HAMFU
HISTORY

KRIEG IM AETHER

Vorlesungen an der Eidgenössischen Technischen Hochschule in Zürich
im Wintersemester 1972/1973
Leitung: Abteilung für Übermittlungstruppen, Oberstdivisionär E. Honegger

Analyse von Chiffrierautomaten kontra Simulation

Referent: Dipl. Math. P. Nyffeler

Diese Vorlesung wurde durch die Stiftung HAMFU digitalisiert und als
PDF Dokument für www.hamfu.ch aufbereitet.

Analyse von Chiffrierautomaten kontra Simulation

Dipl. Math. P. Nyffeler

1. Einleitung

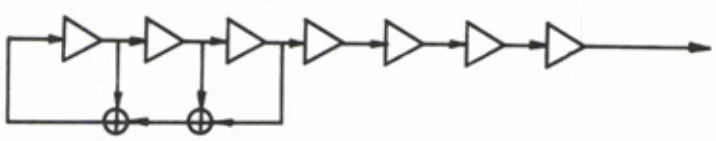
Zur Untersuchung von Automaten, d.h. einer Anordnung von Verzögerungsgliedern mit gegebener Uebergangs- und Ausgabefunktion, stehen im Wesentlichen zwei verschiedene Methoden zur Verfügung. Die eine besteht darin, den Automaten zu simulieren und durch statistische Methoden Aussagen über den Automaten zu erhalten. Bei der anderen Methode, der Analyse, wird irgendein Automat auf eine besonders einfache Klasse von Automaten, die linearen Automaten und insbesondere die linearen Schieberegister, zurückgeführt.

In diesem Vortrag soll nun, wenigstens andeutungsweise, ein computergerechter Algorithmus zur Analyse von Automaten beschrieben werden. Der Erläuterung des Ziels dient die folgende Figur:

GEGEBENER NICHT-LINEARER AUTOMAT:

$$
\begin{aligned}
s_1' &= s_2 \\
s_2' &= s_1 \\
s_3' &= s_4 \\
s_4' &= s_3 \\
s_5' &= s_3 (s_5 + s_6) + s_5 \\
s_6' &= s_3 (s_1 + s_5 + s_6 + s_1 s_5) + s_6 \\
\\
y &= s_5
\end{aligned}
$$

LINEARE REALISIERUNG:



Der durch Uebergangs- und Ausgabegleichungen gegebene nicht-lineare Automat mit sechs Verzögerungsgliedern lässt sich in einen linearen Automaten mit sieben Verzögerungsgliedern einbetten; die einfachere Struktur wird auf Kosten einer Dimensionserhöhung erreicht; insbesondere kommen beim neuen Automaten Zustände hinzu, welche keinem Zustand des alten Automaten entsprechen.

Zur Einbettung eines beliebigen Automaten in einen linearen Automaten wird im folgenden die Theorie der geordneten Partitionen benützt. Doch vorher sollen noch die einschränkenden Voraussetzungen, die für diesen Vortrag gelten, und die möglichen Verallgemeinerungen angegeben werden:

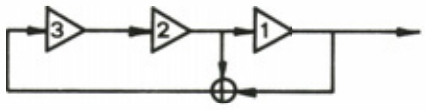
a) Es werden nur solche Automaten behandelt, deren Belegungen aus 0 oder 1 bestehen.
Ohne prinzipielle Schwierigkeiten könnte man 0,1,...,p-1 als Belegungsmöglichkeiten nehmen, wenn p eine Primzahl ist (statt mod 2 rechnet man mod p).

b) Es werden nur autonome Automaten, d.h. Automaten ohne Eingabe, behandelt.
Auch dies lässt sich verallgemeinern, allerdings erhält man dann nur noch eine Einbettung in teilweise lineare Automaten (vgl Bernd Reusch: Ueber vollständige und teilweise lineare Realisierung von Automaten, Gesellschaft für Mathematik und Datenverarbeitung, Bonn).

2. Lineare Realisierung von Automaten

2.1. Einführung der geordneten Partitionen

Die geordneten Partitionen sollen an einem linearen Schieberegister motiviert werden:



Dieses Schieberegister besteht aus den $2^3 = 8$ Zuständen 000 bis 111 (bzw 0 bis 7 in dualer Schreibweise).

Der Zustandskomponente i entspricht in natürlicher Weise diejenige geordnete Partition, welche die Menge aller Zustände aufteilt in solche mit i-ter Komponente 0 ($s_i=0$) und solche mit i-ter Komponente 1 ($s_i=1$); z.B:

$$s_1 \longleftrightarrow P_1 = (P_0^1, P_1^1) = (\{s/s_1=0\},\{s/s_1=1\}) = (0246,1357)$$

$$s_2 \longleftrightarrow P_2 = (P_0^2, P_1^2) = (\{s/s_2=0\},\{s/s_2=1\}) = (0145,2367)$$

In der Uebergangsgleichung für die 3. Komponente des obigen Schieberegisters tritt die mod 2-Summe s_1+s_2 auf; für diese Summe gilt, mit den Symbolen $\wedge$ bzw $\vee$ für die logischen Verknüpfungen "UND" bzw "ODER", offensichtlich:

$$s_1 + s_2 = 0 \longleftrightarrow (s_1=0 \wedge s_2=0) \vee (s_1=1 \wedge s_2=1)$$

$$s_1 + s_2 = 1 \longleftrightarrow (s_1=0 \wedge s_2=1) \vee (s_1=1 \wedge s_2=0)$$

In Analogie zu dieser Beziehung definiert man die Addition der zwei geordneten Partitionen P_1 und P_2:

$$P_1 \oplus P_2 = \{(P_0^1 \cap P_0^2) \cup (P_1^1 \cap P_1^2), (P_0^1 \cap P_1^2) \cup (P_1^1 \cap P_0^2)\}$$

$$P_1 \oplus P_2 = (0347, 1256)$$

0, 3, 4, 7 sind gemäss Konstruktion also gerade diejenigen Zustände, für welche die Summe der Komponenten s_1 und s_2 0 ergibt; 1, 2, 5, 6 diejenigen mit der Summe 1.

2.2. Theorie geordneter Partitionen

In diesem Abschnitt soll nun die Menge aller geordneter Partitionen auf einer gegebenen Menge (nach dem Vorbild des vorhergehenden Abschnitts) strukturiert und untersucht werden.

Definition 1: $A = (A_0, A_1)$ ist eine geordnete Partition auf der Menge M, wenn für das geordnete Mengenpaar (A_0, A_1) von Teilmengen aus M gilt:

a) $A_0 \cap A_1 = \emptyset$

b) $A_0 \cup A_1 = M$

Definition 2: P_M sei die Menge aller geordneter Partitionen auf M;

$$A=(A_0,A_1)\ ,\ B=(B_0,B_1)\in P_M$$

$$\boxed{A \bullet B=\{(A_0\cap B_0)\cup(A_1\cap B_1),(A_0\cap B_1)\cup(A_1\cap B_0)\}}$$

Aus diesen Definitionen ergeben sich die folgenden, leicht zu verifizierenden, Eigenschaften:

1) $(P_M, \bullet)$ ist eine ABELSCHE GRUPPE mit dem NEUTRALEN ELEMENT $M_0 = (M,\emptyset)$ und den INVERSEN

$\bullet A = A$

2) $(P_M, \bullet)$ ist ein VEKTORRAUM über GF(2), wobei $[0] \cdot A = M_0$, $[1] \cdot A = A$.

Damit können also Begriffe wie Basis, lineare Abhängigkeit bzw Unabhängigkeit usw im Raum der geordneten Partitionen über einer Menge sinnvoll verwendet werden.

3) Die Addition kann iteriert werden. Sei $A_i = (A_0^i, A_1^i)$, dann gilt:

$$\bigbullet_{i=1}^{n} A_i = \left(\sum_{k_1+\ldots+k_n=0} \bigcap_{i=1}^{n} A^i_{k_i}\ ,\ \sum_{k_1+\ldots+k_n=1} \bigcap_{i=1}^{n} A^i_{k_i} \right)$$

2.3. Lineare Realisierung von autonomen Automaten

Die im vorigen Abschnitt gewonnenen Aussagen werden nun auf die zwei Mengen S (Zustandsmenge) und Y (Ausgabemenge) sowie die zugeordneten Verktorräume P_S und P_Y angewendet. Vorerst wird in Analogie zur Uebergangs- bzw Ausgabefunktion der "Uebergangsoperator" Π_s bzw der "Ausgabeoperator" Π_y definiert:

Definition 3: a) $P = (P_0, P_1) \in P_S$

$$\Pi_s P = (\Pi_s P_0, \Pi_s P_1) = (\{s/s' \in P_0\}, \{s/s' \in P_1\})$$

b) $R = (R_0, R_1) \in P_y$

$$\Pi_y R = (\Pi_y R_0, \Pi_y R_1) = (\{s/y(s) \in R_0\}, \{s/y(s) \in R_1\})$$

Mit dieser zusätzlichen Definition ergeben sich folgende Eigenschaften:

1) Π_S bzw Π_Y ist mit $\bullet$ verträglich, d.h. Π_S bzw Π_Y ist ein

HOMOMORPHISMUS VON $(P_S, \bullet)$ IN $(P_S, \bullet)$ bzw VON $(P_Y, \bullet)$ IN $(P_Y, \bullet)$

2) $\{P^{(n)}, R^{(m)}\}$ sei eine Realisierung eines autonomen Automaten;

dann gilt:

a) $$s_i' = \sum_{j=1}^{r} s_{\lambda_j} \Longleftrightarrow \Pi_S P_i = \bigoplus_{j=1}^{r} P_{\lambda_j}$$

b) $$y_i = \sum_{j=1}^{t} s_{\mu_j} \Longleftrightarrow \Pi_Y R_i = \bigoplus_{j=1}^{t} P_{\mu_j}$$

wobei $\{\lambda_1, \ldots, \lambda_r\} \subseteq \{1, \ldots, n\}, \{\mu_1, \ldots, \mu_t\} \subseteq \{1, \ldots, n\}$

Diese Aussage ist sofort einleuchtend, wenn man an die folgenden konstruierten Zuordnungen denkt:

$$s_i \longleftrightarrow P_i$$
$$+ \longleftrightarrow \oplus$$
$$s_i' \longleftrightarrow \Pi_S P_i$$
$$y_i \longleftrightarrow \Pi_Y R_i$$

3) Als Korollar zu 2) erhält man:

$\{P^{(n)}, R^{(m)}\}$ sei eine lineare Realisierung eines autonomen Automaten; dann gilt:

a) Π_S ist ein HOMOMORPHISMUS VON $[P_1, \ldots, P_n]$ IN $[P_1, \ldots, P_n]$, wobei $[P_1, \ldots, P_n]$ der von den geordneten Partitionen $P_1, \ldots, P_n$ erzeugte Unterraum von P_S ist.

b) Π_Y ist ein HOMOMORPHISMUS VON $[R_1, \ldots, R_m]$ IN $[P_1, \ldots, P_n]$

Das in 3) beschriebene Korollar gestattet nun, bei vorgegebenen Ausgabepartitionen, die Zustandspartitionen des gesuchten linearen Automaten sukzessive zu berechnen gemäss folgendem Algorithmus:

$$\Pi_y R_1 := P_1 , \Pi_s P_1 := P_2 , \ldots\ldots , P_{k_1}$$

$$\Pi_y R_2 := P_{+1} , \Pi_s P_{+1} := P_{k_1+2} , \ldots\ldots , P_{k_2}$$

$$\Pi_y R_m := P_{k_{m-1}+1} , \ldots\ldots , P_{k_m}$$

In jeder Zeile dieses Algorithmus geht man nur soweit, bis eine lineare Abhängigkeit auftaucht,

z B $\Pi_s P_{k_1} = LK(P_1 , P_2 , \ldots , P_{k_1-1})$

Obiger Algorithmus wird nun im nächsten Abschnitt an drei ausgewählten Beispielen illustriert.

3. Beispiele

Beispiel 1 zeigt die lineare Realisierung eines durch sein Zustandsdiagramm gegebenen Automaten mit eindimensionalem Ausgaberaum. Als Basis zur Prüfung der linearen Abhängigkeit diente die "kanonische Basis" (0,...), (1,...),...,(15,...).

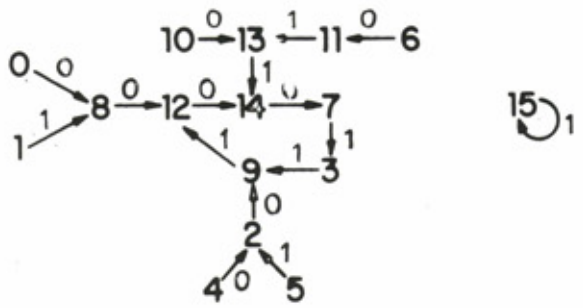
In Beispiel 2 wird ein "taktgesteuertes" Schieberegister linear realisiert. Das Hauptschieberegister macht nur dann einen Takt, wenn der Ausgang des Nebenschieberegisters 1 ist (s_5=1), sonst bleibt es stehen (s_5=0).

Um den Umweg über das Zustandsdiagramm nicht machen zu müssen, werden die in den geordneten Partitionen auftretenden Mengen durch Gleichungen statt durch Aufzählen der Elemente charakterisiert.

In Beispiel 3 schliesslich geht es darum, den Algorithmus bei zwei- bzw mehrdimensionalen Ausgaberäumen zu erläutern.

BEISPIEL 1

Zustandsdiagramm:



Algorithmus zur Linearen Realisierung:

$\pi_y R = (0\,2\,4\,6\,8\,10\,12\,14, 1\,3\,5\,7\,9\,11\,13\,15) := P_1 \Longrightarrow y = S_1$

$\pi_s P_1 = (0\,1\,4\,5\,8\,9\,12\,13, 2\,3\,6\,7\,10\,11\,14\,15) := P_2 \quad S_1' = S_2$

$\pi_s P_2 = (0\,1\,2\,3\,8\,9\,10\,11, 4\,5\,6\,7\,12\,13\,14\,15) := P_3 \quad S_2' = S_3$

$\pi_s P_3 = (0\,1\,2\,3\,4\,5\,6\,7, 8\,9\,10\,11\,12\,13\,14\,15) := P_4 \quad S_3' = S_4$

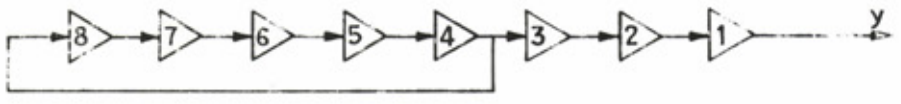
$\pi_s P_4 = (4\,5\,7\,14, 0\,1\,2\,3\,6\,8\,9\,10\,11\,12\,13\,15) := P_5 \quad S_4' = S_5$

$\pi_s P_5 = (12\,13\,14, 0\,1\,2\,3\,4\,5\,6\,7\,8\,9\,10\,11\,15) := P_6 \quad S_5' = S_6$

$\pi_s P_6 = (8\,9\,10\,11\,12\,13, 0\,1\,2\,3\,4\,5\,6\,7\,14\,15) := P_7 \quad S_6' = S_7$

$\pi_s P_7 = (0\,1\,2\,3\,6\,8\,9\,10\,11, 4\,5\,7\,12\,13\,14\,15) := P_8 \quad S_7' = S_8$

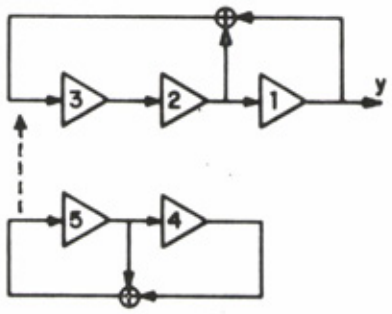
$\pi_s P_8 = (0\,1\,2\,3\,4\,5\,6\,7, 8\,9\,10\,11\,12\,13\,14\,15) = P_4 \quad S_8' = S_4$



1–7

BEISPIEL 2

Zustandsgleichungen:



$$S_1' = S_5 (S_1 + S_2) + S_1$$
$$S_2' = S_5 (S_2 + S_3) + S_2$$
$$S_3' = S_5 (S_1 + S_2 + S_3) + S_3$$
$$S_4' = S_5$$
$$S_5' = S_4 + S_5$$
$$y = S_1$$

Algorithmus zur Linearen Realisierung:

$\pi_y R = (S_1 = 0, \quad S_1 = 1) := P_1$

$\pi_s P_1 = (S_1 + S_1 S_5 + S_2 S_5 = 0, \ldots) := P_2$

$\pi_s P_2 = (S_1 + S_1 S_4 + S_1 S_5 + S_2 S_4 + S_3 S_5 + S_1 S_4 S_5 + S_3 S_4 S_5 = 0, \ldots) := P_3$

$\pi_s P_3 = (S_1 + S_1 S_4 + S_1 S_5 + S_3 S_4 + S_3 S_5 + S_1 S_4 S_5 + S_3 S_4 S_5 = 0, \ldots) := P_4$

$\pi_s P_4 = (S_1 + S_1 S_4 + S_2 S_5 + S_3 S_4 + S_1 S_4 S_5 + S_3 S_4 S_5 = 0, \ldots) := P_5$

$\pi_s P_5 = (S_1 + S_1 S_5 + S_2 S_4 + S_2 S_5 + S_3 S_5 + S_1 S_4 S_5 + S_2 S_4 S_5 + S_3 S_4 S_5 = 0, \ldots) := P_6$

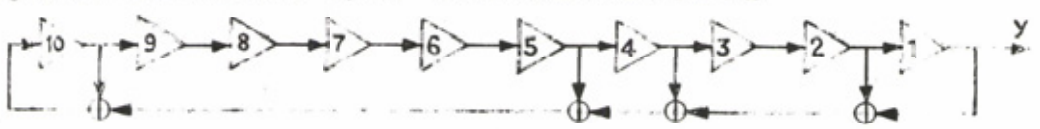
$\pi_s P_6 = (S_1 + S_1 S_4 + S_1 S_5 + S_2 S_4 + S_2 S_5 + S_3 S_4 + S_3 S_5 + S_1 S_4 S_5 + S_2 S_4 S_5 + S_3 S_4 S_5 = 0, \ldots) := P_7$

$\pi_s P_7 = (S_1 + S_1 S_4 + S_2 S_4 + S_2 S_5 + S_3 S_4 + S_3 S_5 + S_1 S_4 S_5 + S_2 S_4 S_5 + S_3 S_4 S_5 = 0, \ldots) := P_8$

$\pi_s P_8 = (S_1 + S_2 S_4 + S_3 S_4 + S_3 S_5 + S_3 S_4 S_5 = 0, \ldots) := P_9$

$\pi_s P_9 = (S_1 + S_3 S_4 + S_3 S_5 + S_3 S_4 S_5 = 0, \ldots) := P_{10}$

$\pi_s P_{10} = (S_1 + S_3 S_4 + S_3 S_4 S_5 = 0, \ldots) = P_1 \oplus P_2 \oplus P_4 \oplus P_5 \oplus P_{10}$

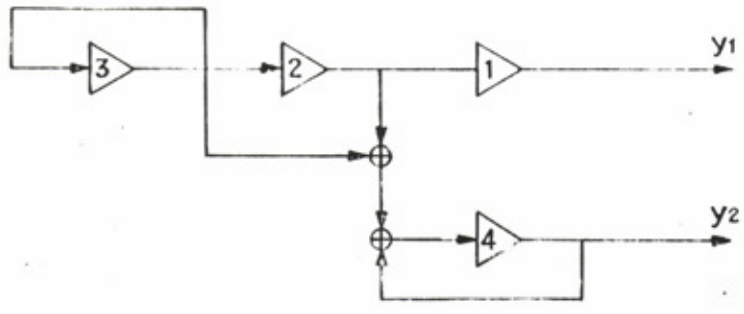


BEISPIEL 3

Zustandsgleichungen:

$$S_1' = S_1 + S_2 + S_3 \qquad y_1 = S_1 + S_2$$
$$S_2' = S_1\,S_2 \qquad y_2 = S_2 \cdot S_3$$
$$S_3' = S_2\,S_3$$

Algorithmus zur Linearen Realisierung:

$\pi_y R_1 = (S_1 + S_2 = 0, \ldots) := P_1$
$\pi_s P_1 = (S_1 + S_2 + S_3 + S_1\,S_2 = 0, \ldots) := P_2$
$\pi_s P_2 = (S_1 + S_2 + S_3 + S_1\,S_2 + S_2\,S_3 + S_1\,S_2 S_3 = 0, \ldots) := P_3$
$\pi_s P_3 = (S_1 + S_2 + S_3 + S_1\,S_2 + S_2\,S_3 + S_1\,S_2 S_3 = 0, \ldots) = P_3$

$\pi_y R_2 = (S_2 S_3 = 0, \ldots) := P_4$
$\pi_s P_4 = (S_1 S_2 S_3 = 0, \ldots) = P_2 + P_3 + P_4$