,.
$$

$B^i(\pi)$ ist also die an der i-ten Stufe tatsächlich vorhandene Information.

3.13. Satz. *Für $i, k \geqq 1$ und alle π gilt $m(A^i(\pi)) \leqq m(B^i(\pi)) \leqq N^{k-i}(m(A^k(\pi)))$, solange für alle τ_j $(j = 1, \ldots, i-1)$ $m(B^j(\pi)) \leqq \tau_j \leqq \dfrac{N^{k-j}(m(A^k(\pi)))}{1 \cdot \tau_1 \cdot \cdots \cdot \tau_{j-1}}$ richtig ist.*

Beweis. Wir zeigen $m(A^i(\pi)) \leqq m(B^i(\pi))$ durch Induktion über i: Für $i = 1$ ist $m(A^1(\pi)) = m(B^1(\pi)) = m(\pi)$. Sei die Ungleichung für $i = l$ gültig. Dann folgt wegen $\tau_l \geqq m(B^l(\pi))$

$$m(B^{l+1}(\pi)) = m(\pi \cdot \tau_1 \cdot \cdots \cdot \tau_l) \geqq m(\pi \cdot \tau_1 \cdot \cdots \cdot \tau_{l-1}\, m(B^l(\pi))) =$$
$$= m(\pi \cdot m(B^l(\pi))) \geqq m(\pi \cdot m(A^l(\pi))) = m(A^{l+1}(\pi))\,.$$

Mit

$$m(B^i(\pi)) \leqq \tau_1 \cdot \cdots \cdot \tau_{i-1} \cdot \tau_i \leqq \tau_1 \cdot \cdots \cdot \tau_{i-1} \frac{N^{k-i}(m(A^k(\pi)))}{1 \cdot \tau_1 \cdot \cdots \cdot \tau_{i-1}}$$

und 3.3 ergibt sich der Rest der Behauptung. ▮

3.14. Folgerung. (i) *Für $i \geqq k$ gilt*

$$m(A^i(\pi)) = m(B^i(\pi)) = N^{k-i}(m(A^k(\pi)))\,.$$

(ii) *Werden die τ_i wie unter 3.13 bestimmt, so ist für $i \leqq n-1$ $[\pi \cdot \tau_1 \cdots \tau_{i-1}, \tau_i]$ PP.* ▮

Für $k \geqq i$ erhält man wegen

$$N^{k+1-i}(m(A^{k+1}(\pi))) = N^{k-i}\left(\frac{Mm \cdot (A^{k+1}(\pi))}{\pi}\right) \overset{1.6.(\mathrm{iv})}{\geqq}$$
$$\geqq N^{k-i}\left(\frac{A^{k+1}(\pi)}{\pi}\right) = N^{k-i}\left(\frac{\pi \cdot m(A^k(\pi))}{\pi}\right) \overset{3.6.(\mathrm{ii})}{\geqq} N^{k-i}(m(A^k(\pi)))$$

den größten Spielraum für die Wahl der τ_i, wenn $m(A^k(\pi)) = 0$ ist.

3.2.2. Über die Auswahl der Kodierungspartitionen

Im vorigen Abschnitt wurde bestimmt, wie grob die Kodierungspartitionen τ_i maximal sein können, ohne die Zahl der Schaltungsstufen zu vergrößern. Es folgen einige Betrachtungen darüber, wie der angegebene Spielraum für eine möglichst gute Auswahl der τ_i genutzt werden kann. Zwei Kriterien sind dabei zu berücksichtigen. (i) Die den τ_i zugeordneten Speicher sollen von möglichst wenig anderen Speichern abhängen; (ii) Jede Ebene soll möglichst wenige Speicher enthalten.

Im allgemeinen sind die τ_i keine Binärpartitionen. Besitzen sie mehrere Blöcke, so sind sie in eine möglichst kleine Zahl von Binärpartitionen $\tau_{i_1}, \ldots, \tau_{i_{s_i}}$ „aufzulösen“.[1])

[1]) Man beachte, daß bei Überdeckungen, die *keine* Zerlegungen sind, den Blöcken nicht ohne weiteres eine beliebige Kodierung zuweisbar ist. Z. B. ist es falsch, die Blöcke von $\tau = /1, 2, 3/2, 3, 4, 6/2, 3, 5, 6/$ mit 00, 01 bzw. 10 zu belegen, da das Produkt $\tau_1 \cdot \tau_2 = /1, 2, 3, 6/2, 3, 4, 6/2, 3, 5, 6/$ der dadurch induzierten Binärpartitionen $\tau_1 = /1, 2, 3, 4, 6/2, 3, 5, 6/$ und $\tau_2 = /1, 2, 3, 5, 6/2, 3, 4, 6/$ *gröber* als τ ist. Wenn bereits eine Binärpartition τ_1 festliegt (bzw. τ_1 das Produkt von bereits ermittelten Partitionen ist), so muß natürlich τ_2 aus $Q(\tau/\tau_1)$ sein. Für das Beispiel ist $\tau_2 =_{\mathrm{Df}} /1, 2, 3/2, 3, 4, 5, 6/ < /1, 2, 3, 5/2, 3, 4, 5, 6/ \leqq \tau/\tau_1$ richtig.

Seien $\pi_1, \dots, \pi_p$ die der Kodierung der Feedbackpartition π entsprechenden Binärpartitionen mit $\prod_{i=1}^{p} \pi_i \leqq \pi$ und sei N_k die Zahl der Elemente einer Menge $\mathfrak{N}_k$. Weiterhin sei $\mathfrak{N}_{i_\sigma} \subseteqq \{\pi_1, \dots, \pi_p, \tau_{1_1}, \dots, \tau_{1_{s_1}}, \tau_{2_1}, \dots, \tau_{i-1_{s_{i-1}}}\}$ so gewählt, daß $\left[\prod_{\tau \in \mathfrak{N}_{i_\sigma}} \tau, \tau_{i_\sigma}\right]$ PP ist für $\sigma = 1, \dots, s_i$. $\{\tau_{i_1}, \dots, \tau_{i_\sigma}, \dots, \tau_{i_s}\}$ heiße *relativ-optimal* in bezug auf $\{\pi_1, \dots, \pi_p, \tau_{1_1}, \dots, \tau_{i-1_{s_{i-1}}}\}$, wenn $\sum_{\sigma=1}^{s_i} \min(N_{i_\sigma})$ relativ zu jeder anderen Festlegung $\{\tau'_{i_1}, \dots, \tau'_{i_{s_i}}\}$ minimal ist. Die Bestimmung der τ_{i_σ} kann dann nach folgendem Schema erfolgen: Für jedes i $(1 \leqq i \leqq k)$ ist eine Menge $\mathfrak{N}_i \subseteqq \{\pi_1, \dots, \pi_p, \tau_{1_1}, \dots, \tau_{i-1_{s_{i-1}}}\}$ mit einer minimalen Anzahl von Partitionen derart zu finden, daß $m\left(\prod_{\tau \in \mathfrak{N}_i} \tau\right) \leqq \dfrac{N^{k-i}\left(m(A^k(\pi))\right)}{1 \cdot \tau_1 \cdot \dots \cdot \tau_{i-1}}$. Existiert keine *Binärpartition* τ_i mit $m\left(\prod_{\tau \in \mathfrak{N}_i} \tau\right) \leqq \tau_i \leqq \dfrac{N^{k-i}\left(m(A^k(\pi))\right)}{1 \cdot \tau_1 \cdot \dots \cdot \tau_{i-1}}$, so sind $s_i = \text{entier}\ [\text{ld}(\#(\tau_i))]$ Binärpartitionen $\tau_{i_1}, \dots, \tau_{i_{s_i}}$ aufzusuchen mit den Eigenschaften:

(i) $\prod_{\sigma=1}^{s_i} \tau_{i_\sigma} \leqq \dfrac{N^{k-i}\left(m(A^k(\pi))\right)}{1 \cdot \tau_1 \cdot \dots \cdot \tau_{i-1}}$;

(ii) die Teilmengen $\mathfrak{N}_{i_\sigma}$ $(\sigma = 1, \dots, s_i)$ von $\mathfrak{N}_i$ mit $\overline{m}\left(\prod_{\tau \in \mathfrak{N}_{i_\sigma}} \tau\right) \leqq \tau_{i_\sigma}$ (Unabhängigkeit vom Eingabesignal x) *oder* $m\left(\prod_{\tau \in \mathfrak{N}'_{i_\sigma}} \tau\right) \leqq \tau_{i_\sigma}$ enthalten eine minimale Anzahl von Binärpartitionen.[1])

Ein weiteres Auswahlkriterium ist die Einfachheit der Funktionen, die $\pi_1, \dots, \pi_p$ realisieren.

Sind alle τ_{i_σ} in bezug auf $\pi_1, \dots, \pi_p, \tau_{1_1}, \dots, \tau_{i-1_{s_i-1}}$ relativ-optimal gewählt, so folgt daraus noch nicht, daß die Gesamtkodierung optimal oder quasioptimal ist. Einerseits beeinflußt die Kodierung der Feedbackpartition π, d. h. die Festlegung von $\pi_1, \dots, \pi_p$, namentlich bei einer höheren Zahl von Blöcken sehr wesentlich den Aufwand. Zum anderen kommt es vor (vgl. Abschn. 3.3), daß gerade die nicht-relativ-optimale Wahl eines (oder mehrerer) τ_{k_σ} günstigere Festlegungen anderer Partitionen τ_i mit $i > k$ erlaubt. Weiter kann eine *Vergrößerung der Stufenzahl* der Schaltung dann von Vorteil sein, wenn es Stufen mit mehr als einem Binärspeicher gibt (entsprechend einem τ_i mit mehr als zwei Blöcken). Erlaubt man die Vergrößerung der Stufenzahl k um r, und ist i eine fixierte Stufe mit $\#(\tau_i) > 2$, so sind die Kodierungspartitionen $\tau'_{i-r+1}, \dots, \tau'_i$ noch gröber als die ursprünglichen $\tau_{i-r+1}, \dots, \tau_i$ wählbar, wobei folgende Einschränkungen gelten:

(i) $m\left(B^{i-r+j}(\pi)\right) \leqq \tau'_{i-r+j}$;

(ii) $\prod_{l=1}^{r} \tau'_{i-r+l} \leqq \dfrac{N^{k-(i-r)}\left(m(A^k(\pi))\right)}{1 \cdot \tau_1 \cdot \dots \cdot \tau_{i-r}}$

für $1 \leqq j \leqq r$. Dann ist (vgl. Beweisführung 3.13)

$$m\left(B^{i-r+j}(\pi)\right) \leqq N^{k-(i-r)}\left(m(A^k(\pi))\right),$$

---

[1]) Die Abhängigkeit von τ_{i_σ} von einer minimalen Anzahl von Variablen garantiert noch keinen minimalen Aufwand für die Realisierung der Ansteuergleichungen von τ_{i_σ}. Beispiel: $z_1 z_2 \vee \bar{z}_2 z_3$ erfordert einen höheren Aufwand als $z_1 z_2 \bar{z}_3 z_4$.

d. h., bis zur i-ten Stufe ist nur soviel Information fixiert worden, daß für die vollständige Realisierung eine Höchstzahl von weiteren $k - i + r$ Stufen garantiert ist (vgl. Abschnitt 3.3).

3.3. Beispieldiskussion

Gegeben sei der Halbautomat $A = [\{0, 1\}, \{1, 2, \ldots, 10\}, f]$ mit

f	1	2	3	4	5	6	7	8	9	10
0	10	10	10	10	10	10	7	7	7	9
1	5	5	6	5	6	6	1	2	4	3

Er soll über die Feedbackpartition $\pi = /3, 9 \,/\, 1, 2, 4, 5, 6, 7, 8, 10 /$ realisiert werden.

Es ist:

$$m(\pi) = /7, 9, 10 \,/\, 4, 6 \,/\, 1, 2, 3, 5, 6 \,/\, 8/;$$

$$A^2(\pi) = /7, 10 \,/\, 9 \,/\, 4, 6 \,/\, 1, 2, 5, 6 \,/\, 8/;$$

$$m(A^2(\pi)) = /1, 3 \,/\, 2 \,/\, 4 \,/\, 5, 6 \,/\, 7, 9 \,/\, 8 \,/\, 10/;$$

$$A^3(\pi) = /1 \,/\, 2 \,/\, 3 \,/\, 4 \,/\, 5, 6 \,/\, 7 \,/\, 8 \,/\, 9 \,/\, 10/;$$

$$m(A^3(\pi)) = 0;$$

$$A^4(\pi) = 0;$$

$$M(0) = /1, 2, 4 \,/\, 3, 5, 6 \,/\, 7 \,/\, 8 \,/\, 9 \,/\, 10/;$$

$$N^1(0) = \frac{M(0)}{\pi} = /1, 2, 3, 4 \,/\, 1, 2, 4, 9 \,/\, 3, 5, 6 \,/\, 3, 7 \,/\, 3, 8 \,/\, 3, 10 \,/\, 5, 6, 9 \,/\, 7, 9 \,/\, 9, 10 \,/\, 8, 9/;$$

$$M(N^1(0)) = /1-6, 10 \,/\, 7-10/;$$

$$N^2(0) = /1-6, 10 \,/\, 7-10 \,/\, 3, 7, 8 \,/\, 1, 2, 4, 5, 6, 9/.$$

a) Eine konventionelle Lösung.

$$\tau_1 = m(\pi) = /7, 9, 10 \,/\, 4, 6 \,/\, 1, 2, 3, 5, 6 \,/\, 8/, \quad \text{Kodierung: } /00 \,/\, 01 \,/\, 11 \,/\, 10/;$$

$$\tau_{11} = /4, 6, 7, 9, 10 \,/\, 1, 2, 3, 5, 6, 8/,$$

$$\tau_{12} = /7, 8, 9, 10 \,/\, 1-6/;$$

$$\tau_2 = /1, 3, 7, 9 \,/\, 5, 6, 10 \,/\, 2, 4, 8/, \quad \text{Kodierung: } /00 \,/\, 10 \,/\, 10/;$$

$$\tau_{21} = /1, 3, 5, 6, 7, 9, 10 \,/\, 2, 4, 8/,$$

$$\tau_{22} = /5, 6, 10 \,/\, 1-4, 7, 8, 9/;$$

$$\tau_3 = /1, 2, 4, 5, 7, 8, 10 \,/\, 3, 6, 9/, \quad \text{Kodierung: } /0 \,/1/;$$

$$\tau_{22} \cdot \tau_3 < \pi = /3, 9 \,/\, 1, 2, 4-8, 10/, \quad \text{Kodierung: } /01 \,/\, 00 \vee 1-/.$$

Daraus ergeben sich für die fünf Speicher (D-Flip-Flops) folgende Ansteuergleichungen:

$$\tau_{11} = \bar{\tau}_{22}\tau_3 \vee \bar{x}; \quad \tau_{12} = \bar{x};$$

$$\tau_{21} = x \cdot \tau_{11}\bar{\tau}_{12} \vee x\bar{\tau}_{22}\bar{\tau}_{12}; \; \tau_{22} = \tau_{12};$$

$$\tau_3 = \bar{\tau}_{12}\tau_{22} \vee x\tau_{22} \vee x\bar{\tau}_{22}\tau_3\tau_{11}\tau_{21};$$

(Vgl. Abb. 5a).

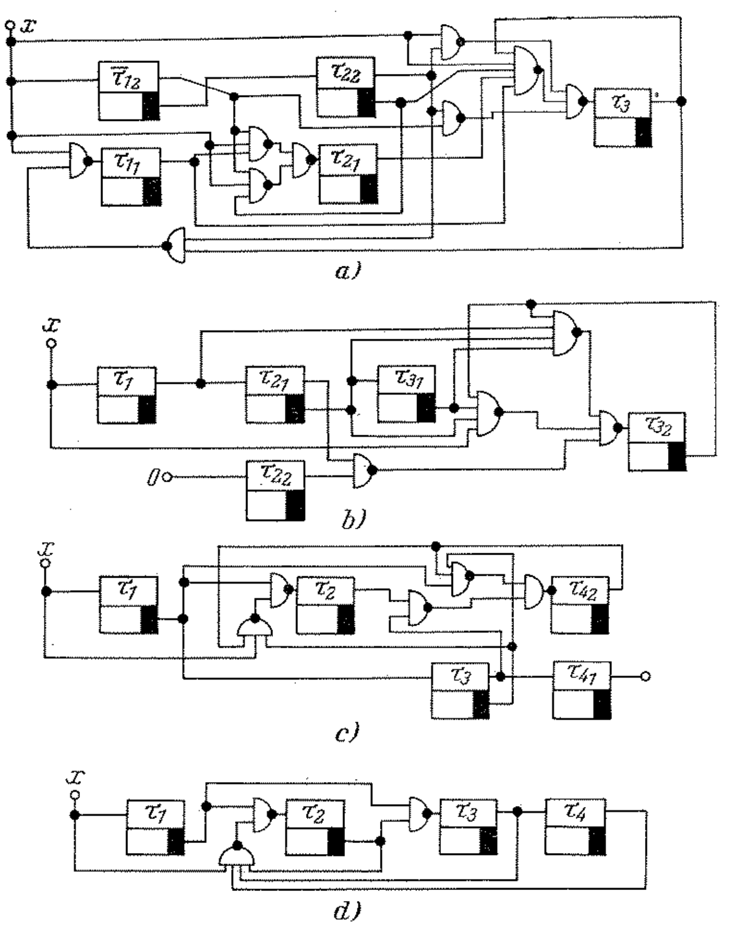

Abb. 5. Verschiedene Feedback-orientierte Lösungen des Kodierungsproblems für einen Automaten (Beispiel unter 3.3). Als logische Glieder wurden NAND-Gatter verwendet. (a) eine Lösung ohne Ausnutzung der abgeschwächten Bedingungen; (b) Lösung bei Anwendung des beschriebenen Verfahrens; (c) Einfügen einer weiteren Ebene (τ_2); (d) durch eine nicht-relativ-optimale Wahl von τ_3 fällt in der 4. Ebene ein Flip-Flop-Speicher weg (τ_{42})

b) Lösung nach Wahl der τ_i gemäß Abschnitt 3.2.2.

$$\tau_1 = /1-6 \,/\, 7-10/, \text{ Kodierung: } /1 \,/\, 0/;$$

$$B^2(\pi) = /1, 2, 4, 5, 6 \,/\, 3 \,/\, 7, 8, 10 \,/\, 9/;$$

$$m(B^2(\pi)) = /7, 9 \,/\, 10 \,/\, 1, 2, 3 \,/\, 4 \,/\, 5, 6 \,/\, 8/;$$

$$\frac{N^1(0)}{\tau_1} = /1-4, 7, 9 \,/\, 1-4, 9, 10 \,/\, 1-4, 8, 9 \,/$$
$$/3, 5, 6, 7, 9 \,/\, 3, 5, 6, 9, 10 \,/\, 3, 5, 6, 8, 9/$$

$$\geqq \tau_2 = \overline{m}(\tau_1) = /5, 6, 10 \,/\, 1-4, 7, 9 \,/\, 8/,$$
$$\text{Kodierung: } /11 \,/\, 01 \,/\, 10/;$$

$$\tau_{21} = /5, 6, 8, 10 \,/\, 1-4, 7, 9/;$$

$$\tau_{22} = /8 \,/\, 1-7, 9, 10/;$$

$$\tau_1 \cdot \tau_2 = /5, 6 \,/\, 8 \,/\, 10 \,/\, 1-4 \,/\, 7, 9/;$$

$$B^3(\pi) = \pi \cdot \tau_1 \cdot \tau_2 = /5, 6 \,/\, 10 \,/\, 3 \,/\, 9 \,/\, 1, 2, 4 \,/\, 7 \,/\, 8/;$$

$$m(B^3(\pi)) = 0;$$

$$\overline{m}(\tau_{21}) \leqq \tau_{31} = /2, 3, 6-10 \,/\, 1, 4-7, 10/, \text{ Kodierung: } /0 \,/\, 1/;$$

$$\tau_{32} = /3, 4, 6, 9, 10 \,/\, 1, 2, 5, 7, 8, 10/,$$
$$\text{Kodierung: } /0 \,/\, 1/;$$

$$\tau_3 = /3, 9, 10 \,/\, 4, 6, 10 \,/\, 2, 5, 7, 8, 10 \,/\, 1, 5, 7, 10/;$$

$$\tau_{21} \cdot \tau_{31} \cdot \tau_{32} < \pi = /3, 9 \,/\, 1, 2, 4-8, 10 \,/, \text{ Kodierung: } /000 \,/\, \overline{000}/.$$

Daraus ergibt sich:

$$\tau_1 = x; \quad \tau_{21} = \tau_1; \quad \tau_{22} = 1;$$

(τ_{22} repräsentiert den Zustand 8; der unerreichbar ist);

$$\tau_{31} = \overline{\tau}_{21};$$

$$\tau_{32} = \tau_{21}\tau_{22} \vee \tau_1\overline{\tau}_{21}\overline{\tau}_{31}\overline{\tau}_{32} \vee x\overline{\tau}_{21}\overline{\tau}_{31}\overline{\tau}_{32};$$

(Abb. 5b).

c) Modifikation durch Vergrößerung der Stufenzahl

Wir setzen $\tau_2 = /1, 2, 3, 7, 8, 9 \,/\, 4, 5, 6, 10/$, Kodierung: $/0 \,/\, 1/$;
Damit ist $m(B^2(\pi)) = m(\pi \cdot \tau_1) \leqq \tau_2$, $\tau_1 \cdot \tau_2 \leqq N^2(0)$, und es sind höchstens noch weitere zwei Stufen erforderlich.

$$\pi \cdot \tau_1 \cdot \tau_2 = /3 \,/\, 1, 2 \,/\, 4, 5, 6 \,/\, 7, 8 \,/\, 9 \,/\, 10/;$$

$$\frac{N^1(0)}{\tau_1 \cdot \tau_2} \geqq /1-4, 7, 9, 10 \,/\, 1-4, 8, 9 \,/\, 3, 5, 6, 7, 9, 10 \,/$$
$$/3, 5, 6, 8, 9, 10/;$$

$$\overline{m}(\tau_1) \leqq \tau_3 = /1-4, 7, 9 \,/\, 5, 6, 8, 10 \,/, \text{ Kodierung: } /0 \,/\, 1/;$$

$$m(\tau_1 \cdot \tau_2 \cdot \tau_3 \cdot \pi) = 0;$$

$$\tau_1 \cdot \tau_2 \cdot \tau_3 = /1, 2, 3 \,/\, 4 \,/\, 5, 6 \,/\, 7, 9 \,/\, 8 \,/\, 10/;$$

$$\tau_4 = /1, 4, 5, 7, 8, 10 \,/\, 2, 7, 8, 10 \,/\, 3, 6, 9, 10/\,,$$
$$\text{Kodierung: } /00 \,/\, 10 \,/\, 11/;$$
$$\overline{m}(\tau_3) \leqq \tau_{41} = /1, 4, 5, 6, 7, 8, 10 \,/\, 2, 3, 6, 7-10/, \text{Kodierung: } /0 \,/\, 1/;$$
$$\tau_{42} = /1, 2, 4, 5, 7, 8, 10 \,/\, 3, 6, 9, 10/\,, \text{ Kodierung: } /0 \,/\, 1/;$$
$$\tau_3 \cdot \tau_{42} \leqq \pi\,, \text{ Kodierung: } /01 \,/\, 00 \vee 1-/\,.$$

Ansteuergleichungen:

$$\tau_1 = x; \quad \tau_2 = \tau_1 \vee x\bar{\tau}_3\tau_{42}; \quad \tau_3 = \bar{\tau}_1;$$
$$\tau_{41} = \tau_3; \quad \tau_{42} = \tau_1\bar{\tau}_3\tau_{42} \vee \tau_2\tau_3;$$

(Abb. 5c).

d) Auswirkungen einer nicht-relativ-optimalen Wahl von τ_3.

Ausgehend von (c) setzen wir $\tau_3 = /1, 2, 4, 7 \,/\, 3, 5, 6, 8, 9, 10/$, Kodierung: $/0 \,/\, 1/$;

$$\overline{m}(\tau_1 \cdot \tau_2) = /5, 6, 10 \,/\, 1, 2, 4, 7 \,/\, 3, 9 \,/\, 8/ \leqq \tau_3 \leqq \frac{N^1(0)}{\tau_1 \cdot \tau_2}.$$

Dadurch wird erreicht, daß $\tau_1 \cdot \tau_2 \cdot \tau_3$ feiner als das entsprechende Produkt in (c) ist:

$$\tau_1 \cdot \tau_2 \cdot \tau_3 = /1, 2 \,/\, 3 \,/\, 7 \,/\, 8, 9 \,/\, 4 \,/\, 5, 6 \,/\, 10/;$$
$$m(\pi \cdot \tau_1 \cdot \tau_2 \cdot \tau_3) = 0;$$

τ_4 kann nun als *Binär*partition festgelegt werden:

$$\overline{m}(\tau_3) \leqq \tau_4 = /1, 5, 7, 8, 10 \,/\, 2-4, 6, 7, 9, 10/\,, \text{ Kodierung: } /0 \,/\, 1/;$$
$$\tau_2 \cdot \tau_3 \cdot \tau_4 < \pi, \text{ Kodierung: } /011 \,/\, \overline{011}/\,.$$

Ansteuergleichungen:

$$\tau_1 = x; \quad \tau_2 = \tau_1 \vee x\bar{\tau}_2\tau_3\tau_{42}; \quad \tau_3 = \tau_1 \vee \bar{\tau}_2; \quad \tau_4 = \tau_3;$$

(Abb. 5d).

Das Beispiel zeigt, daß der kombinatorische Aufwand bei Benutzung des beschriebenen Verfahrens (b) wesentlich gegenüber dem für das klassische (a) gesenkt werden kann. Die in 3.2.2 angedeutete Vergrößerung der Stufenzahl (c) bringt hier nur in Verbindung mit einer nicht-relativ-optimalen Wahl von τ_3 (d) erheblichen Vorteil, da die Einfügung der ,,Zwischenebene'' τ_2 und die angegebene Festlegung von τ_3 die Wahl eines sehr einfachen τ_4 ermöglicht (nur zwei Blöcke gegenüber (c), τ_{42} in (c) fällt ganz weg).

4. Diskussion

Die Anwendung eines Feedback-orientierten Kodierungsalgorithmus ist natürlich nur dann sinnvoll, wenn entweder aufgrund der Problemstellung eine Schaltungsstruktur mit vorgegebener Feedbackpartition verlangt wird, oder wenn es möglich ist, die Funktion λ_π sehr einfach zu gestalten. In den vorigen Abschnitten wurden für die Kodierungspartitionen τ_i der Schaltungsebenen die Toleranzen ermittelt, für die sich die Zahl der Stufen nicht vergrößert. Darüberhinaus wurde beschrieben, wie die τ_i in bezug auf die bereits fixierten Partitionen $\tau_1, \ldots, \tau_{i-1}$ der vorigen Stufen relativ-optimal zu bestimmen sind. Das Beispiel unter 3.3 zeigt, daß eine relativ-optimale Festlegung der τ_i und die

Beibehaltung der durch die Feedbackpartition bestimmten Stufenzahl nicht notwendig eine optimale Gesamtschaltung liefert, da die noch zu realisierenden Schaltungsebenen nicht berücksichtigt werden können. Eventuell ist es möglich, durch einen Vergleich der Partitionen $\tau_1^{ij}, \ldots, \tau_l^{ij}$ $\left(M(\tau_{ij}) \geqq \prod_{\lambda=1}^{l} \tau_\lambda^{ij}\right)$, von denen τ_{ij} abhängt, mit anderen möglichen, aber nicht mehr relativ-optimalen Partitionen $\tilde{\tau}_1^{ij}, \ldots, \tilde{\tau}_{l'}^{ij}$, eine neue Auswahl mit $l' < l$ zu ermitteln. Dabei darf der Zusatzaufwand zur Ansteuerung der neuen $\tilde{\tau}_{\lambda'}^{ij}$ nicht größer sein als die Aufwandsverminderung bei τ_{ij}.

Weiter ist es zweckmäßig, nicht einfach das gröbste τ_i mit $\tau_i \leqq \dfrac{N^{k-i}\big(m(A^k(\pi))\big)}{1 \cdot \tau_1 \cdots \tau_{i-1}}$ zu wählen, sondern in jeder Ebene *soviel wie möglich nicht-redundante* Information zu realisieren. Mirkin und Černyj [18] haben eine Abstandsdefinition für Partitionen angegeben, die hier Verwendung finden kann: π und τ haben einen um so größeren Abstand, je feiner ihr Produkt (i. b. a. π und τ) ist. Für die Auswahl der $\tau_{1_1}, \ldots, \tau_{i_{s_i}}$ bedeutet das, daß ihre Festlegung um so weniger ‚redundant' ist, je größer die Abstände untereinander sind. Wird das beachtet, so werden die $B^i(\pi)$ möglicherweise feiner, und die τ_l $(l > i)$ sind gröber wählbar. Darüber hinaus besteht eine größere Freiheit bei der Bestimmung der Mengen $\mathfrak{N}_l$ und $\mathfrak{N}_\pi$.

Für größere Automaten erscheint es vorteilhaft, das Verfahren mit anderen Kodierungsalgorithmen zu verbinden. Die Ermittlung der relativ-optimalen τ_{i_σ} beispielsweise kann durch Anwendung des in [8] vorgelegten Auswahlalgorithmus erfolgen, bei dem jeder *möglichen* Kodierungspartition ihr minimaler ‚Realisierungsaufwand' zugewiesen wird.

Nicht betrachtet wurde die auf ähnlicher Grundlage wie 3.2 mögliche Reduktion der Abhängigkeiten von den einzelnen Eingangsvariablen.

Noch ungelöst ist das Problem, „gute" Feedbackpartitionen zu ermitteln. In [13] ist ein Verfahren dafür angegeben, wie alle möglichen Feedbackpartitionen aufzufinden sind. Allerdings scheint es kein Kriterium für ihre Bewertung zu geben, ohne daß die entsprechende Realisierung berechnet wird.

Die vorliegende Untersuchung wurde durch die praktische Aufgabe angeregt, mit Hilfe automatentheoretischer Methoden ein digitales Entwurfsproblem zu lösen. Ich möchte den Mitarbeitern der Abteilung Technische Probleme im ORZ des Instituts für Nachrichtentechnik für das Interesse an dieser Problematik und die diesbezüglichen anregenden Diskussionen danken.

Literatur

[1] Hartmanis, J., Stearns, R. E., Algebraic Structure Theory of Sequential Machines. Englewood Cliffs, N.J., 1966.

[2] Starke, P. H., Abstrakte Automaten. Berlin 1969.

[3] Dolotta, T. A., McCluskey, E. J., The coding of internal states of sequential circuits. IEEE Trans. Electron. Comp. **EC-13** (1964), 549–563.

[4] Armstrong, D. B., A programmed algorithm for assigning internal codes to sequential machines. IRE Trans. Electron. Comp. **EC-11** (1962), 466–472.

[5] Armstrong, D. B., On the efficient assignment of internal codes to sequential machines. IRE Trans. Electron. Comp. **EC-11** (1962), 611–622.

[6] Reichel, L., Beitrag zur minimalen Darstellung von Automaten. XV. Internat. wiss. Kolloq. 28. Sept.–2. Okt. 1970, TH Ilmenau, Band C 2.

[7] Torng, H. C., An algorithm for finding secondary assignments of synchronous sequential circuits. IEEE Trans. Comp. **C-17** (1968), 461–469.

[8] STORY, J. R., HARRISON, H. J., REINHARD, E. A., Optimum State Assignment for Synchronous Sequential Circuits. IEEE Trans. Comp. **C-21** (1972) 12, 1365–1373.

[9] WEINER, P., SMITH, E. J., Optimization of reduced dependencies for synchronous sequential machines. IEEE Trans. Electron. Comp. **EC-16** (1967), 835–847.

[10] KARP, R. M., Some Techniques of state assignment for synchronous sequential machines. IEEE Trans. Electron. Comp. **EC-13** (1964), 507–518.

[11] ZAHLE, T. U., On Coding the States of Sequential Machines with the Use of Partition Pairs. IEEE Trans. Electron. Comp. **EC-15** (1966), 337–354.

[12] CURTIS, H. A., Multiple reduction of variable dependency of sequential machines. J. ACM **9** (1962), 324–344.

[13] DABADGHAO, S. V., A method for finding feedback-partitions for sequential machines. IEEE Trans. Comp., **C-18** (1969) 5, 465–467.

[14] LIU, C. L., Lattice Functions, Pair Algebras and Finite-State Machines. J. ACM **16** (1969) 3, 442–454.

[15] Мороз Д. З., Об одном алгоритме кодирования состояний автомата. Автом. и Вычисл. Техника, 4/1970, 21—24.

[16] Шишков Д. Б., Минимизация конечных автоматов. Kybernetika (Praha) **8** (1972) 4, 297—315.

[17] Карташева С. П., Обобщение канонические графы и рациональное кодирование автоматов. Автоматика и телемеханика 6/1969, 132—137.

[18] Миркин Б. Т., Черный Д. Б., Об измерении близости между различными разбиениями конечного множества объектов. Автоматика и телемеханика 5/1970, 120—127.

Kurzfassung

Mit Hilfe eines Quotientenbegriffes für Überdeckungen wird die in HARTMANIS/STEARNS [1] angegebene Feedback-Theorie für eine aufwandsgünstige Kodierung der inneren Zustände endlicher Automaten aufbereitet. Es werden Betrachtungen darüber angestellt, wie die ermittelten abgeschwächten Bedingungen für die Kodierungsüberdeckungen bei vorgegebener Feedbacküberdeckung (bzw. -partition) und unter Beibehaltung der Anzahl der Schaltungsstufen zu einer Vereinfachung der Ansteuerungskombinatorik der Speicher führen können.

Abstract

The algebraic feedback theory of HARTMANIS and STEARNS [1] is applicated to finding economical state assignments for finite automata using the introduced conception of set system division. It is discussed, how the found out weakened conditions for the assigment covers can lead to a reduction of the state logic for any given feedback cover (or partition, respectively) without enlarging the number of stages in the network scheme.

Резюме

На основе понятия деления разбиений конечных множеств применяется теория обратной связи Хартманиса и Стирнса [1] для экономичного кодирования внутренних состояний конечных автоматов. Для данного разбиения обратной связи рассматривается упрощение логических функций запоминающих элементов при помощи найденных ослабленных условий, причем число ступеней схемы остается первоначальным.

(Eingegangen am 22. 5. 1973)

Anschrift des Verfassers:
Dipl.-Math. K.-A. Zech
1058 Berlin
Oderberger Str. 42
DDR