

5. Simulation graphischer Transformationen auf Digitalrechnern

Ein Nachteil der konventionellen Automatentafel ist ihre Unbrauchbarkeit zur Darstellung indeterministischer Strukturen, da infolge Matrixform der Tabelle nur jeweils ein δ -Wert je Index-Paar (z_v, x_i) angegeben werden kann. Es muß also zu einer Simulation des in Abschnitt 3 angegebenen graphischen Verfahrens bereits für den atomaren Fall eine andere Listen-Darstellung der δ -Funktion gefunden werden. Hierzu eignet sich die *Suchlisten-Darstellung* /59/, bei welcher im Gegensatz zur konventionellen Automatentafel die einzelnen Zeilen unterschiedliche Längen haben können. Die Länge der Zeile für einen Zustand entspricht dabei genau der Anzahl der vom entsprechenden Knoten abgehenden Zweige bei Darstellung durch einen Zustandsgraphen. Im Kapitel 5.1 wird die Suchlistendarstellung für echte Zustandsdiagramme beschrieben.

Bei deterministischen vollständig definierten Automaten ist die Zeilenlänge m in der Automatentafel gleich der Länge m des Eingabe-Alphabetes. Ist jedoch z.B. einer der Übergänge zweideutig, so wird die Zeilenlänge $m+1$ benötigt, was bei Suchlistendarstellung realisiert werden kann, bei Anwendung der konventionellen Darstellung jedoch nicht. Sind jedoch in einer Zeile beispielsweise 2 Übergänge nicht spezifiziert, so hat die Suchlistendarstellung die Zeilenlänge $m-2$, was dem Fortfall von Zweigen bei graphischer Darstellung entspricht. Diese Analogie gegenüber dem Zustandsgraphen gilt auch bei mehrdeutigen Übergängen, für die mehrere Zweige je Eingabesymbol benützt werden. Die Suchlistendarstellung hat bei Simulationsprogrammen gegenüber der konventionellen Automatentafel folgende Vorteile, wie in Kapitel 5.1 gezeigt wird.

1.) Vorteile der Suchlisten-Darstellung:

- a.) Meist erhebliche Speicherplatz-Ersparnis durch Weglassen der Rückweisungs-Spezifikation (gemäß Akzeptor-Definition)
- b.) Darstellbarkeit mehrdeutiger Übergänge

- c.) Sehr hohe Ersparnis an Testschritten bei globalen Prüfungen durch Analyse und Vergleich von Zuständen und ihrer Umgebung
- d.) Abkürzung der Einzelschritte zu c.
- e.) Sehr starke Rechenzeit-Ersparnis bei Reduktionsprozessen infolge c und d.
- f.) Flexibilität des Speicherplatz-Bedarf infolge gleitendem Listenformat.

Es ist für gerichtete Graphen charakteristisch, daß sowohl die Vorwärts- als auch die Rückwärts-Umgebung eines jeden Knoten durch visuelle Inspektion unmittelbar erfaßt werden kann. Eine solche unmittelbare Erfassung ist bei Automaten-tafeln nur für die Vorwärts-Umgebung von Zuständen möglich, während die Rückwärtsumgebung nur durch allumfassende Suchprozesse zugänglich gemacht werden kann. Dieser Nachteil der konventionellen Automaten-tafel gegenüber dem Graphen ist darauf zurückzuführen, daß im ersteren Fall nur die Übergangsfunktion δ , bei letzterem hingegen gleichzeitig mit δ auch die inverse Übergangsfunktion δ^{-1} explizit beschrieben ist (vgl. Def. 3.1.4 und Satz 3.1.3). Um die obigen Vorzüge des gerichteten Graphen auch für Computer-Anwendung nutzbar zu machen, was bei gewissen Prozessen Rechenzeiterparnisse um mehrere Größenordnungen ermöglichen kann, wurde die *doppelte Suchlistendarstellung* entwickelt, die gleichzeitig die Funktionen δ und δ^{-1} in Suchlistendarstellung erfaßt. Dieses Prinzip hat gegenüber der einfachen Suchliste folgende Vorteile bei Simulationsprogrammen, wie in Kapitel 5.2 gezeigt wird.

2.) Weitere Vorteile bei doppelter Suchlisten-Darstellung:

- a.) Starke Rechenzeiterparnis bei lokalen Operationen, wie beispielsweise Zustandsverschmelzung;
- b.) Gegenüber 1.e.) starke zusätzliche Ersparnis an Test- und Operations-Schritten bei Reduktionsprozessen infolge 1.c/d) und 2.a.) obiger Liste;

- c.) Reduktionsprozesse auch anwendbar bei Umwandlung indeterministischer Strukturen in deterministische Strukturen;

Mit Hilfe der beiden obigen Darstellungsformen für Automaten-tafeln können nur *softe Zustandsgraphen* per Liste erfaßt werden, da nur atomare Übergangszweige darstellbar sind. In Kapitel 5.3 wird ein Vorschlag zur rekursiven Erweiterung der Suchlistendarstellung eingeführt. Die so konzipierte *rekursive Suchliste* gestattet zusätzlich auch die Listen-Erfassung *uneigentlicher Zustandsgraphen*, da hierbei den dargestellten Zweigen nicht nur Symbole, sondern auch reguläre Ausdrücke zugeordnet sein dürfen. Auch die rekursive Suchliste kann in Duplexdarstellung angewandt werden. Gegenüber der nicht-rekursiven doppelten Suchliste ergeben sich bei Simulation folgende Vorteile, wie in Kapitel 5.3 gezeigt wird.

- 3.) Weitere Vorteile bei rekursiver Suchliste:
 - a.) Darstellbarkeit uneigentlicher Zustandsdiagramme
 - b.) Manipulierbarkeit regulärer Ausdrücke
 - c.) Anwendbarkeit zu Synthese- und Analyse-Algorithmen auf der Grundlage regulärer Ausdrücke bei Akzeptoren, Moore-Klassifikatoren und informationsreduzierenden Transduktoren

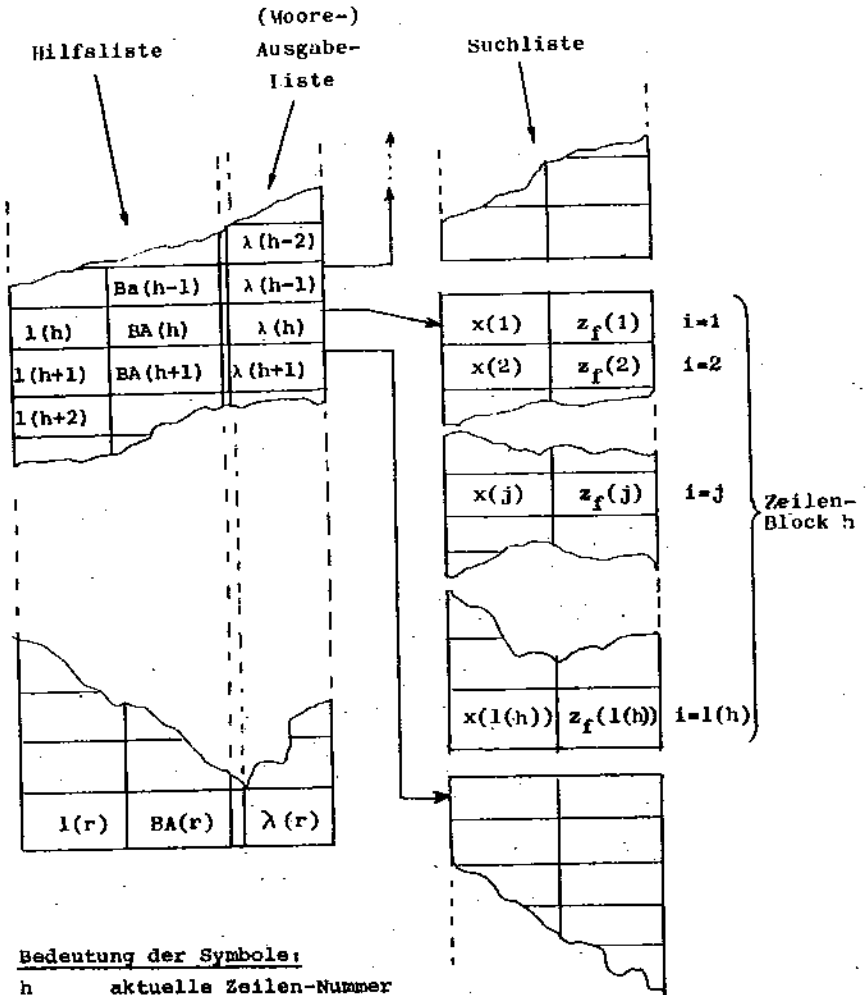
Schließlich wird in Kapitel 5.4 gezeigt werden, daß der in Abschnitt 3 beschriebene Satz graphischer Transformationsregeln auf einen einfachen Satz von elementaren Prüfungen und Elementar-Operationen zurückgeführt werden kann. Es wird eine computer-gerechte Strategie vorgeschlagen, die auf der Grundlage dieses Satzes von Prüfungen und Operationen eine vollautomatische Simulation der in Abschnitt 3 und 4 beschriebenen Synthese-Methode mit Hilfe der doppelten und rekursiven Suchlistendarstellung ermöglicht.

5.1 Die Suchlisten-Darstellung von Automatentafeln

In diesem Kapitel wird die *Suchlisten-Darstellung* für Automatentafeln vorgeschlagen, deren wichtigste Vorteile obenstehend aufgezählt sind. Die Verwandtschaft dieser Darstellung zu Graphen ist auf eine durch Hilfs Worte bedingte Listenstruktur nach Art eines Entscheidungsbaumes zurückzuführen. Zunächst seien einige Begriffe definiert, um eine Beschreibung der Suchlistenstruktur zu erleichtern.

Bei konventionellen Automatentafeln in Matrixform ist jeder definierte Übergang durch Angabe des Folgezustandes z_f als nicht-negative ganze Zahl angegeben, wobei jeder nicht explizit definierte Übergang durch ein besonderes Leerzeichen in der Tabelle gekennzeichnet ist. Bei der Suchlisten-darstellung wird nur jeder explizit definierte Übergang durch ein geordnetes Paar in der Form (x_i, z_f) spezifiziert, das *Übergangsregel* genannt sei. Die Gesamtheit aller Übergangsregeln des gleichen Vorzustandes z_h wird als *Zeilenblock* Nr. h bezeichnet. Jedem Zeilenblock bei Suchlisten-darstellung entspricht eine Zeile bei Vorliegen einer konventionellen Automatentafel. Ein Zeilenblock h einer Automata-tafel in Suchlistendarstellung ist die listenmäßige Beschreibung des abgehenden Gesamtbündels von Knoten h eines entsprechenden Zustandsgraphen.

Eine Automatentafel in Suchlistendarstellung besteht aus drei verschiedenen Listen, die *Suchliste*, *Hilfsliste* und *Ausgabefliste* genannt werden. Die Gesamtheit aller Zeilenblöcke bildet die *Suchliste*. Wenn ein Vorzustand z_h gegeben ist und somit der diesem zugeordnete Zeilenblock bekannt ist, wird der Folgezustand auf ein gegebenes Eingabesymbol x_i nicht durch Indizierung, sondern durch einen Suchprozeß ermittelt. Hierbei wird der aktuelle Zeilenblock abgesucht nach einer Übergangsregel, die das Eingabesymbol x_i enthält (Bild 3.2). Wird eine solche Übergangsregel



Bedeutung der Symbole:

- h aktuelle Zeilen-Nummer
- r Anzahl der Zustände des Automaten
- $l(h)$ Länge des Zeilenblockes der Zeile h
- $BA(h)$ Basis-Adresse des Zeilenblockes der Zeile h
- $x(i)$ i -tes Eingabesymbol im Zeilenblock h
- $z_f(i)$ Folgezustand bei Eingabesymbol $x(i)$

Bild 5.1.1 Einfache Suchlistendarstellung eines Moore-Automaten

(x_i, z_f) gefunden, so ist damit der Folgezustand z_f ermittelt. Der beschriebene Suchprozeß ist der Grund dafür, weshalb die Bezeichnung *Suchlistendarstellung* gewählt wurde.

Die *Hilfsliste* hat die Aufgabe, zu jedem Vorzustand z_h den zugehörigen Zeilenblock Nr. h zugänglich zu machen. Dies erfolgt dadurch, daß jedem Zeilenblock ein Wertepaar $(l(h), BA(h))$ der Hilfsliste zugeordnet ist, wobei die *Basis-Adresse* $BA(h)$ angibt, wo Zeilenblock Nr. h beginnt, und die *Zeilenlänge* $l(h)$ angibt, aus wievielen Übergangsregeln der Zeilenblock h besteht. Eine beliebige Position h der Hilfsliste kann unmittelbar durch Indizierung aufgefunden werden. Bei Simulation des Erkennungsbetriebes eines Akzeptors erfolgt ein Zustandsübergang gemäß dem Strukturdiagramm in Bild 5.1.2.

Bei der Suchlistendarstellung von Zustandsgraphen entspricht jedem Knoten ein Doppelwort der Hilfsliste, und jedem Zweig eine Übergangsregel, sowie jedem Zeilenblock ein Gesamtbündel (Def. 3.1.8). Eine Automatentafel in Suchlistendarstellung ist somit die formale Beschreibung eines entsprechenden Zustandsgraphen.

Die Ausgabefunktion wird bei der Suchlistendarstellung in der sogenannten *Ausgabeliste* spezifiziert. Für die Darstellung solcher Akzeptoren oder Transduktoren, die sich nach Art eines Moore-Automaten beschreiben lassen (vgl. Abschnitt 4), wird die Hilfsliste um eine Spalte (Ausgabespalte) erweitert, so daß jedem Zustand z_v ein Tripel $(l(h), BA(h), \lambda(z_h))$ zugeordnet ist. Mit einer solchen Listen-Anordnung können folgende Automatentypen spezifiziert werden:

- 1.) Moore-Akzeptor mit $\lambda(z_h) = -1$ für Anfangszustände, $\lambda(z_h) = +1$ für Endzustände, und $\lambda(z_h) = 0$ für gewöhnliche Zustände,

2. Moore-Klassifikator mit $\lambda(z_h)$ wie unter 1., jedoch mit $+1 \leq \lambda(z_h) \leq k$ für die Klassenzuordnung der Endzustände,
3. Moore-Transduktor mit Anfangszustand mit $\lambda(z_h)$ wie unter 2.,
4. Allgemeiner Moore-Transduktor mit $\lambda(z_h)$ wie unter 2.

In entsprechender Weise kann die Suchlistendarstellung auch nach dem Mealy-Typ gestaltet werden, indem nicht die Hilfsliste, sondern die Suchliste um eine Spalte (Ausgabespalte) erweitert wird. Auf diese Weise erscheint jede Übergangsregel nicht als Paar, sondern als Tripel $(x_1, \delta(z_h, x_1), \lambda(z_h, x_1))$. Hierbei werden durch $\lambda = 0$ oder $\lambda > 0$ zwei Typen von Übergangsregeln unterschieden, wobei $\lambda = 0$ für δ -Übergänge und $\lambda > 0$ für δ' -Übergänge steht. Durch diese Listenanordnung können mit Hilfe der Suchlisten-Darstellung Mealy-Akzeptoren und Mealy-Transduktoren mit Anfangszustand und darüber hinaus allgemeine Mealy-Transduktoren spezifiziert werden.

Wenn eine Menge von Eingabefolgen nebst Klassenzuordnungen als Klassifikationsproblem vorgegeben ist, reicht die beschriebene *einfache Suchlistendarstellung* mit Hilfe der in Tafel 5.1.1 angeführten Manipulationsprozeduren aus zur automatisierten Synthese und Reduktion von Moore-Klassifikatoren (vgl. Kapitel 4.1), sowie zu deren simuliertem Klassifikationsbetrieb. Darüber hinaus können durch Einzelanwendung dieser Prozeduren eine Reihe weiterer Manipulationen durchgeführt werden, wie beispielsweise das Löschen von Übergängen und Zuständen, gezieltes Einsetzen von Übergängen, und gezieltes Verschmelzen von Zuständen. Diese

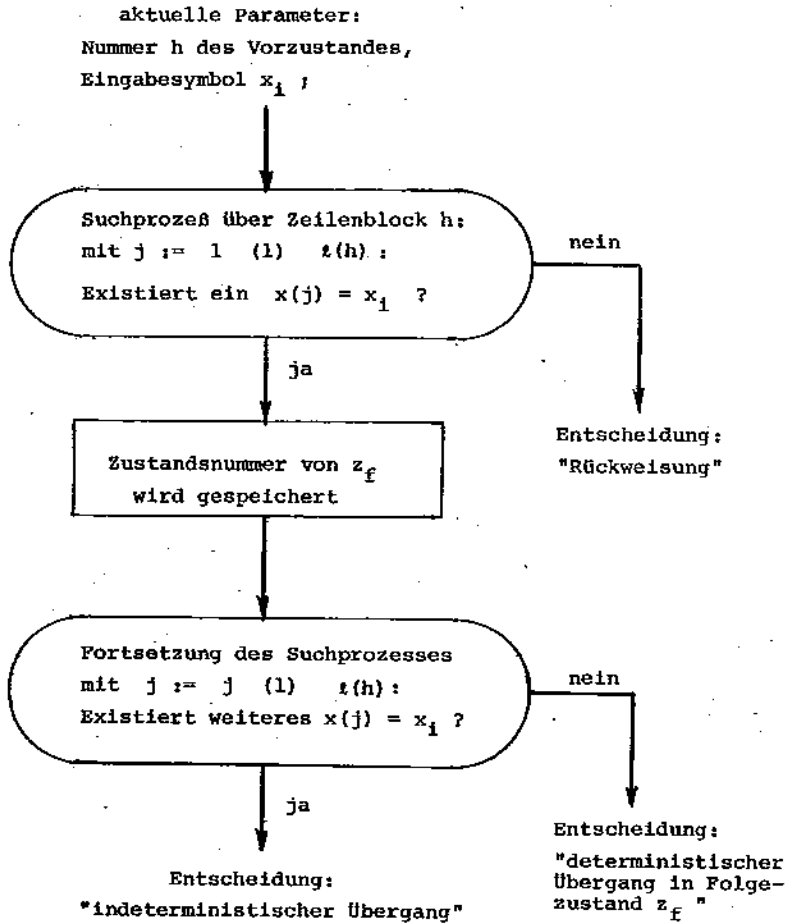


Bild 5.2 Simulation des Klassifikationsbetrieb eines Automaten in einfacher Suchlistendarstellung: Beispiel für Ermittlung eines Überganges.

Technik wurde in Form von Maschinenprogrammen auf einer Rechenanlage vom Typ CD 3300 zur Synthese von Moore-Klassifikatoren mit mehr als 8000 Zuständen verifiziert /47/. Bei Manipulationen (wie beispielsweise bei Zustandsverschmelzung) freiwerdende Zeilen sind für spätere Schritte wieder verwendbar, was in der Hilfsliste durch einen "Frei"-Merker angezeigt wird. Freiwerdende Übergangsregeln innerhalb nicht-gelöschter Zeilen gestatteten über eine Listenkompressions-Prozedur die Verkleinerung der Liste zwecks Erzielung von Speichereffektivität durch *gleitendes Listenformat*. Hierbei wird die bei modernen Rechnern meist vorauszusetzende Existenz delegierbarer schneller Block-Transfer- und Suchbefehle ausgenutzt.

Die Suchlistendarstellung von Akzeptoren und Moore-Klassifikatoren führt bei der Programmierung von Reduktions-Algorithmen nach der Methode des sukzessiven Verschmelzens (GILL: /26/) zu erheblichen Einsparungen an Rechenzeit. Dies ist darauf zurückzuführen, daß die Menge der Zustände aufgrund der unterschiedlichen Zeilenlängen in Klassen unterteilt ist. Durch die Erfassung der Zeilenlängen in der Hilfsliste liegt also diese Klassenunterteilung der Knoten des simulierten Graphen aufgrund der Größe ihrer abgehenden Bündel unmittelbar ablesbar vor. Da nur Knoten mit gleichgroßen abgehenden Bündeln nicht-unterscheidbar sein können, kann der paarweise Vergleich der Zeilen entsprechend der aktuellen Zeilenlänge auf die jeweilige Untermenge von Zuständen beschränkt werden. Zur Veranschaulichung diene folgendes Beispiel eines Rechenzeitvergleiches. Ein Akzeptor habe r Zustände und ein Eingabe-Alphabet der Länge m , wobei Gleichverteilung der Zeilenlängen vorausgesetzt sei. Die Kombinatorik lehrt, daß bei paarweisem Vergleich von r Elementen $r(r-1)/2$ einzelne Vergleichsschritte notwendig sind. Für große r ist die Anzahl von Schritten näherungsweise $r^2/2$. Bei nach Längenklassen getrenntem Vergleich ist die Anzahl

Parameter	Operation
Zustand z_v , Eingabesymbol x_i	Definition eines neuen Zustandes z_f , Definition eines Überganges $z_f = \delta(z_v, x_i)$
Zustand z_v , Zustand z_f , Eingabesymbol x_i	Definition eines Überganges $z_f = \delta(z_v, x_i)$, ggf. Widerspruch-Meldung
zwei Zustände z_i und z_j	Verschmelzung von z_i und z_j
Zustand z_v , Eingabesymbol x_i	Ermittlung des Folgezustandes $z_f = \delta(z_v, x_i)$, ggf. Rückweisungs-Meldung
Zustand z_i , Ausgabesymbol y	Definiere Ausgabe $\lambda(z_i) = y$
Ein Eingabesymbol x_i , zwei Zustände z_v , $z_f =$ $z_f = \delta(z_v, x_i)$	Löschen der Übergangs- Definition $\delta(z_v, x_i)$
Zustand z_i	Löschen des Zustandes z_i
	Ordnen der Zeilenblöcke nach steigenden (x_i, z_f)
	Minimisierung durch sukzessi- ves Verschmelzen einfach- äquivalenter Zustände
	Kompression der Liste durch Entfernen frei gewordener Listenplätze

Tafel 5.1.1: Vorhandene Unterprogramme zur Manipulation von Moore-Klassifikatoren in einfacher Suchlisten-darstellung.

der Schritte $r \cdot r/m$ ($r/m - 1$)/2 oder für große r näherungsweise $r^2/2m$. Dies bedeutet eine Einsparung an Vergleichsschritten um den Faktor m , wobei noch nicht berücksichtigt wurde, daß der einzelne Vergleichsschritt dadurch eine zusätzliche Einsparung bringt, daß die durchschnittliche Länge der zu vergleichenden Zeilen beispielsweise bei Gleichverteilung nur die Hälfte beträgt. (Durch Einhaltung eines Ordnungsschemas für Zeilenblöcke ist ein Zeilenvergleich ohne Permutation der Übergangsregeln möglich.) Zur vollständigen Reduktion von Moore-Klassifikatoren mit $r \approx 8000$ und $m = 9$ wurden auf einem Digitalrechner CD 3300 Rechenzeiten zwischen 5 und 10 Minuten benötigt /59/.

5.2 Die doppelte Suchliste als Abbild des gerichteten Graphen

Die in Kapitel 5.1 beschriebene einfache Suchlisten-Darstellung einer Automatentafel ist lediglich eine Darstellung der δ -Funktion. Eine echte Simulation von Graphen wird erst durch die *doppelte Suchlistendarstellung* möglich, die sowohl die δ -Funktion als auch die hierzu inverse Funktion δ^{-1} darstellt, wobei für diese beiden Funktionen getrennte, aufeinander abgestimmte Listensysteme mit eigenen Hilfslisten vorhanden sind. Durch eine doppelte konventionelle Automatentafel in konventioneller Form wäre selbst im deterministischen Falle die Spezifikation von Automaten unmöglich, da die Inversion δ^{-1} einer eindeutigen δ -Funktion nicht notwendigerweise eindeutig ist.

Ein Vorteil der Doppeldarstellung besteht darin, daß zur Feststellung indeterministischer Strukturen erster Art erforderliche paarweise Vergleiche von Knoten bezüglich ihrer Rückwärts-Umgebung (vgl. Bild 3.2 und 3.2.4 b) mit dem gleichen Unterprogramm durchgeführt werden können, welches durch Vergleiche der Vorwärtsumgebung von Knoten einfach-äquivalente Zustände ermittelt (Bild 3.2.4 a). Desgleichen ist eine Vorwärts-Minimisierung und eine Rückwärts-Minimisierung mit Hilfe des gleichen Unterprogrammes möglich. Ein weiterer Vorteil der Doppeldarstellung, der sich bei einer computergerechten Simulationsstrategie für graphische Verfahren auswirkt, wird in Kapitel 5.4 beschrieben.

darstellung ergibt sich hierbei ein besonders einfacher Satz elementarer Operationen dadurch, daß ein Teil dieser Operationen unverändert sowohl bezüglich δ als auch bezüglich δ^{-1} angewandt werden kann.

Ein weiterer Vorteil der doppelten Suchliste ist eine erhebliche Rechenzeit-Ersparnis bei Verschmelzungsprozessen. Bei einer Einfach-Darstellung von δ werden zwei Zustände z_i und z_j zu z_i verschmolzen, indem die Zeile j für z_j eliminiert wird, und unter Absuchen der gesamten δ -Liste alle Folgezustände z_j durch z_i ersetzt werden. Bei doppelter Darstellung kann auf diesen Suchprozeß verzichtet werden, da Vorzustände von z_j , deren Zeilen zu ändern sind, durch die inverse Funktion δ^{-1} direkt definiert sind. Zur Veranschaulichung sei ein Rechenzeitvergleich zwischen einfacher konventioneller und doppelter Suchlistendarstellung für diesen Suchprozeß angestellt. Bei einer Alphabettlänge m und r Zuständen erfordert die einfache konventionelle Automaten-tafel $r \cdot m$ Suchschritte. Bei einer mittleren Zeilenlänge von $m/2$ für beide Suchlisten erfordert die doppelte Suchlistendarstellung $m/2 \cdot m/2 = m^2/4$ Suchschritte ($m/2$ Vorzustände mit $m/2$ Zeilenlänge als Durchschnittswerte). Für die Zahlenwerte $r = 1000$ und $m = 10$ ergibt sich beispielsweise eine Schritt-Anzahl von 10.000 gegenüber 25.

5.3 Die Manipulation regulärer Ausdrücke mit Hilfe der Suchlistendarstellung

Mit den in Kapitel 5.1 und 5.2 bisher beschriebenen Methoden der Suchlistendarstellung können nur echte Zustandsgraphen tabellarisch erfaßt werden. Für die Manipulation uneigentlicher Zustandsgraphen muß eine zusätzliche Möglichkeit zur Darstellung von regulären Ausdrücken einbezogen werden. Zu diesem Zweck wird das erste Wort der betreffenden Übergangsregel modifiziert in der Weise, daß es statt eines Symbolen $x_i \in X$ eine Adresse AD und einen *Indirekt-Merker* enthält.

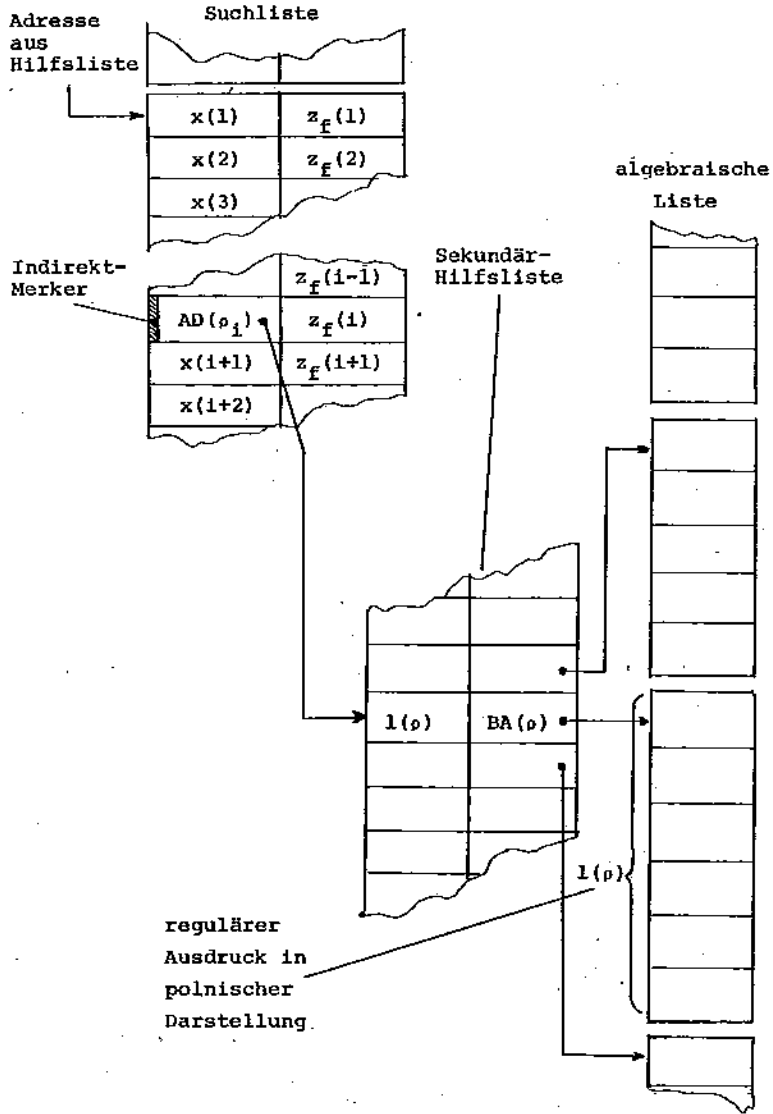


Bild 5.3.1 Darstellung nicht-atomarer Zweige durch rekursive Suchlisten-Darstellung

Der "Indirekt"-Merker soll Verwechslungen zwischen Symbolen und Adressen verhindern. Die Adresse AD ist ein Index, der das Auffinden des regulären Ausdruckes in einer *algebraischen Liste* ermöglicht, wie in Bild 5.3.1 gezeigt wird. Die algebraische Liste hat die gleiche Struktur, die auch dem Aufbau der Suchliste zugrundeliegt. Ein Doppelwort einer *Sekundär-Hilfsliste* gibt Basisadresse und Zeilenlänge eines Blockes der algebraischen Liste an. Die Zeilenlänge ist durch die Länge des in diesem Block gespeicherten regulären Ausdruckes bestimmt. Für die Struktur eines solchen Blockes wird eine *polnische Darstellung* des regulären Ausdruckes vorgeschlagen. Das Prinzip der *polnischen Darstellung* /57/ algebraischer Ausdrücke ist die Wiedergabe ohne Verwendung von Klammern. Die Priorität der einzelnen Operatoren innerhalb des Ausdruckes ergibt sich dabei eindeutig aus der Folge. Das bekannte Schema nach LUKASIEWICZ (beschrieben beispielsweise in /57/) muß für die KLEENE'sche Algebra zwecks Einbeziehung der einstelligen Iterations-Operation modifiziert werden. Folgendes Beispiel zeigt die polnische Darstellung eines regulären Ausdruckes.

Beispiel: Gegeben ist $\rho = \{a \cup b(cd)^*\}^*$.

Die polnische Darstellung des regulären Ausdruckes ρ lautet:

$* \cup a \cdot b * \cdot c d.$

Die Reihenfolge der Operator-Symbole in einem regulären Ausdruck in polnischer Darstellung ist ein geeigneter Ansatz zur Formulierung eines Dekompositions-Algorithmus für uneigentliche Zustandsgraphen nach folgendem Satz.

Satz 5.3.1: Die Reihenfolge der Operator-Symbole in der polnischen Darstellung eines regulären Ausdruckes entspricht der Folge der einzelnen Dekompositions-Schritte des Synthese-Algorithmus nach OTT/FEINSTEIN.

Auf die Wiedergabe des Beweises wird verzichtet. Zur Veranschaulichung diene Bild 2.3.9, welches die Dekompositions-

Schritte am Beispiel des obigen regulären Ausdruckes graphisch zeigt. Ein weiteres einfaches Beispiel in Bild 5.3.2 veranschaulicht die Anwendung der polnischen Darstellung in Suchlistenform zur Dekomposition eines regulären Ausdruckes.

Bild 5.3.2 a zeigt als Beispiel einer Dekompositions-Aufgabe einen Ausschnitt aus einem uneigentlichen Zustandsdiagramm in Suchlistendarstellung. Der zu verarbeitende reguläre Ausdruck in der Form $\rho = 1 \cdot 2 \vee 3$ hat die polnische Darstellung $\cup 3 \cdot 2 1$, welche in inverser Folge ihrer Symbole in der algebraischen Liste abgespeichert ist. Die den vorliegenden Übergang $\delta(z_7, \rho) = z_8$ beschreibende Übergangsregel in der Suchliste enthält anstelle eines Eingabesymbols (wie im atomaren Fall) eine Adresse, was durch einen Indirekt-Merkter angezeigt ist. Die Simulation der Dekomposition von ρ durch Transformation der in Teil a) des Bildes gezeigten Suchlistendarstellung erfolgt in den folgenden Schritten:

1. Schritt: Transformation von Darstellung a) in Darstellung b. Das letzte Symbol im vorliegenden Block der algebraischen Liste bestimmt, welche Transformation zuerst vorgenommen wird: aus \vee folgt die disjunktive Dekomposition. Der Block der algebraischen Liste wird gespalten, indem ein neues Wertepaar in die Sekundär-Hilfsliste eingeführt, und das alte Wertepaar in seiner Längenangabe $l(\rho)$ korrigiert wird von 5 auf 3 (Bild 5.3.2 b). Das Symbol für den Disjunktions-Operator wird gelöscht, was im Bild durch Schraffur angedeutet ist. Für den durch Abspaltung eines Unterausdruckes gebildeten zusätzlichen Block der algebraischen Liste wird eine neue Übergangsregel definiert durch Verlängerung des Zeilenblockes in der Suchliste (Bild b). Der Graph hat nunmehr zwei Pfade von Knoten 7 nach Knoten 8.

2. Schritt: Transformation von Darstellung b) in Darstellung c. Der durch Blockspaltung in der algebraischen Liste durch Schritt 1 entstandene neue Block hat die Länge 1, was durch das zugeordnete neue Wertepaar der Sekundär-Hilfsliste

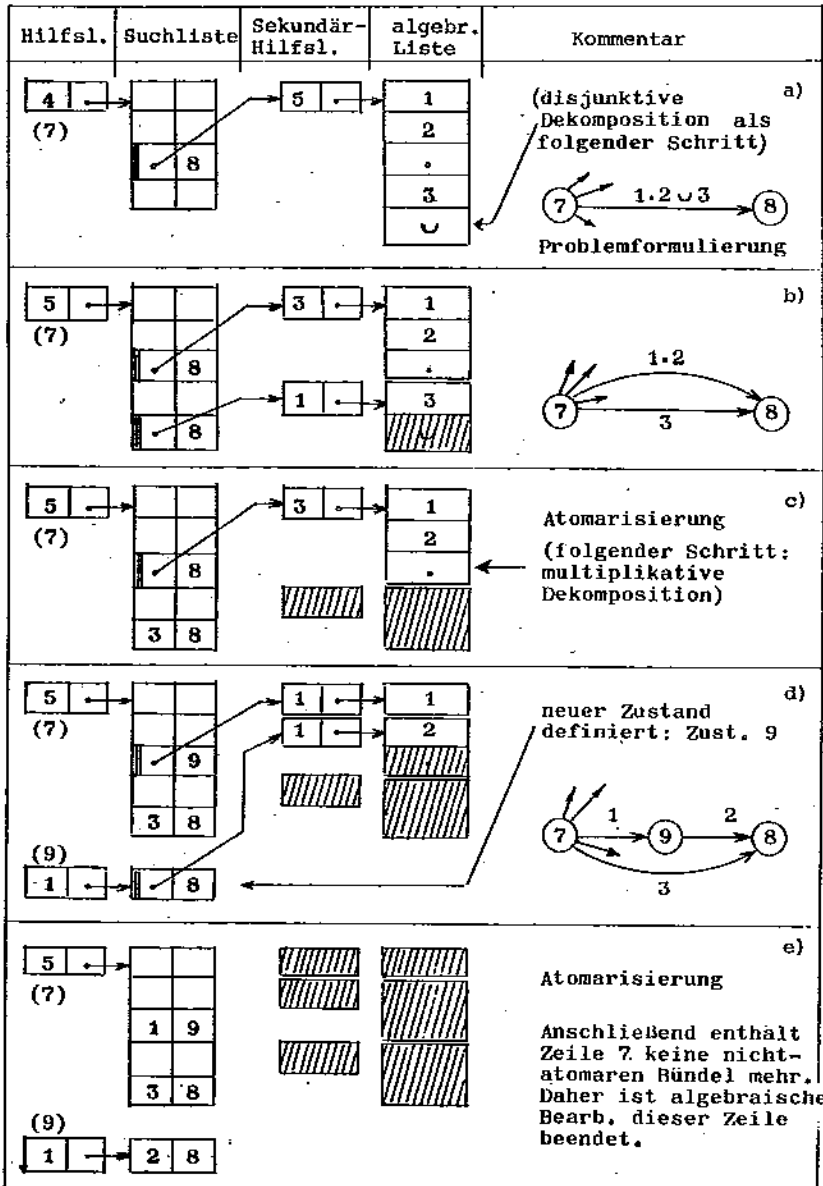


Bild 5.3.2 Beispiel für die Simulation graphischer Transformationen an uneigentlichen Zustandsgraphen

angezeigt wird. Dies bedeutet, daß der reguläre Ausdruck atomar ist, also ein Einzelsymbol des Eingabealphabetes ist (Symbol: 3). Der als *Atomarisierung* bezeichnete Schritt verlagert den Ort der Darstellung dieses Symbols von der algebraischen Liste in die Suchliste, indem in der entsprechenden (5.) Position des zugeordneten Zeilenblockes Adresse und Indirekt-Merker gemeinsam durch das Eingabesymbol ersetzt werden. Der dadurch überflüssig gewordene Block in der algebraischen Liste und das diesem zugeordnete Parameter-Paar in der Sekundär-Hilfsliste werden gelöscht, was durch Schraffur angedeutet wurde (Bild 3.3.2 c).

3. Schritt: Transformation von Darstellung c) in Darstellung d). Die Operation in diesem Schritt wird durch das letzte Symbol im vorliegenden Block der algebraischen Liste bestimmt. Aus diesem Symbol "." folgt die Multiplikative Dekomposition. Hierzu muß ein neuer Knoten eingeführt werden (Knoten Nr. 9 nach Bild 5.3.2 d). Dies geschieht durch Anlegen einer neuen Zeile über die Einführung eines neuen Wertepaares in eine freie Position der Hilfsliste (Pos. 9 in diesem Beispiel). Der reguläre Ausdruck in der algebraischen Liste wird gespalten, wie in Schritt 1, jedoch mit folgenden Unterschieden. 1) Die neu zu definierende Übergangsregel für den zusätzlichen Pfad im Graphen wird in den Zeilenblock des neu definierten Zustandes z_9 eingeführt. 2) Die Folgezustands-Eintragung (Pos. 3 in Zeilenblock von Zustand 7) für den ersten vorliegenden Block der algebraischen Liste muß von 8 auf 9 geändert werden. Bild 5.3.2 d zeigt den Graphen der durch Schritt 3 transformierten Listenstruktur.

4. Schritt: Transformation von Darstellung d) in Darstellung e). Durch den 3. Schritt sind in der algebraischen Liste 2 atomare Ausdrücke entstanden. Diese werden durch 2 Atomarisierungs-Schritte wie Schritt 2 in die Suchliste verlagert. Bild 5.3.2 e zeigt die als Endergebnis auftretende Listenstruktur für den Graphen nach Bild 5.3.2 d.

Die oben eingeführte *einfach-rekursive Suchlisten-Darstellung* eines uneigentlichen Zustandsgraphen beruht auf zwei Ebenen der Darstellung. In der ersten Ebene, die durch direkte Adressierung in der Suchliste zugänglich ist, werden atomare Zweige des Graphen erfaßt. In der zweiten Ebene, die durch indirekte Adressierung (algebraische Liste) zugänglich ist, werden solche Zweige des Graphen wiedergegeben, denen Ausdrücke mit atomaren Komponenten zugeordnet sind. Dies bedeutet, daß in der zweiten Ebene nur Symbole des Eingabe-Alphabetes oder Konnektive gespeichert werden können. Durch Zulassen des aus der ersten Ebene (Suchliste) bekannten "Indirekt"-Merkers in der algebraischen Liste kann die Listenhierarchie rekursiv um beliebig viele Ebenen erweitert werden. Hierdurch können auch solche reguläre Ausdrücke erfaßt werden, die nicht-atomare Komponenten enthalten, wie z.B. Deskriptoren von Residuen-Graphen (Kap. 4.2).

5.4 Strategien zur Simulation graphischer Synthese-Verfahren mittels Suchlistendarstellungen

Das in den Abschnitten 3 und 4 eingeführte graphische Syntheseverfahren auf der Grundlage des abstrakten Akzeptor-Modells geht hinsichtlich des gewählten Instrumentariums an Transformationsregeln von der Fähigkeit des Menschen zur visuellen Erkennung relativ komplexer graphischer Strukturen aus. Gegenüber diesem auf der Technik der *Inspektion* basierenden Instrumentarium wird für den Einsatz von Rechenanlagen zu Simulationszwecken ein mehr computer-gerechtes Instrumentarium benötigt. Auf der einen Seite ist das Programmieren von Erkennungsalgorithmen, wie etwa zur Struktur-Erkennung relativ schwierig und aufwendig. Gleichzeitig liegt aber

die Stärke des Computers in der Anwendung zyklischer Unterprogramme. Aus diesem Grunde sind bei Simulation systematische Techniken strukturellen Methoden vorzuziehen. Aus Gründen der Effizienz hinsichtlich Programmierung und Speicherbedarf ist es wünschenswert, daß die Unterprogramme möglichst elementar operieren und deren Anzahl möglichst gering ist. Im Folgenden werden die in Abschnitt 3 eingeführten graphischen Transformationen auf wenige elementare Operationen zurückgeführt.

Die Technik der *Inspektion* bei manuell ausgeführten graphischen Verfahren beruht auf elementaren Prüfungen von Knoten und deren Umgebung (ankommende und abgehende Bündel). Dabei können einstellige und zweistellige Prüfungen unterschieden werden nach folgender Liste. Bei Zugrundelegung der dualen Darstellung der δ -Funktion (doppelte Suchliste) ergibt sich für einen Teil dieser Prüfungen eine Version a und eine Version b, wobei der einzige Unterschied durch die Anwendung auf die Funktion δ oder δ^{-1} bedingt ist.

A. Einstellige Prüfungen (das Bündel oder die Eigenschaft eines einzigen Knoten z_v wird geprüft).

- Test 1 Alle abgehenden Zweige werden einzeln auf Atomarität geprüft;
- Test 2a/(b) Alle abgehenden (ankommenden) Zweige werden einzeln auf das Leersymbol Λ geprüft;
- Test 3 Alle abgehenden Zweige werden paarweise auf Gleichheit geprüft (Zweideutigkeit);
- Test 4a/(b) Knoten wird auf End-(Anfangs-)knoten-Eigenschaft mit $\lambda > 0$ ($\lambda < 0$) geprüft;
- Test 5a/(b) Knoten wird auf Unzugänglichkeit (Isoliertheit) geprüft, d.h. auf Existenz eines ankommenden (abgehenden) Zweiges; (vgl. Def. 5.4.1)

B. Zweistellige Prüfungen (die Bündel zweier Knoten z_i und z_j werden verglichen).

- Test 6a/(b) Prüfe Knotenpaar auf Vorwärts-(Rückwärts-)Äquivalenz;

Aufgrund der Entscheidungen solcher Tests zur Bündel-Untersuchung kann die Anwendung von Transformationen bestimmt werden. Sämtliche in Abschnitt 3 beschriebenen Transformationen lassen sich auf die folgenden elementaren Operationen zurückführen, die wiederum in einstellige und zweistellige Operationen gegliedert sind nach der Anzahl der Bezugs-Operanden.

C. Einstellige Operationen (primär ist ein Knoten z_v bzw. z_f betroffen)

- Operation 1 Definition eines neuen Knotens (in freige-wordener Zeile oder - falls erforderlich - am Listen-Ende);
- Operation 2a/(b) Entfernung eines unzugänglichen (isolier-ten) Knotens und seines Bündels, falls $\lambda \geq 0$ ($\lambda \leq 0$); (vgl. Def. 5.4.1)
- Operation 3 Entfernung eines einzelnen abgehenden Zweiges;
- Operation 4 Definition eines einzelnen abgehenden Zweiges;

D. Zweistellige Operationen (primär sind zwei Zustände z_1 und z_j betroffen)

- Operation 5 Doppelung eines abgehenden Gesamt-Bündels (das abgehende Gesamtbündel des Knotens z_1 wird zusätzlich als Bündel des Knotens z_j definiert);
- Operation 6a/(b) Verschiebung eines abgehenden (ankommen-den) Gesamtbündels (das betr. Gesamtbündel des Knotens z_1 wird für z_j definiert und bei z_1 entfernt);
- Operation 7 Doppelung einer Zustands-Eigenschaft (falls $\lambda(z_1) \neq 0$ und $\lambda(z_j) = 0$ erfolgt Wertzuweisung $\lambda(z_j) = \lambda(z_1)$);

E. Algebraische Operation (primär ist ein Knoten z_v betroffen)

Operation 8 Disjunktive, multiplikative oder iterative Dekomposition eines abgehenden nicht-atomaren Zweiges;

Es sei vorgeschlagen, daß sinnvolle Elementarkombinationen der obigen Prüfungen und Operationen zu einfachen Transformationsschleifen programmiert werden nach einem Schema, das in Bild 5.4.1 als Strukturdiagramm dargestellt ist. Diese Programmschleifen erstrecken sich über die Gesamtmenge aller Zustände oder Zustandspaare. Dies bedeutet, daß zwischen zwei Zyklen die gesamte Automatentafel Zwischenergebnis-Charakter annehmen kann, wie beispielsweise bei Bearbeitung indeterministischer Strukturen zweiter Art (1. Schleife: Prozedur 4, 2. Schleife: Prozedur 2, 3. Schleife: Prozedur 3 a). Die folgende Liste zeigt den vorgeschlagenen Satz an Transformationsschleifen für die Synthese aus regulären Ausdrücken. Dieser Satz von Transformationsschleifen enthält außerdem alle zur Reduktion explizit oder implizit vollständig definierter endlicher Automaten erforderlichen Operationen, wobei die Reduktion durch sukzessives Verschmelzen erfolgt /26/.

F. Einstellige Transformations-Schleifen (betroffen ist die Zustandsmenge $\{z_v \mid v = 0, 1, 2, \dots, r\}$).

Prozedur 1 Dekomposition (Transformationen nach Bild 3.2.2 und 3.2.9 b);
(Zyklus: Test 1, Operation 8)

Prozedur 2 Leerzweig-Entfernung (Transformation nach Bild 3.2.6);
(Zyklus: Test 2a, Operationen 5, 3)

Prozedur 3a/(b) Entfernung von unzugänglichen (isolierten) Knoten mit $\lambda = 0$ (zusammen mit vorausgehender Prozedur 1: Transformationen nach Bild 3.2.5);
(Zyklus: Teste 5a(b), 4b(a), Operation 2a(b),)

Prozedur 4 Determinisierung vom Typ 2; nur Schritt 1 nach Bild 3.2.1.3 (Schritt 2 durch nachfolgende Prozeduren 2 und 3));
(Zyklus: Test 3, Operationen 1, 4, 3a, 3a)

G. Zweistellige Transformationschleifen (betroffen ist die Menge $\{(z_1, z_j) \mid 1 = 0, 1, \dots (r-1); j = (1+1), \dots r\}$ der Zustandspaare).

Prozedur 5a/(b) Vorwärts-(Rückwärts-)Minimisierung zusammen mit nachfolgender Prozedur 3a(b); Transformationen nach Bild 3.2.1 a (b));
(Zyklus: Test 6a(b), Operation 6b(a),)

Prozedur 6 Doppelung von Knoten-Eigenschaften (Regel 3.2.1);
(Zyklus: Tests 2a(b), 4b(a), Operation 7)

Gemäß den in obiger Liste eingetragenen Bild-Hinweisen läßt sich im allgemeinen leicht zeigen, daß sich der gesamte Satz von Transformationsregeln nach Abschnitt 3 durch diese Transformations-Schleifen ersetzen läßt. Die Determinisierung durch Prozeduren 4, 2, und 3a wird wegen eines scheinbaren Umweges wie folgt erklärt.

Definition 5.4.1: Ein Knoten ist *unzugänglich (isoliert)*, wenn er keinen einzigen ankommenden (abgehenden) Zweig besitzt. /49/.

Wird zur Leerzweig-Entfernung die Regel zur verschmelzungsfreien Leerzweig-Entfernung (Bild 3.2.6) angewandt, obwohl verschmelzende Leerzweig-Entfernung (Bild 3.2.5) zulässig ist, so entsteht ein unzugänglicher Knoten (Knoten 2 in Bild 3.2.6). Ein solcher kann durch Prozedur 3 a aus obiger Liste gelöscht werden. Bei Verfügbarkeit und späterer Anwendung von Prozedur 3a wird folglich nur noch eine einzige Prozedur für die Leerzweig-Entfernung benötigt, was bei der Zusammenstellung obigen Satzes von Transformationen ausgenützt wurde.

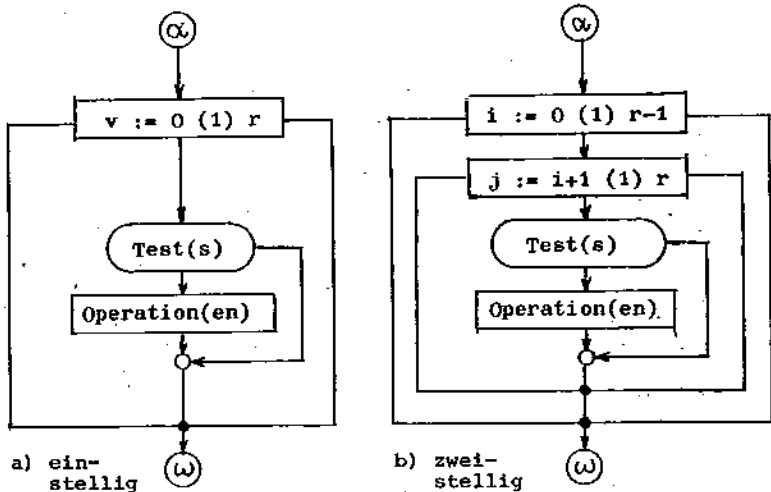


Bild 5.4.1 Aufbauschema für Transformations-Schleifen

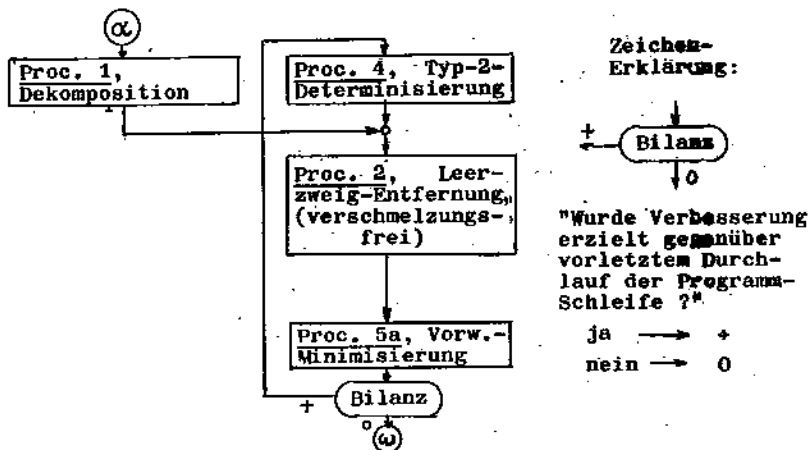


Bild 5.4.2 Beispiel für eine Strategie zur Simulation der Synthese von Minimalautomaten

Unter Zugrundelegung der obigen Liste von Transformations-Schleifen wird folgende Strategie für einen Synthese-Algorithmus auf der Grundlage der Suchlistendarstellung vorgeschlagen, die als Strukturdiagramm in Bild 5.4.2 dargestellt ist. Nach Satz 5.3.1 ist die Umwandlung vom unechten in ein atomares Zustandsdiagramm mit Leerzweigen durch Prozedur 1 möglich. Prozedur 2 erlaubt die Entfernung sämtlicher Leerzweige. Der so entstandene Automat ist in der Regel nicht minimal und kann durch Prozedur 5a reduziert werden. Bei komplexeren Syntheseproblemen auftretende Mehrdeutigkeiten (vgl. Abschnitt 4 beispielsweise) werden durch Prozedur 4 beherrscht.

Literaturhinweise

- /1/ Steinbuch, K. "Automatische Zeichenerkennung" *Taschenbuch der Nachrichtenverarbeitung*, Springer-Verlag Berlin/Heidelberg/New York 1967 (2. Auflage)
- /2/ Steinbuch, K. Kazmierczak, H. "Grundlagen und Anwendung der automatischen Zeichenerkennung", *Jahrbuch des elektr. Fernmeldewesens*, 1962
- /3/ Kazmierczak, H. Steinbuch, K. "Adaptive Systems in Pattern Recognition", *IEEE-Transactions on EC-12* (1963)
- /4/ Kazmierczak, H. "Einführung in die Automatische Zeichenerkennung", Oldenbourg-Verlag München/Wien, 1969
- /5/ Kazmierczak, H. Holdermann, F. Hartenstein, R. "Verarbeitung, Erkennung und Erzeugung von Zeichenmustern", in: *Nichtnumerische Informationsverarbeitung*, (Hrsg. R. Gunzenhäuser), Springer-Verlag Wien/New York 1968
- /6/ Steinbuch, K. "Die Lernmatrix", *Kybernetik* 1 (1961), H. 1
- /7/ Alzerman, M.A. "Lernvorgänge bei der Erkennung von Zeichenklassen", in *Neuere Ergebnisse der Kybernetik* (Hrsg. K. Steinbuch, S.W. Wagner), Oldenbourg-Verlag München, 1964
- /8/ Arkadjew, A.G. Brawerman, E.M. "Zeichenerkennung und Maschinelles Lernen" (übers. a. d. Russischen), Oldenbourg-Verlag München/Wien, 1966
- /9/ Wendt, S. "Trainingsmethoden zur stückweise linearen Separierung mit der Lernmatrix", Dissertation an der Universität Karlsruhe, 1968
- /10/ Groner, G.F. "Real-Time Recognition of Hand-Printed Text", *AFIPS Conf. Proc.*, 29, 1966 FJCC
- /11/ Licklider, J.C.R. "Man-Computer Symbiosis" *IRE Transaction on HFE-1*, 1960
- /12/ Steinbuch, K. "Die Informierte Gesellschaft", DVA Stuttgart, 1966

- /13/ Davis, M.R. "The RAND Tablet: A Man-Machine Graphical Communication Device" AFIPS Conf. Proc. 26, 1964 PJCC
- /14/ Stotz, R. "Man-Machine Console Facilities for Computer-Aided Design", AFIPS Conf. Proc. 23, 1963 SJCC
- /15/ Behnke, H.
Tietz, H. (Hrsg.) "Mathematik I", *Fischer-Lexikon*
Fischer-Bücherei, Frankfurt 1966
- /16/ Nilson, N.J. "Learning Machines", McGraw Hill,
New York 1965
- /17/ Sebestyen, G.S. "Practical Limitations of Pattern
Recognition Research", Int. Conf.
on Methodologies of Pattern Recognition,
University of Hawaii, Honolulu,
Jan. 1968
- /18/ Munson, J.H. "Some Views on Pattern-Recognition
Methodology", Manuskript eines Vortrages
zu: Internat. Conf. on Methodologies
of Pattern-Recognition,
Univ. of Hawaii, Honolulu, Jan. 1968
- /19/ Brick, D.B. "Pattern Recognition, the Challenge,
are we meeting it?", Vortrag in:
Int. Conf. on Methodologies of Pattern
Recognition, Univ. of Hawaii,
Honolulu, Jan. 1968
- /20/ Bernstein, M.I.
Williams, T.G. "An interactive Programming System
for the Casual User", Computers and
Automation, Febr. 1968
- /21/ Aiserman, M.A.
et al. "Logik, Automaten, Algorithmen"
(Übers. a. d. Russischen), Oldenbourg-
Verlag München/Wien, 1967
- /22/ Gluschkow, W.M. "Theorie der Abstrakten Automaten"
(Übers. a. d. Russischen), VEB Verlag
der Wissenschaften, Berlin 1963
- /23/ Schulte, D. "Kombinatorische und Sequentielle
Netzwerke", Oldenbourg-Verlag München/
Wien, 1967
- /24/ Pöllinger, O.
Weber, W. "Methoden der Schaltalgebra", Olden-
bourg-Verlag München/Wien, 1967
- /25/ Menzel, W. "Einführung in die Automatentheorie",
Vorlesung an der Universität Karlsruhe,
gehalten im WS 1968/69

- /26/ Gill, A. "Introduction to the Theory of Finite-State Machines", McGraw-Hill Publ., New York/Toronto/London 1962
- /27/ Nadler, M. "An Analog-Digital Character Recognition System", IEEE Transactions on EC-12, 1963
- /28/ Hosking, K.H.
Thompson, K. W. "A Feature Detection Method for Optical Character Recognition", Proc. IEE Conf. on Pattern Recognition, Teddington, Middx., July 1968
- /29/ Grimsdale, R.L.
Bullingham, J.H. "Character Recognition by Digital Computer Using a Special Flying-Spot Scanner", The Computer Journal 4, 1961
- /30/ Perotto, P.G. "A New Method for Automatic Character Recognition", IEEE Transactions on EC-12, Oct. 1963
- /31/ Gattner, G. "Automatische Ziffern-Erkennung und ihre Anwendung beim Auswerten von Gesprächszähler-Fotografien", Siemens-Inf. Fernspr.-Vermittlungstechnik, 2-65 (Aug. 1965)
- /32/ Greanias, F.C.
Meagher, P.F.
Norman, R.J.
Essinger, P. "The Recognition of Handwritten Numerals by Contour Analysis", IBM - Journal of Research and Development 7, 1963, H. 1
- /33/ McNaughton, R.
Yamada, H. "Regular Expressions and State Graphs for Automata", IRE Trans. on EC-9, 1960
- /34/ Brzozowski, J.A.
McCluskey jr., E.J. "Signal Flow Graph Techniques for Sequential Circuit State Diagrams", IRE Transactions on EC-12, 1963
- /35/ Kleene, S.C. "Representation of Events in Nerve Nets and Finite Automata", in *Automata Studies*, Annals of Math. Studies Princeton Univ. Press, Princeton, N.J. 1956
- /36/ Brzozowski, J.A. "A Survey of Regular Expressions and their Applications", IRE Trans. on EC-11, 1962
- /37/ Mason, S.J. "Feedback Theory - Some Properties of Signal Flow Graphs", Proc. IRE 41, 1953

- /38/ Mason, S.J. "Feedback Theory - Further Properties of Signal Flow Graphs", Proc. IRE 44, 1956
- /39/ Seshu, S.
Reed, M.B. "Linear Graphs and Electrical Networks", Addison Wesley Pub., Reading, Mass. 1961
- /40/ Copi, I.M.
Elgot, C.C.
Wright, J.B. "Realization of Events by Logical Nets", J.A.S.C.M. 5, 1958
- /41/ Gluschkow, W.M. "Über einen Algorithmus zur Synthese abstrakter Automaten" (russisch). U. M. Zh-12, 1960
- /42/ Glushkov, V.M. "Introduction to Cybernetics" (Übers. a. d. Russischen ins Englische), Academic Press New York/London, 1966
- /43/ Frishkopf, L.S.
Harmon, L.D. "Machine Reading of Cursive Script" in: *Information Theory* (Ed. Colin Cherry), Butterworths, London 1961
- /44/ Paul, H.-D. "Untersuchungen zur Erkennung eines begrenzten Repertoires kursiv geschriebener Worte", Diplomarbeit am Institut für Nachrichtenverarbeitung und Nachrichtenübertragung der Universität Karlsruhe 1966
- /45/ Lipp, H.M. "Entwurf eines geschlossenen adaptiven Systems für diskrete deterministische Prozesse", Dissertation an der Universität Karlsruhe, 1967
- /46/ Ott, G.H.
Feinstein, N.H. "Design of Sequential Machines from their Regular Expressions", JACM 8 (1961)
- /47/ Binder, G. "Simulation eines Erkennungsverfahrens für handgeschriebene Zeichen mit automatisiertem Klassifikator-Entwurf", Diplomarbeit am Institut für Nachrichtenverarbeitung und Nachrichtenübertragung der Universität Karlsruhe, 1968
- /48/ Fogel, L.J.
Owens, A.J.
Walsh, J.W. "Artificial Intelligence through simulated evolution", J. Wiley and Sons, New York/London/Sidney 1966
- /49/ Nelson, R.J. "Introduction to Automata" J. Wiley and Sons, New York/London/Sidney 1968

- /50/ Riebel, H. "Untersuchungen von Beschreibungsformen für komplexe Schaltwerke", Diplomarbeit am Institut für Nachrichtenverarbeitung und Nachrichtenübertragung der Universität Karlsruhe, 1968
- /51/ Miller, R.E. "Sequential Circuits and Machines" in: *Switching Theory, Vol. 2*, Wiley and Sons, New York 1965
- /52/ Brzozowski, J.A. "Derivatives of Regular Expressions", J.A.C.M 11 (1964), 481-494
- /53/ Rabin, M.O.
Scott, D. "Finite Automata and their decision problems", IBM J. Res. Dev. 3 (1959) 114-125
- /54/ Perles, M.
Rabin, M.O.
Shamir, E. "The Theory of Definite Automata" IEEE-Trans. EC-12 (1963), 233-243
- /55/ Paz, A.
Peleg, B. "Ultimate-Definite and Symmetric-Definite Events and Automata", JACH 12 (1965), 399-410
- /56/ Moore, E.F. "Gedanken Experiments on Sequential Machines", in *Automata Studies*, Ann. of Math. Studies No. 34, Princeton Univ. Press, 1956, 129-153
- /57/ Fischer, K. "Methoden der Formelübersetzung" in: *Nicht-numerische Informationsverarbeitung*, (Hrsg. R. Gunzenhäuser), Springer-Verlag, Wien/New York 1968
- /58/ Zorn, W. "An Algorithm for Adapting Threshold Elements by a Digital Computer", Int. Conf. on Pattern Recognition, Grenoble, 11.-13. Sept. 1968
- /59/ Hartenstein, R. "Darstellung, Synthese, und Minimierung von Automaten mit Hilfe eines Digitalrechners", Seminarvortrag am Institut für Nachrichtenverarbeitung und Nachrichtenübertragung der Universität Karlsruhe, 27. Okt. 1967
- /60/ Busacker, R. G.
Saaty, T. L. "Endliche Graphen und Netzwerke", Oldenbourg-V., München/Wien 1968
- /61/ Hartenstein, R. "Sequentielle Verfahren zur Automatischen Zeichenerkennung", Seminarvortrag am Insitut für Nachrichtenverarbeitung und Nachrichtenübertragung der Universität Karlsruhe, 11. Nov. 1966

D a n k s a g u n g

Herrn Prof. Dr.-Ing. K. Steinbuch danke ich für die sehr interessante Themenstellung und für sein stetes Interesse an der Entwicklung der Arbeit, sowie für wichtige Hinweise.

Herrn Prof. Dr. rer. nat. K. Nickel bin ich für die Übernahme des Korreferates und die Durchsicht der Arbeit zu Dank verpflichtet.

Herrn Dr.-Ing. H. Kazmierczak danke ich für seine Kritik. Herrn Priv.-Doz. Dr. rer. nat. W. Menzel und Herrn Dipl.-Ing. W. Zorn danke ich für anregende Diskussionen.

Frl. G. Schuster gilt mein Dank für ihre Bemühungen um die Reinschrift des Manuskriptes.

Lebenslauf

18. 12. 1934 geboren in Berlin als Sohn des Obering.
Jacob Hartenstein
- 1941 - 1946 Volksschulen in Berlin und Heidelberg
- 1946 - 1950 Oberschule in Berlin
- 1950 - 1954 Realgymnasium in Bruchsal/Baden,
Schulabschluß durch Reifeprüfung
- 1954 - 1959 Studium der Nachrichtentechnik an der
Universität Karlsruhe,
Nebenfach: Lichttechnik
Studienabschluß: Hauptdiplom
- Praktika:
Siemens AG, Werk Bruchsal (Vermittlungs-
technik), Deutsche Philips GmbH, Werk
Krefeld-Linn (Fernsehtechnik)
- 1960 - 1965 Entwicklungs-Ingenieur für Elektronik am
Kernforschungszentrum Karlsruhe,
1962 Informationsreise in die U.S.A.
ab 1962 stellv. Strahlenschutzverantwortlicher
auf Abteilungsebene
ab 1963 Gruppenführer
- seit 1965 wissenschaftlicher Mitarbeiter am Institut
für Nachrichtenverarbeitung und Nachrichten-
übertragung der Universität Karlsruhe (Dir.:
Prof. Dr.-Ing. K. Steinbuch)

Offsetdruck-Fötodruck

1. Hohl, 8520 Erlangen Hauptstr. 109, (Nähe Martin-Luther-Platz) Tel. 22990