

2. Theoretische Grundlagen

2.1 Die KLEENE'sche Algebra

2.1.1 Die Axiomatik der KLEENE'schen Algebra

Die bekannteste formale Sprache zur Darstellung von Ereignissen ist die von KLEENE angegebene "Sprache der regulären Ereignisse". Diese Sprache ist eine algebraische Struktur, weshalb sie als "KLEENE'sche Algebra" bezeichnet sei. In dieser Algebra sind zwei elementare Operationen definiert. Die Ausdrücke der KLEENE'schen Algebra heißen "reguläre Ausdrücke". (Ein Ausdruck einer Algebra ist eine Verknüpfung von Elementen dieser Algebra. Ein Ausdruck ist atomar, wenn er nicht zerlegbar ist.) Ein Ereignis ist dann und nur dann ein "reguläres Ereignis", wenn es einen Ausdruck der KLEENE'schen Algebra d. h. einen "regulären Ausdruck" dazu gibt. Wichtig für Probleme der abstrakten Synthese sind die beiden folgenden Sätze von KLEENE:

1. KLEENE'scher Satz: "Für jedes reguläre Ereignis existiert ein endlicher Automat, der dieses Ereignis darstellt."

2. KLEENE'scher Satz: "Durch endliche Automaten sind nur reguläre Ereignisse darstellbar."

(Auf den Begriff der "Darstellung" wird später ausführlicher eingegangen.) Die Folgerung daraus ist, daß das abstrakte Syntheseproblem für die Darstellung aller derjenigen Ereignisse lösbar ist, die durch reguläre Ausdrücke beschrieben werden können. Die Unterscheidung von Ereignissen und regulären Ereignissen ist für die Anwendung nicht von Bedeutung. Bei praktischen Syntheseproblemen treten stets nur reguläre Ereignisse auf. Im Folgenden wird eine axiomatische Einführung der KLEENE'schen Algebra gegeben werden unter gleichzeitiger Interpretation der KLEENE'schen Operationen im Gebiet der regulären Mengen.

Gegeben seien als atomare Ausdrücke für die Menge aller

Interpretation: Die Disjunktion $\gamma = \alpha \cup \beta$ zweier regulärer Ausdrücke α und β bezeichnet genau diejenigen Elemente, die Mitglied der regulären Mengen α oder β sind.

Beispiel: Wenn $\alpha = \{\langle a \rangle, \langle b, a \rangle\}$
und $\beta = \{\langle a \rangle, \langle c, a \rangle\}$

dann folgt: $\alpha \cup \beta = \{\langle a \rangle, \langle b, a \rangle, \langle c, a \rangle\}$

Die KLEENE'sche Disjunktion entspricht also genau den aus der BOOLE'schen Algebra und der Mengen-Algebra bekannten Operationen der Disjunktion. Ähnlich, wie in der Mengen-Algebra und der BOOLE'schen Algebra, sei die folgende abkürzende Schreibweise für disjunktive Vielfach-Verknüpfungen eingeführt.

Schreibweise:

$$\alpha_1 \cup \alpha_2 \cup \alpha_3 \cup \dots \cup \alpha_n = \bigcup_{j=1}^n \alpha_j$$

Diese zusammenfassende Schreibweise wird später bei der Definition einer dritten, abgeleiteten Operation verwendet.

Die zweite elementare Operation, die über der Menge aller regulären Ausdrücke definiert ist, wird wie folgt definiert.

Definition 2.1.2: Das *Produkt* (oder auch *Kettenprodukt*) γ zweier regulärer Ausdrücke α und β :

$$\gamma = \alpha \cdot \beta$$

Sprechweise: für $\alpha \cdot \beta$ spricht: α gefolgt von β (oder kürzer: α vor β)

Interpretation: Das Produkt (oder auch "Kettenprodukt") $\gamma = \alpha \cdot \beta$ zweier regulärer Ausdrücke α und β bedeutet die Produktmenge γ aus der Menge von Verkettungen aller geordneten Paare von Elementen der Mengen α und β nach folgender Vorschrift:

$$\gamma = \alpha \cdot \beta = \bigcup_{\substack{u \in M(\alpha) \\ v \in M(\beta)}} \langle u, v \rangle$$

Das Operator-Symbol für die Vielfach-Disjunktion wurde hier mit doppelter Kontur ausgeführt, um diese Operation, - als zur Interpretationsebene der regulären Mengen gehörig - von der Disjunktion im Definitionsbereich der regulären Ausdrücke, d.h. in der KLEENE'schen Algebra zu unterscheiden.

Beispiel: Wenn $\alpha = \{ \langle a \rangle, \langle b, a \rangle \}$

und $\beta = \{ \langle a \rangle, \langle c, a \rangle \}$

dann folgt $\alpha \cdot \beta = \{ \langle a, a \rangle, \langle a, c, a \rangle, \langle b, a, a \rangle, \langle b, a, c, a \rangle \}$

Andere Schreibweise: Das Multiplikationszeichen des Kettenproduktes kann weggelassen werden. Beispielsweise für $(\alpha \cdot \beta)$ ist $\alpha\beta$ die abgekürzte Schreibweise. Durch Einführung einer Priorität für die Multiplikation kann hierbei auch ein Klammern-Paar eingespart werden.

Für die Bezeichnung von Vielfach-Produkten gleicher Operanden sei die Bezeichnung *Potenz* (oder *Ketten-Potenz*) eingeführt unter Vereinbarung folgender Schreibweise:

Wenn α ein regulärer Ausdruck ist, dann bedeutet die (Ketten-)Potenz α^i die multiplikative Verknüpfung von genau i regulären Ausdrücken α , wie folgt:

$$\alpha^i = \underbrace{\alpha\alpha\alpha\alpha\dots\alpha}_{i\text{-mal}}$$

daraus folgt:

$$\alpha^0 = \Lambda, \quad \alpha^1 = \alpha, \quad \alpha^{n+1} = \alpha^n \cdot \alpha$$

Beispiel: Für den atomaren Fall mit $a = a$ ergeben sich für obige Schreibweise die folgenden Interpretations-Beispiele:

$$\begin{aligned} a^0 &= \langle \rangle; & a^1 &= \langle a \rangle; & a^2 &= \langle a, a \rangle; \\ a^n &= \langle a, a, a, a, \dots, a, a \rangle, \text{ worin } n\text{-mal} \\ &\text{das Element } a \text{ vorkommt.} \end{aligned}$$

In Tafel 2.1.1 ist das Axiomen-System der KLEENE'schen Algebra schematisch wiedergegeben. Tafel 2.1.2 zeigt die Formulierungen der Axiome der KLEENE'schen Algebra.

Durch die bis hier eingeführten Symbole, Definitionen und Axiome ist das tragende Gerüst der KLEENE'schen Algebra vollständig formuliert. Alle sonstigen Sätze und Regeln sind Ableitungen.

Die Tatsache, daß die Multiplikation nicht idempotent und nicht kommutativ ist, kann leicht daraus erklärt werden, daß reguläre Ausdrücke Deskriptoren für Mengen von Folgen sind (vgl. "Beispiel" auf Seite 23 und 24). Eine Folge der Länge 2 kann z.B. nicht gleich einer Folge der Länge 1 sein (Relation 1 b in Tafel 2.1.2). In einer Folge führt z.B. ein Vertauschen von Elementen i. allg. zu einer nicht äquivalenten neuen Folge (Relation 3 b). Auch Relation 5 b in Tafel 2.1.2 läßt sich leicht anhand der Interpretation als Deskriptor für Folgen zeigen. Letztlich lassen sich alle Strukturunterschiede zwischen der BOOLE'schen Algebra und der KLEENE'schen Algebra darauf zurückführen, daß die BOOLE'sche Algebra Mengen beschreibt, wo beispielsweise gilt:

$$\{a, b\} = \{b, a\} \quad \text{und:} \quad \{a, a\} = \{a\},$$

während die KLEENE'sche Algebra Folgen beschreibt, wo z.B. gilt:

$$\langle a, b \rangle \neq \langle b, a \rangle \quad \text{und:} \quad \langle a, a \rangle \neq \langle a \rangle$$

Axiom	a) Disjunktion	b) Multiplikation
1. Idempotenz-Gesetz (Gesetz der Tautologie)	gilt: $a \vee a = a$	gilt nicht: $a \cdot a \neq a$
2. Assoziativ-Gesetz	gilt: $(a \vee b) \vee \gamma = a \vee (b \vee \gamma)$	gilt: $(a \cdot b) \cdot \gamma = a \cdot (b \cdot \gamma)$
3. Kommutativ-Gesetz	gilt: $a \vee b = b \vee a$	gilt nicht: $a \cdot b \neq b \cdot a$
4. Adjunktiv-Gesetz (Absorptionsgesetz)	gilt nicht	gilt nicht
5. Distributiv-Gesetz	gilt: $a \vee (b \cdot \gamma) = (a \vee b) \cdot (a \vee \gamma)$	gilt nicht: $a \cdot (b \vee \gamma) \neq (a \cdot b) \vee (a \cdot \gamma)$
6. Komplementär-Gesetz	gilt nicht	gilt nicht
7. Null-Element	gilt nicht	gilt: $\phi \cdot \gamma = \gamma \cdot \phi = \phi$
8. Eins-Element	gilt: $\gamma \vee \phi = \phi \vee \gamma = \gamma$	gilt: $A \cdot \gamma = \gamma \cdot A = \gamma$

Tafel 2.1.2 Axiome der KLEENE'schen Algebra

Ein weiterer wesentlicher Unterschied besteht in der anschließend eingeführten abgeleiteten Iterations-Operation, zu der es in der BOOLE'schen Algebra kein Analogon gibt.

2.1.2 Die Iterations-Operation

Von KLEENE wird neben den bereits eingeführten beiden Operationen eine weitere, Iteration genannte Operation eingeführt, die aus den beiden ersteren Operationen abgeleitet werden kann:

Definition 2.1.3: Die Iteration a^* eines regulären Ausdruckes a sei erklärt durch folgende Vorschrift:

$$a^* = \bigcup_{j=0}^{\infty} a^j = \Lambda \vee a^1 \vee a^2 \vee a^3 \vee \dots$$

Von Interesse sind noch die folgenden abgeleiteten Regeln /22,35,36/:

Regel 2.1.1 $\Lambda^* = \Lambda$

Regel 2.1.2 $\Phi^* = \Lambda$

Regel 2.1.3 $a^* a^* = a^*$

Regel 2.1.4 $\Lambda a^* = a^*$

Regel 2.1.5 $(a^*)^* = a^*$

Regel 2.1.6 $(a^* b^*)^* = (a \vee b)^*$

Regel 2.1.7 $a^* = \Lambda \vee a a^*$

Abschließend sei mittels eines einfachen Beispielles die Beschreibung einer Menge von Folgen durch einen regulären Ausdruck ρ gezeigt. Der reguläre Ausdruck $\rho = b(cd^*a^*)$ ist Deskriptor für eine Menge, die untenstehende Folgen umfaßt:

$$\rho = \{ \langle b, c, d \rangle, \langle b \rangle, \langle b, a \rangle, \langle b, a, a \rangle, \langle b, a, a, a \rangle, \dots \langle b, a, a, a, \dots \rangle \},$$

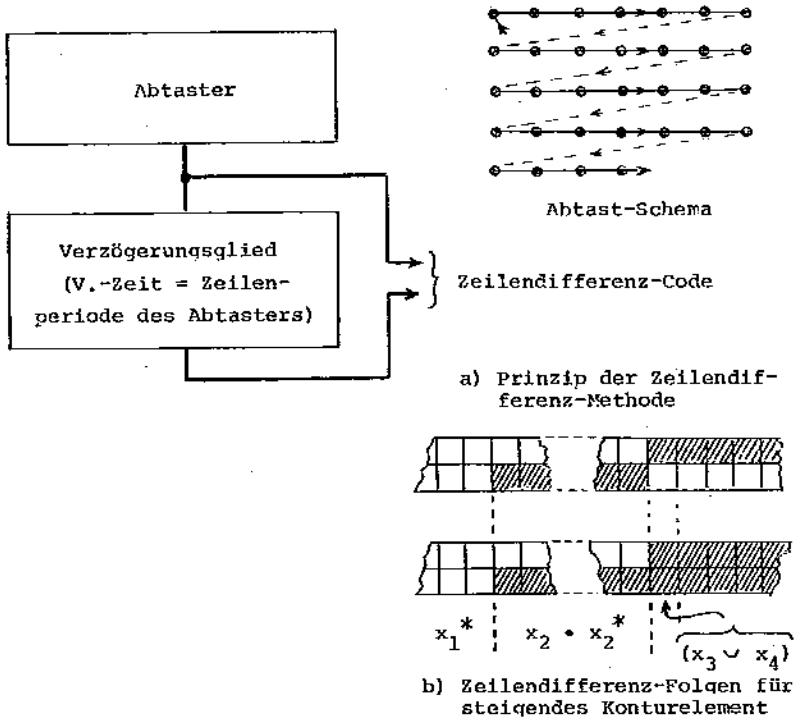


Bild 2.1.1 Kontur-Erfassung nach der Zeilendifferenz-Methode

Symbol	Code-Wort	Bedeutung
x_1	00	
x_2	01	
x_3	10	
x_4	11	
x_5		Zeilen-Ende

Tafel 2.1.3 Alphabet bei Zeilendifferenz-Abtastung einer Binär-Matrix

x	x_1	x_2	x_3	x_1	x_m
z_v					
z_1					
z_2					
z_3					
z_j				$\delta(z_j, x_1)$	
z_r					

Tafel 2.2.1 Übergangstabelle in allgemeiner Form

Die Richtigkeit dieser Folge ist vielleicht leichter einzusehen, wenn die disjunktive Normalform $\rho = bcd \cup ba^*$ betrachtet wird.

Zur Veranschaulichung diene ein Beispiel aus der Automatischen Zeichenerkennung. Ein schwarzes Linienmuster auf weißem Hintergrund, welches quantisiert in Form einer Binärmatrix vorliegt, soll zur Erkennung steigender Kontur-elemente nach der Zeilendifferenz-Methode abgetastet werden /29/. Das Prinzip der Zeilendifferenz-Methode läßt sich veranschaulichen durch eine Maske mit einem zwei übereinanderliegende Rasterelemente zeigenden Fenster, das zeilenweise über die Binär-Matrix bewegt wird. Dabei entsteht eine Folge von 2-bit-Worten, die mit Hilfe des Alphabetes $\{x_1, x_2, x_3, x_4\}$ dargestellt werden können. Durch Hinzufügen des Symbols x_5 für "Zeilenende" ergibt sich das Alphabet $X = \{x_1, x_2, x_3, x_4, x_5\}$ mit Bedeutungen gemäß Tafel 2.1.3. Wie sich mit Hilfe von Bild 2.1.1 leicht zeigen läßt, ist $\rho = x_1^* x_2 \cdot x_2^* \cdot (x_3 \cup x_4)$ der gesuchte reguläre Ausdruck. Der reguläre Ausdruck $\rho = x_1^* x_2^* (x_3 \cup x_4)$ wäre falsch, da er neben der Bedeutung "steigend" auch noch die Bedeutung "senkrecht" einbeziehen würde.

2.2 Der abstrakte Automat als Deskriptor von Ereignissen

In diesem Kapitel sollen die wichtigsten Definitionen der Automatentheorie eingeführt sowie einige wichtige Sätze zitiert werden /21-26,49/.

Definition 2.2.1: Der endliche, diskrete Automat sei definiert als ein Quintupel $(X, Y, Z, \delta, \lambda)$ (Bild 2.2.1), das aus den drei endlichen nichtleeren Alphabeten X, Y, Z und den zwei Abbildungen δ, λ besteht.

Hierbei ist:

X die Menge aller Eingabesymbole $X = \{x_1, x_2, \dots, x_m\}$

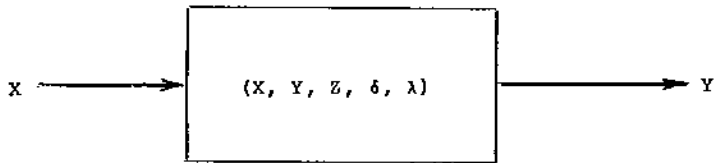


Bild 2.2.1 Black Box - Modell des abstrakten Automaten.

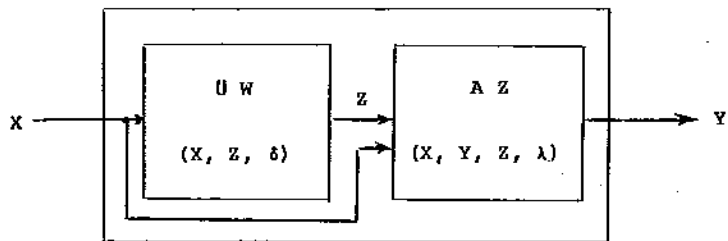


Bild 2.2.2 Die Funktions-Einheiten des abstrakten Automaten.

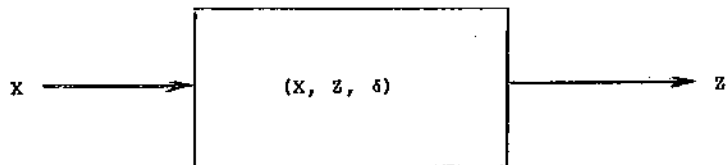


Bild 2.2.3 Abstrakter Automat ohne Ausgangs-Zuordner.

- Y die Menge aller Ausgabesymbole $Y = \{y_1, y_2, \dots, y_n\}$
- Z die Menge aller inneren Zustände $Z = \{z_1, z_2, \dots, z_r\}$
- δ die Übergangsfunktion
- λ die Ausgangsfunktion

Ein Automat kann in zwei Funktionsgruppen zerlegt werden. Diese sind das Überführungswerk (ÜW), definiert durch die Funktion δ , und der Ausgangszuordner (AZ), welcher durch die Funktion λ definiert ist (Bild 2.2.2). Das "Darstellen" oder "Akzeptieren" von Eingabefolgen durch einen Automaten erfolgt ausschließlich im Überführungswert, wie noch gezeigt wird. Für die folgenden Betrachtungen genügt daher ein Automat ohne Ausgangszuordner, welcher ein Tripel (X, Z, δ) ist (Bild 2.2.3).

Die Abbildung δ kann in Form einer "Übergangstabelle" wiedergegeben werden (Tafel 2.2.1). Die Anzahl der Spalten ist durch die Länge m des Eingangsalphabetes gegeben und die Anzahl der Zeilen ist durch die Anzahl r der möglichen Zustände des Automaten bestimmt.

Jeder spezielle Wert $\delta(z_j, x_i) \in Z$ in einer solchen Übergangstabelle heißt "Folgezustand", wobei z_j der vorhergehende Zustand ("Vorzustand") und x_i das den Übergang in den Folgezustand auslösende Eingabesymbol ist. Ein Automat heißt dann "vollständig definiert", wenn zu jedem Paar (z_j, x_i) ein Folgezustand existiert, d.h. jeder Platz in der Übergangstabelle besetzt ist. Ein Automat heißt "unvollständig definiert", wenn nicht zu jedem Paar (z_j, x_i) ein Folgezustand existiert, d.h. nicht jeder Platz in der Übergangstabelle besetzt ist. Ein Automat heißt "deterministisch", wenn zu jedem Paar (z_j, x_i) höchstens ein Folgezustand definiert ist. Er heißt "nichtdeterministisch", wenn es Paare (z_j, x_i) gibt, zu denen mehr als ein Folgezustand $\delta(z_j, x_i) \in Z$ existiert, d.h., wenn in der Übergangstabelle Plätze vorkommen, die mit mehr als einem Wert

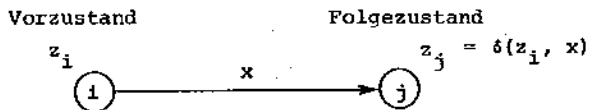


Bild 2.2.4 Darstellung eines Überganges durch einen gerichteten Zweig in einem Graphen.



Bild 2.2.5 Spezialfall eines Überganges mit $z_i = \delta(z_i, x)$ (Schlinge).

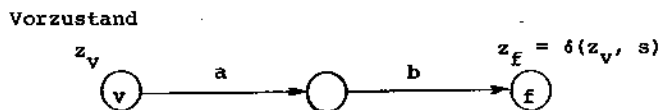


Bild 2.2.6 Darstellung eines verallgemeinerten Überganges aufgrund einer Folge $s = \langle a, b \rangle$ durch einen Pfad in einem Zustandsgraphen.

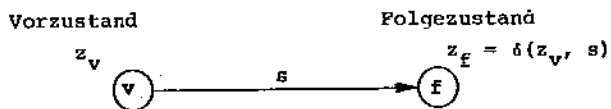


Bild 2.2.7 Vereinfachte Darstellung eines Pfades in uneigentlichen Zustandsgraphen ($s = \text{Folge}$).

besetzt sind.

Eine andere und weit anschaulichere Möglichkeit zur Beschreibung eines Automaten bietet der "Zustandsgraph". Hierbei handelt es sich um einen gerichteten Graphen, in welchem Zustände durch Knoten und Übergänge durch gerichtete Zweige repräsentiert sind. Bild 2.2.4 zeigt das Prinzip der Darstellung eines Überganges durch ein Eingabesymbol x von Vorzustand z_i zum Folgezustand z_j . Bild 2.2.5 zeigt einen entsprechenden Graphen für den Spezialfall $z_i = z_j$. Ein Zustandsgraph enthält genau einen Knoten für jeden Zustand des dargestellten Automaten, sowie genau einen gerichteten Zweig für jeden definierten Übergang.

Durch rekursive Anwendung der Funktion δ kann der Begriff "Folgezustand" verallgemeinert werden derart, daß $\delta(z_0, s)$ auch dann "Folgezustand" genannt wird, wenn s eine Folge über dem Eingabealphabet X ist und z_0 der Vorzustand vor Empfang dieser Folge s ist. Folgendes Beispiel soll dies demonstrieren:

Beispiel: für $s = \langle a, b \rangle$ folgt: $\delta(z_0, s) = \delta(\delta(z_0, a), b)$

Die graphische Darstellung eines solchen verallgemeinerten Überganges (Bild 2.2.6) ist eine Kette von Zweigen (auch "Pfad" genannt), die vom Vorzustand z_0 zum Folgezustand $z_f = \delta(z_0, s)$ führt. Das graphische Analogon zur abgekürzten Schreibweise $\delta(z_0, s)$ ist die vereinfachte Darstellung des entsprechenden Pfades gemäß Bild 2.2.7 durch einen einzigen mit s beschrifteten Zweig, wobei alle Knoten außer z_0 und z_f weggelassen werden. Ein derart aufgebauter Graph eines Automaten heißt "uneigentlicher Zustandsgraph" /46/, da nicht alle Zustände durch Knoten repräsentiert sind. Die Zusammenfassung zu uneigentlichen Zustandsgraphen kann darüber hinaus noch weiter getrieben werden, indem jeweils mehrere Pfade durch einen einzigen Zweig beschrieben werden, der jeweils mit einem regulären Ausdruck beschriftet

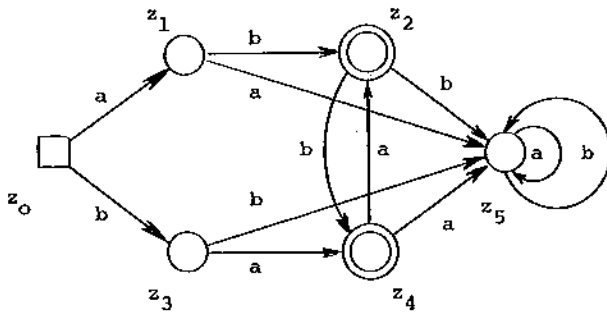


Bild 2.2.8 Zustandsdiagramm eines *deterministischen* Akzeptors.

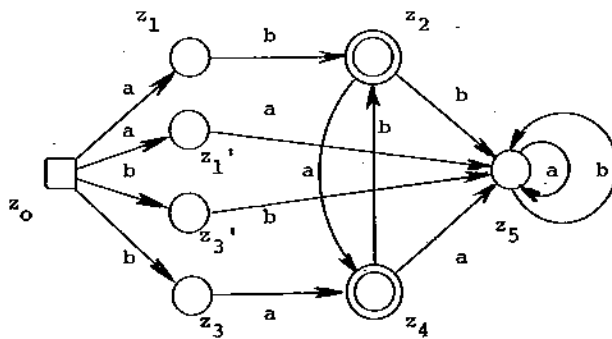


Bild 2.2.9 Zustandsdiagramm eines *nicht-deterministischen* Akzeptors.

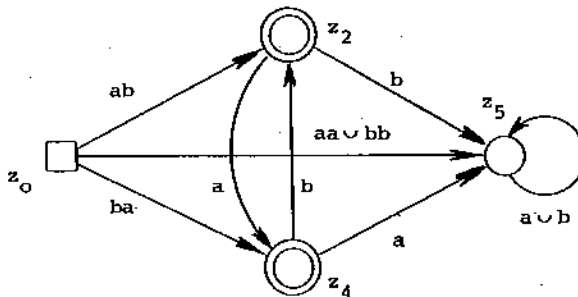


Bild 2.2.10 Zustandsdiagramm eines *uneigentlichen* Akzeptors.

wird.

Bild 2.2.8 bis 2.2.10 zeigen Beispiele für die erwähnten verschiedenen Typen von Zustandsgraphen von (unvollständig definierten) Automaten. Beim Zustandsgraphen eines deterministischen Automaten darf von jedem Knoten nur ein einziger Zweig mit gleichem Symbol abgehen (Bild 2.2.8). Beim Zustandsgraphen eines nicht-deterministischen Automaten muß von mindestens einem Knoten mehr als ein Zweig gleicher Beschriftung ausgehen (vgl. Knoten z_0 in Bild 2.2.9). Beim uneigentlichen Zustandsgraphen dürfen Zweige mit Deskriptoren für Folgen oder für Mengen von Folgen (z.B. mit atomaren oder zusammengesetzten regulären Ausdrücken) beschriftet sein. Bild 2.2.10 zeigt einen solchen Graphen mit atomaren regulären Ausdrücken.

Nun sei der Begriff der "Darstellung" von Folgen durch Automaten eingeführt.

Definition 2.2.2: Eine Eingabefolge s eines Automaten gilt als *dargestellt* durch einen Automaten im Zustand z_0 , wenn zur Folge s ein Folgezustand existiert. Ein Ereignis E gilt als *dargestellt* in z_0 , wenn sämtliche Folgen $s \in E$ in Zustand z_0 dargestellt sind.

Nach dem Begriff *Darstellung* sei der Begriff des *Akzeptierens* von Ereignissen definiert. Das hierzu geeignete Automaten-Modell ist der *Akzeptor* /49, 53/, der wie folgt definiert ist und keinen Ausgangszuordner besitzt.

Definition 2.2.3: Ein *deterministischer Akzeptor* \mathcal{A} ist ein diskreter Automat $\mathcal{A} = (X, Z, z_0, \xi, Z', \delta)$, bei welchem z_0 der einzige vor dem Empfang einer jeden Eingabefolge einzunehmende Anfangszustand, und Z' die Menge aller Endzustände nach Empfang einer jeden Eingabefolge ist. (Wenn keine weiteren Angaben vorliegen, sei unter der Bezeichnung *Akzeptor* von hier ab ein endlicher Akzeptor verstanden.)

Wie für jeden deterministischen Automaten, gilt selbstverständlich auch für einen deterministischen Akzeptor die bekannte Forderung der Eindeutigkeit für jeden definierten Wert $\delta(z_i, x_j) \in Z \cup X$.

Definition 2.2.4: Eine durch einen deterministischen Akzeptor \mathcal{A} empfangene Eingabefolge s gilt genau dann als *akzeptiert durch den Akzeptor \mathcal{A}* , wenn sie der Forderung $\delta(z_0, s) \in Z'$ genügt, d.h. der Folgezustand auf den Anfangszustand z_0 einer der Endzustände aus der Menge Z' ist. Eine Eingabefolge s gilt dann als *zurückgewiesen durch den Akzeptor \mathcal{A}* , wenn für sie gilt: $\delta(z_0, s) \notin Z'$.

Die Menge aller akzeptierten Eingabefolgen ist also im allgemeinen eine echte Untermenge der Menge aller durch den Automaten dargestellten Folgen und heißt: das durch den Akzeptor \mathcal{A} beschriebene "Ereignis" $U(\mathcal{A})$.

Mit Hilfe der Definitionen 2.2.3 und 2.2.4 läßt sich eine Lösung des Erkennungsproblem für Ereignisse wie folgt formulieren. Wenn erkannt werden soll, ob eine Folge s ein Element eines Ereignisses E ist, entwerfe man einen Akzeptor \mathcal{A} derart, daß $E = U(\mathcal{A})$ das durch den Akzeptor \mathcal{A} akzeptierte Ereignis $U(\mathcal{A})$ ist. Unter dieser Voraussetzung gilt eine Folge s dann als *erkannt*, wenn sie vom Akzeptor \mathcal{A} akzeptiert wird. Andernfalls gilt sie als *zurückgewiesen*. An anderer Stelle in der vorliegenden Arbeit wird gezeigt, daß auf der Grundlage dieser Erkennungsfunktion eine Theorie zum Entwurf sequentieller Systeme zur Zeichenerkennung auf der Modellvorstellung des Akzeptors aufgebaut werden kann.

Bild 2.2.8 zeigt den Zustandsgraphen eines deterministischen Akzeptors. Die Knoten der Endzustände sind durch einen Doppelkreis von den übrigen Zuständen unterschieden, während der Knoten des Anfangszustandes z_0 ein Quadrat ist.

Rein formal läßt sich auch ein *nicht-deterministischer* Akzeptor definieren, der oben beschriebene Erkennungsfunktion realisiert. Hierbei kann es jedoch infolge Mehrdeutigkeiten

dazu kommen, daß über mehr als eine Zustandsfolge die gleiche Eingabefolge dargestellt oder akzeptiert wird. Ein nicht-deterministischer Akzeptor kann mehrere Anfangszustände haben.

Definition 2.2.5: Ein *nicht-deterministischer* Akzeptor \mathcal{A} ist ein Quintupel $\mathcal{A} = (X, Z, Z_0 \subseteq Z, Z' \subseteq Z, \delta)$, wobei Z_0 die Menge der Anfangszustände und Z' die Menge der Endzustände ist. Die Funktion δ ist nicht eindeutig, die Erkennungsfunktion hingegen eindeutig.

Im Falle eines Automaten mit nicht-eindeutiger Erkennungsfunktion wird als die Bezeichnung *Akzeptor* nicht zugelassen. Ein für Syntheseprobleme wichtiger Akzeptor-Typ ist der nachfolgend definierte sog. *einfache Akzeptor* /53/.

Definition 2.2.6: Ein deterministischer oder nicht-deterministischer Akzeptor $\mathcal{A} = (X, Z, \delta, \{z_0\}, \{z_f\})$ ist *einfach* genau dann, wenn er nur einen einzigen Anfangszustand z_0 und nur einen einzigen Endzustand z_f besitzt, und dabei z_0 nur Folgezustand und z_f nur Folgezustand ist, sowie $z_f \neq z_0$ gilt.

Im Zustandsgraphen eines einfachen nicht-deterministischen Akzeptors werden Anfangs- und Endzustand jeweils durch Extremalknoten dargestellt (z_0 als Anfangsknoten, z_f als Endknoten). Weiter gilt der folgende für den Beweis von Syntheseverfahren aufgrund regulärer Ausdrücke wichtige Satz /53/:

Satz 2.2.1: Wenn ein nicht-deterministischer oder deterministischer Akzeptor $\mathcal{A} = (X, Z_A, Z_{A0}, \delta_A, Z_A')$ die Menge $U(\mathcal{A}_A)$ akzeptiert, dann existiert ein einfacher nicht-deterministischer Akzeptor \mathcal{A}_B , für den gilt: $U(\mathcal{A}_A) = U(\mathcal{A}_B)$.

Ebenfalls bedeutungsvoll ist folgende Eigenschaft des Akzeptors /53/:

Satz 2.2.2: Ein Akzeptor ist (implizit) vollständig definiert. Für jeden nicht explizit angegebenen Übergang wird ein Übergang in einen stabilen *Rückweisungszustand* vorausgesetzt. Es ist üblich, in Diagrammen und Tafeln diesen speziellen Zustand und die Übergänge in diesen nicht einzutragen /46/.

2.3 Verfahren zur Synthese von Akzeptoren aus regulären Ausdrücken

Im folgenden Kapitel soll ein Überblick über Verfahren zur Synthese abstrakter Akzeptoren gegeben werden, die von einer Problemformulierung in Form eines regulären Ausdruckes ausgehen. Aufgrund der erfassenden Aufbereitung des gegebenen regulären Ausdruckes am Anfang dieser Algorithmen können drei Gruppen von Verfahren unterschieden werden, wobei die Erfassung erfolgen kann

- 1.) durch Anwendung von Methoden zur *Dekomposition* auf der Grundlage der KLEENE'schen Algebra
- 2.) durch Aufstellung einer Liste aller *Übergangspaare*
- 3.) durch Ermittlung der *charakteristischen Ableitungen*

Die genannten Begriffe werden im Zusammenhang mit den entsprechenden Syntheseverfahren im folgenden abschnittsweise behandelt.

2.3.1 Komposition von Zustandsgraphen

Eine Reihe bekannter Methoden zur Synthese von Akzeptoren aus regulären Ausdrücken beruht auf folgenden Kompositionsprinzipien /46/, die graphisch dargestellt seien /46,49/. Hierzu wird zunächst der in Bild 2.3.1 gezeigte *black box-Graph* eines allgemeinen deterministischen oder nicht-deterministisch einfachen Akzeptors \mathcal{N}_a definiert. Die beiden in Bild 2.3.1 sichtbaren Knoten z_0 (Anfangszustand) und z_f (Endzustand) seien die extremalen Knoten des Zustandsgraphen, während dessen übrige Knoten nebst zugehörigen Übergangs-Zweigen durch den rechteckigen Kasten repräsentiert seien. Durch ein in diesen Kasten eingetragenes Symbol α sei der reguläre Ausdruck für die durch den Akzeptor akzeptierte Menge $\alpha = U(\mathcal{N}_a)$ von Ereignissen bezeichnet.

- 1) Regel 2.3.1 (*disjunktive Komposition*). Wenn je ein einfacher Akzeptor \mathcal{N}_a und \mathcal{N}_b mit Zustandsgraphen gemäß

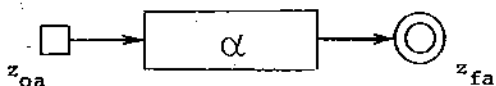


Bild 2.3.1 Black box - Zustandsgraph eines Akzeptors \mathcal{A}_a mit $U(\mathcal{A}_a) = a$.

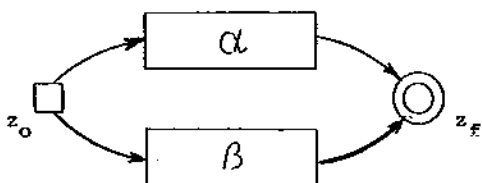


Bild 2.3.2 Komposition eines Akzeptors \mathcal{A} mit $U(\mathcal{A}) = \alpha \cup \beta$.

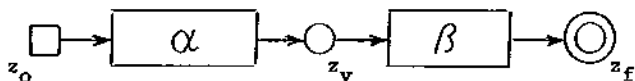


Bild 2.3.3 Komposition eines Akzeptors \mathcal{A} mit $U(\mathcal{A}) = \alpha\beta$.

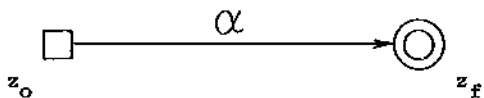


Bild 2.3.4 Triviale Lösung des Syntheseproblems nach Bild 2.3.1 für den atomaren Fall $\alpha = a$.

Bild 2.3.1 die durch α bzw. β bezeichneten regulären Ereignisse akzeptiert, dann wird die durch den regulären Ausdruck $\alpha \cup \beta$ bezeichnete reguläre Menge durch einen Akzeptor \mathcal{N} akzeptiert, dessen Graph durch Vereinigung der Graphen von \mathcal{N}_α und \mathcal{N}_β durch Verschmelzen der beiden Anfangszustände $z_{\alpha 0}$ und $z_{\beta 0}$ zu z_0 , sowie der beiden Endzustände $z_{\alpha f}$ und $z_{\beta f}$ zu z_f entsteht (Bild 2.3.2).

2) Regel 2.3.2 (multiplikative Komposition). Die durch den regulären Ausdruck $\alpha\beta$ bezeichneten Ereignisse werden durch einen Akzeptor \mathcal{N} akzeptiert, dessen Graph durch Vereinigung der Graphen von \mathcal{N}_α mit $\alpha = U(\mathcal{N}_\alpha)$ und \mathcal{N}_β mit $\beta = U(\mathcal{N}_\beta)$ dadurch entsteht, daß der Endzustand $z_{\alpha f}$ von \mathcal{N}_α mit dem Anfangszustand $z_{\beta 0}$ von \mathcal{N}_β zu z_v verschmolzen wird (Bild 2.3.3).

Wenn nun beispielsweise ein zusammengesetzter regulärer Ausdruck α als Formulierung eines abstrakten Syntheseproblems gegeben ist, so entspricht die Problemformulierung dem Graphen nach Bild 2.3.1. Es sind also zunächst nur die beiden extremalen Knoten $z_{\alpha 0}$ und $z_{\alpha f}$ bekannt. Wenn der reguläre Ausdruck α ohne Anwendung von Iterations-Operationen zusammengesetzt ist, so kann das Syntheseproblem auf einfache Weise durch schrittweise Dekomposition unter inverser Anwendung der obigen Regeln 2.3.1 und 2.3.2 gelöst werden. Hierbei erhöht sich die Zahl der dargestellten Knoten sukzessive. Die Zerlegungsprozedur wird solange fortgesetzt, bis nur noch solche Teil-Akzeptoren vorliegen, deren zugehöriger regulärer Ausdruck atomar ist. Da für den atomaren regulären Ausdruck die Lösung des Syntheseproblems trivial ist (Bild 2.3.4), liegt somit die Lösung des Gesamtproblems vor.

Eine einfachere Darstellungsweise des "black box"-Zustandsgraphen besteht darin, daß eine Wiedergabe gemäß Bild 2.3.4 auch dann zugelassen wird, wenn der reguläre Ausdruck α nicht atomar ist. Hierbei handelt es sich um eine gegen-

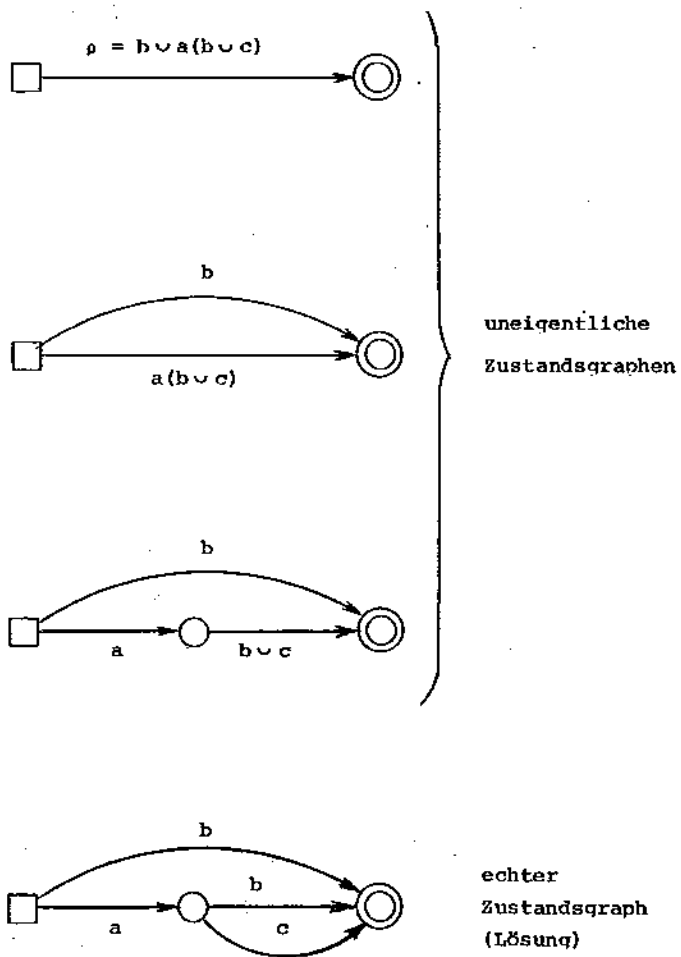


Bild 2.3.5 Beispiel für die abstrakte Synthese aus einem regulären Ausdruck $p = b \vee a(b \vee c)$.

über dem Graphen in Bild 2.2.7 noch weitergehende Verallgemeinerung des Folgezustands-Begriffes, indem ein Zweig des Graphen mehrere Pfade gleichzeitig und somit ein Ereignis aus mehreren Folgen beschreibt. Ein derart zusammengesetzter Graph ist, wie bereits erwähnt, ein "uneigentlicher" Zustandsgraph, wobei "atomare" und "nicht-atomare" Zweige unterschieden werden. Bild 2.3.5 zeigt in dieser Darstellungsweise die einzelnen Schritte für ein einfaches Synthesebeispiel für den regulären Ausdruck $\rho = b \cup a(b \cup c)$. (Zur Vereinfachung sei auf Zustandszuordnung der Knoten verzichtet.)

Bisher wurde die abstrakte Synthese aus regulären Ausdrücken unter Ausschluß der Iteration behandelt, da diese zusätzliche Probleme bringt, auf die im Folgenden eingegangen werden soll.

Regel 2.3.3 a: Wenn ein Akzeptor \mathcal{N}_a gegeben ist, der die reguläre Menge a akzeptiert, so kann daraus ein Automat \mathcal{A} gewonnen werden, der die reguläre Menge a^* darstellt, indem der Anfangszustand z_0 mit dem Endzustand z_f zu Zustand z_v verschmolzen wird (Bild 2.3.6).

Der nach Regel 2.3.3 a erzeugte Automat \mathcal{A} ist kein einfacher Akzeptor, da kein getrennter Anfangs- und Endzustand mehr vorhanden ist. Die Anwendung dieser Regel zu Zwecken der Komposition oder Dekomposition kann deshalb zu falschen Ergebnissen führen, was jedoch durch eine Zusatzmaßnahme gemäß Regel 2.2.3 b verhindert werden kann. Zuvor sei jedoch ein Beispiel gezeigt für einen solchen Fehler durch Aufstellung eines Graphen für den folgenden regulären Ausdruck (Bild 2.3.8a):

$$\rho = a^*b \cup c$$

Der Fehler besteht darin, daß der ohne Anwendung von Regel 2.2.3 b gewonnene Graph nicht den vorgegebenen obigen Ausdruck, sondern den folgenden regulären Ausdruck beschreibt,

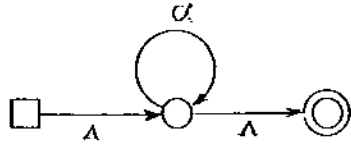
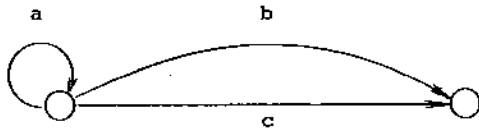
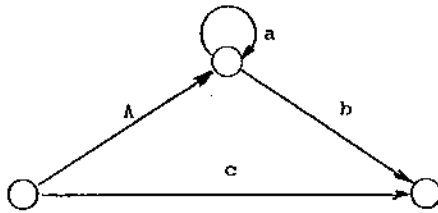


Bild 2.3.6 Darstellender Automat \mathcal{A} für $U(\mathcal{S}) = a^*$.

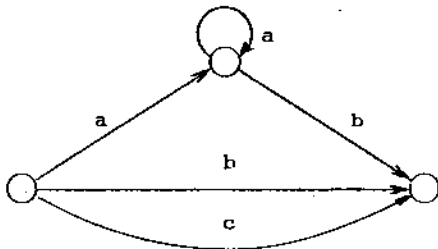
Bild 2.3.7 Akzeptor \mathcal{A} für $U(\mathcal{L}) = a^*$.



a).



b).



c).

Bild 2.3.8 Der Leerzweig als Hilfe zur Vermeidung von Fehlern bei Abbildung regulärer Ausdrücke in Graphen (Beispiel: $\rho = a^*b \cup c$; a) falsch, b) richtig, c) nach Entf. des Leerzweiges aus b)).

wie sich durch Verfolgung der einzelnen Pfade des Graphen leicht zeigen läßt:

$$\rho = a^*b \cup a^*c$$

Regel 2.3.3 b: (iterative Komposition). Ein nach Regel 2.3.3 a gewonnener Automat \mathcal{A} wird formal zum Akzeptor \mathcal{N} mit $U(\mathcal{N}) = a^*$, indem wiederum ein Anfangszustand z_0 und ein Endzustand z_f , sowie je ein *Leer-Übergang* (Leerzweig) von z_0 nach z_v und von z_v nach z_f hinzugefügt werden (entsprechend dem Graphen in Bild 2.3.7).

Bild 2.3.8 b zeigt den Graph zur richtigen Lösung des oben genannten Beispiels, die durch Anwendung von Regel 2.3.3 b gewonnen wurde.

Durch Regel 2.3.3 b eingeführte Leer-Übergänge sind nur als neutrales Element λ der KLEENE'schen Algebra, jedoch nicht als Teil der Spezifikation eines Automaten interpretierbar. Eine Synthese-Prozedur kann deshalb erst dann als beendet angesehen werden, wenn evtl. auftretende Leer-Übergänge wieder eliminiert sind (Beispiel: Bild 2.3.8 c aus Bild 2.3.8 b), was in bekannten Syntheseverfahren auf der Grundlage regulärer Ausdrücke stets durch einen Tabellen-Verarbeitungs-Algorithmus erfolgt.

2.3.2 Synthese-Algorithmus nach OTT/FEINSTEIN

Auf den beschriebenen Dekompositionsprinzipien beruht das Verfahren nach OTT und FEINSTEIN /46/ zur Synthese von Akzeptoren aus regulären Ausdrücken. Der erste Schritt des Algorithmus geht von einem "black box"-Graphen aus (vgl. Bild 2.3.1) und besteht in einer stufenweisen Dekomposition dieses Graphen, wobei der als Aufgabenstellung gegebene reguläre Ausdruck von links nach rechts abgearbeitet wird. Das Ergebnis dieses ersten Schrittes, dessen Verlauf durch ein Beispiel in Bild 2.3.9 veranschaulicht wird, ist im allgemeinen ein uneigentlicher Zustandsgraph, der Leerzweige enthält. Der zweite Schritt ist die Entfernung von Leerzweigen nach folgender Vorschrift:

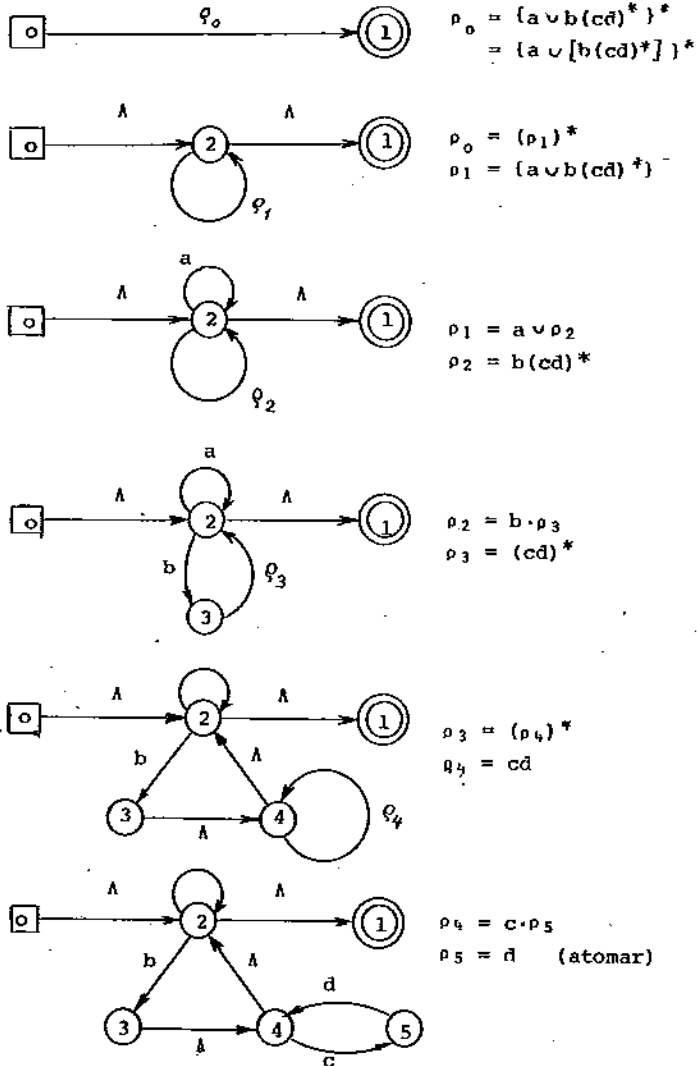


Bild 2.3.9 Beispiel für die Durchführung des ersten Teiles des Synthese-Verfahrens nach OPT/FEINSTEIN.

Regel 2.3.4

Alle diejenigen Knoten, die einen Leerzweig als einzigen abgehenden (ankommenden) Zweig haben, werden mit ihrem Folgezustands-Knoten (Vorzustands-Knoten) verschmolzen.

Bild 2.3.10 zeigt die Anwendung dieser Vorschrift auf das Beispiel nach Bild 2.3.9 mit folgenden Verschmelzungsvorgängen:

$$\{z_2, z_0\} \Rightarrow z_0, \{z_1, z_0\} \Rightarrow z_0, \text{ und } \{z_3, z_4\} \Rightarrow z_4.$$

Damit sind bei diesem speziellen Beispiel Anfangs- und Endknoten identisch. Für den noch übrigbleibenden Leerzweig von z_4 nach z_0 ist die obige Vorschrift nicht definiert. Leerzweige dieses Typs werden in nicht-graphischer Form in einem dritten Schritt des Algorithmus eliminiert, nachdem der Graph in eine nichtdeterministische Übergangstabelle übersetzt worden ist. Dieser numerische Teil der Methode ÖTF/FEINSTEIN ist in Stichworten wie folgt formuliert.

- a) Aufstellung der Übergangstabelle (δ -Tabelle) aus dem Graphen, wobei für jeden Knoten eine Zeile angelegt wird,
- b) Für jeden Knoten i , von dem ein Leerzweig ausgeht (nach Knoten j), wird $\{z_i, z_j\}$ als Folgezustand in die δ -Tabelle eingetragen (Beispiel: Bild 2.3.11 a),
- c) Für alle unterschiedlichen $\{z_i, z_j\}$ der δ -Tabelle werden neue Zustände $z_d = \{z_i, z_j\}$ definiert durch Einführung neuer Zeilen der δ -Tabelle,
- d) Jede Folgezustandseintragung der Form $\{z_i, z_j\}$ wird durch das ihr in Schritt c zugeordnete z_d ersetzt (z.B. Bild 2.3.11 b),
- e) Für Zeilen der Form $z_d = \{z_i, z_j\}$ wird die δ -Funktion wie folgt bestimmt: Zeile d ist die Vereinigung der Zeilen i und j der δ -Tabelle (Beispiel Bild 2.3.11 b);

Bild 2.3.11 b zeigt die Lösung des Syntheseproblems nach Bild 2.3.9 als Übergangstabelle und Bild 2.3.11 c als Zustandsgraph. Wie auch an diesem Beispiel zu sehen ist, er-

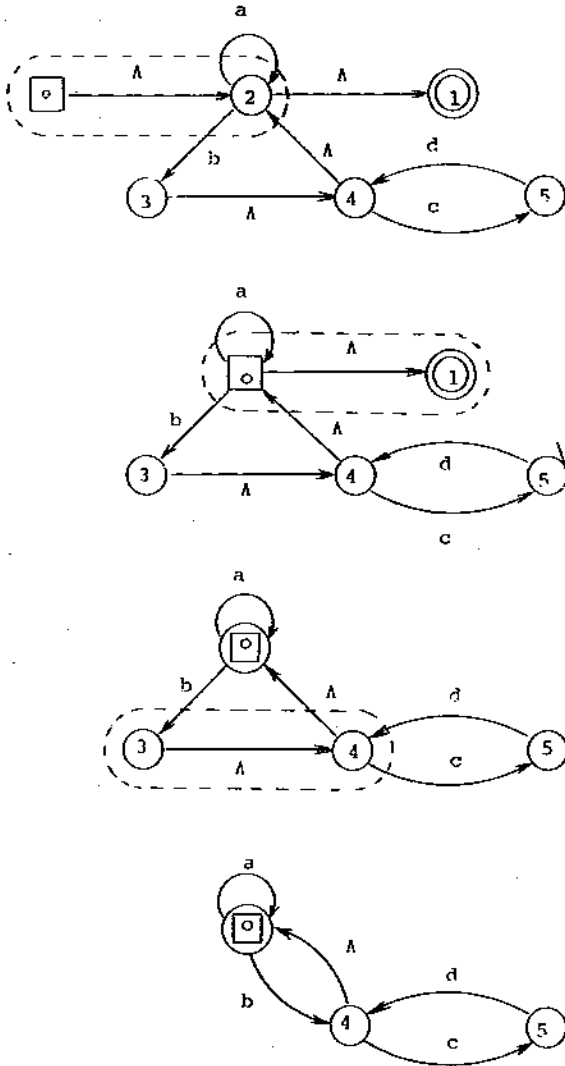


Bild 2.3.10 Graphische Entfernung von Leerzweigen nach der Methode von OTT und PFINSPEIN.

gibt die Methode nach OTT und FEINSTEIN nicht notwendigerweise einen Minimal-Automaten als Lösung (z_4 und z_6 sind äquivalente Zustände). Das Ergebnis ist dennoch eine zulässige Lösung, da bei deterministischen Automaten das Syntheseproblem - im Gegensatz zum Analyseproblem - nicht eindeutig ist. Die Anzahl der Zustände bei der gefundenen Lösung wird bei dieser Methode im allgemeinen größer als die Anzahl der Zustände des dazu äquivalenten Minimal-Automaten sein, wobei die Anzahl der überflüssigen Zustände vom Typ und von der Form des vorgegebenen regulären Ausdruckes abhängig ist. Zur Gewinnung einer Minimallösung ist die zusätzliche Anwendung eines Reduktions-Algorithmus notwendig.

2.3.3 Relationen zwischen Folgen und regulären Ausdrücken

Bei einigen Verfahren zur Synthese abstrakter Automaten aus regulären Ausdrücken müssen Ereignisse und darin enthaltene Folgen und Teilfolgen zueinander in Beziehung gebracht werden. In diesem Zusammenhang ist es zweckmäßig, vor der Beschreibung solcher Synthese-Verfahren, die nachstehenden Begriffsdefinitionen einzuführen /33,52,55/. Mit Hilfe dieser Definitionen werden später Methoden zur Ermittlung von Übergangspaaren und charakteristischen Gleichungen aus regulären Ausdrücken beschrieben.

Definition 2.3.1: Jedes reguläre Ereignis E wird demjenigen regulären Ausdruck $\rho(E)$ gleichgesetzt, welcher Deskriptor des regulären Ereignisses E ist. Für Folgen s , die in einem regulären Ereignis E enthalten sind, ergibt sich dabei:

$$s \in E \quad + \quad s \in \rho(E)$$

Diese durch Gleichsetzung von regulären Ereignissen mit regulären Ausdrücken gewonnene Definition gestattet eine vereinfachte Schreibweise.

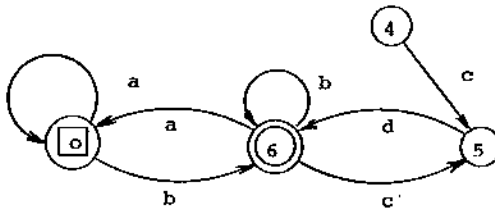
Definition 2.3.2: Eine *Silbe* einer Folge s ist eine Teilfolge von s , gewonnen durch Wegschneiden eines oder mehre-

$z \backslash x$	a	b	c	d
z_0	z_0	$\{z_4, z_0\}$		
z_4			z_5	
z_5				$\{z_4, z_0\}$

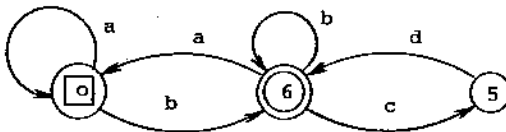
a)

$z \backslash x$	a	b	c	d
z_0	z_0	z_6		
z_4			z_5	
z_5				z_6
$\{z_0, z_4\} = z_6$	z_0	z_6	z_5	

b)



c)



d)

Bild 2.3.11 Letzter Teil und Ergebnis des Synthese-Verfahrens nach OTF/FEINSTEIN am Beispiel nach Bild 2.3.9/10; a) und b) letzter Schritt, c) Zustandsgraph des Ergebnisses, d) Minimallösung (zum Vergleich).

rer Symbole am Anfang und/oder Ende der Folge s .

Beispiel:

$\langle b, c, d \rangle$ ist eine Silbe der Folge $\langle a, b, c, d, e, f \rangle$.

Definition 2.3.3: Eine *Vorsilbe* einer Folge s ist eine Silbe von s , gewonnen durch Wegschneiden eines oder mehrerer Symbole am Ende von s .

Beispiel:

$\langle a, b, c \rangle$ ist eine Vorsilbe der Folge $\langle a, b, c, d, e, f \rangle$.

Definition 2.3.4: Eine *Nachsilbe* einer Folge s ist eine Silbe von s , gewonnen durch Wegschneiden eines oder mehrerer Symbole am Anfang der Folge s .

Beispiel:

$\langle e, f \rangle$ und $\langle f \rangle$ sind Nachsilben von $\langle a, b, c, d, e, f \rangle$.

Definition 2.3.5: Eine *Zwischensilbe* einer Folge s ist eine Silbe von s , die weder Vorsilbe noch Nachsilbe von s ist.

Definition 2.3.6: Ein *Anfangssymbol* (*Endsymbol*) einer Folge s ist eine Vorsilbe (Nachsilbe) von s mit der Länge $l = 1$.

Definition 2.3.7: Ein *Übergangspaar* einer Folge s ist eine Vor- Zwischen- oder Nachsilbe der Länge $l = 2$ aus der Folge s .

Beispiel: Die Folge $\langle a, b, c, d, e, f \rangle$ hat folgende Übergangspaare: $\langle a, b \rangle$, $\langle b, c \rangle$, $\langle c, d \rangle$, $\langle d, e \rangle$, $\langle e, f \rangle$.

Die Definitionen 2.3.2 bis 2.3.7 gelten nicht nur für Folgen über dem Alphabet X , sondern durch Verallgemeinerung auch in Bezug auf Ereignisse über dem Alphabet X und somit gemäß Definition 2.3.1 auch bezüglich regulärer Ausdrücke über dem Alphabet X . So ist beispielsweise die Folge t eine

Vorsilbe des regulären Ausdruckes ρ , genau dann, wenn die Folge t die Vorsilbe einer Folge $s \in \rho$ ist. Ebenfalls auf reguläre Ausdrücke über dem Alphabet X bezieht sich die folgende Definition der Δ -Funktion, die reguläre Ausdrücke in zwei verschiedene Kategorien einteilt /52/, wobei die Existenz der Leerfolge im Ereignis das Kriterium ist.

Definition 2.3.8: Die Δ -Funktion $\Delta(\rho)$ eines regulären Ausdruckes ρ ist definiert durch:

$$\Lambda \in \rho \rightarrow \Delta(\rho) = \Lambda \quad \text{und:} \quad \Lambda \notin \rho \rightarrow \Delta(\rho) = \emptyset.$$

Beispiele:

$$\begin{aligned} \Delta(\Lambda) &= \Lambda, & \Delta(\emptyset) &= \emptyset, \\ \Delta(\rho^*) &= \Lambda & & \text{(gemäß Definition 2.1.3),} \\ \Delta(x_1) &= \emptyset & & \text{für } x_1 \in X. \end{aligned}$$

Bei mehreren bekannten Syntheseverfahren auf der Grundlage regulärer Ausdrücke besteht der erste Schritt im Prinzip in einer Ermittlung sämtlicher Übergangspaare des als Problemformulierung vorgegebenen regulären Ausdruckes. Hierzu werden zunächst sämtliche Stellen von Symbolen $x_1 \in X$ innerhalb des regulären Ausdruckes durch Indizierung gekennzeichnet, wobei verschiedene Indizierungs-Schemata bekannt sind. Aus dem so vorbereiteten regulären Ausdruck können bei einiger Übung mühelos sämtliche Übergangspaare durch Inspektion herausgefunden werden /22/.

Beispiel: Gegeben ist der reguläre Ausdruck:

$\rho = a b c (d \cup a b)$. Mit Stellen-Indices lautet der Ausdruck: $\rho = a_1 b_2 c_3 (d_4 \cup a_5 b_6)$. Die Übergangspaare des regulären Ausdruckes ρ sind:

$$\langle a_1, b_2 \rangle, \langle b_2, c_3 \rangle, \langle c_3, d_4 \rangle, \langle c_3, a_5 \rangle, \langle a_5, b_6 \rangle.$$

Im Prinzip beruht die Ermittlung der Übergangspaare eines regulären Ausdruckes darauf, daß in rekursiver Weise aus

allen vorkommenden Kettenprodukten $\alpha\beta$ alle Paare aus Endsymbolen von α und Anfangssymbolen von β ermittelt werden /33/. Ein Sonderfall liegt hierbei vor, wenn die Δ -Funktion des Operanden β den Wert $\Delta(\beta) = \Lambda$ hat. In diesem Fall wird der Unter-Ausdruck β als mittlerer Operand innerhalb eines aus drei Operanden bestehenden Kettenproduktes $(\alpha\beta\gamma)$ betrachtet gemäß folgendem Schema.

Gesucht werden sämtliche Übergangspaare des (Unter-)Ausdruckes $\rho = \alpha \cdot \beta^* \cdot \gamma$. Hierbei ist $\Delta(\beta^*) = \Lambda$, $\Delta(\alpha) = \Phi$, und $\Delta(\gamma) = \Phi$. Übergangspaare dieses Ausdruckes ρ sind:

- 1) alle Paare aus Endsymbolen von α und Anfangssymbolen von β^* ,
- 2) alle Paare aus Endsymbolen von α und Anfangssymbolen von γ ,
- 3) alle Paare aus Endsymbolen von β^* und Anfangssymbolen von γ .
- 4) alle Übergangspaare innerhalb von α , β und γ , falls diese nicht atomar sind.

Der Prozeß der Paar-Ermittlung ist beendet, sobald sämtliche im vorliegenden regulären Ausdruck enthaltenen Kettenmultiplikationsoperatoren erfaßt sind.

2.3.4 Synthese-Algorithmus nach McNAUGHTON/YAMADA

McNAUGHTON und YAMADA /33/ geben ein Verfahren an zur abstrakten Synthese aus regulären Ausdrücken, das von der Erstellung einer Liste der Übergangspaare ausgeht. Eine Besonderheit besteht darin, daß die beiden Autoren von einem auf die Symbole 0 und 1 beschränkten Eingabealphabet $X = \{0, 1\}$ ausgehen. Das angegebene Verfahren ist jedoch auch für andere Alphabetlängen anwendbar. Der erste Schritt des Verfahrens besteht in der bereits erwähnten Indizierung des regulären Ausdruckes und der Aufstellung einer Liste der Übergangspaare. Zusätzlich wird eine Liste der Anfangssymbole und der Endsymbole des regulären Ausdruckes aufgestellt. Die einzelnen Anfangssymbole dieser Liste werden durch Voranstellen des Symbolen z_0 zu Paaren ergänzt und die erste Zeile der Liste der Übergangspaare eingetragen. Tafel 2.3.1

zeigt als Beispiel die entsprechende Liste der Übergangspaare für den regulären Ausdruck $\rho = 1(00 \cup 01)^* 0$.

Das Verfahren wird auf graphischem Wege fortgesetzt. Zunächst wird ein Knoten z_0 für den Anfangszustand gezeichnet. Alle weiteren elementaren Schritte im Aufbau des gesuchten Zustandsgraphen bestehen nun darin, daß jeweils zu einem vorhandenen Knoten ein Übergangszweig und der dazugehörige Folgezustands-Knoten für jedes Symbol des Eingabealphabetes eingetragen wird. Hierbei wird formal jeder Knoten einer oder mehreren Stellen des regulären Ausdruckes zugeordnet nach untenstehender Vorschrift, wie beispielsweise in $z_k \hat{=} M(z_k) = \{a_1, a_u, a_v, a_w\}$ mit den Folgerungen $a_1 \in M(z_k)$, $a_u \in M(z_k)$, $a_v \in M(z_k)$, und $a_w \in M(z_k)$. Die Konstruktionsvorschrift für einen Zustandsgraphen des gesuchten Automaten beruht unter dieser Voraussetzung auf folgendem Satz:

Wenn $\langle a_i, b_j \rangle$ ein Übergangspaar mit $a_i \in M(z_v)$ ist,
dann folgt daraus $b_j \in M(z_f)$ mit $z_f = \delta(z_v, b)$.

Für einen vorliegenden Knoten z_v ergibt sich folgende Vorschrift für die Ermittlung seiner unmittelbaren Folgezustands-Knoten und der entsprechenden Übergangszweige:

- 1) Man erfasse alle Übergangspaare, deren linkes Symbol in z_v enthalten ist, durch Auswählen der entsprechenden Zeilen in der Liste der Übergangspaare.
- 2) Aus dieser Menge fasse man nach der Zugehörigkeit zum Eingabealphabet alle rechten Symbole dieser Paare zu getrennten Teilmengen zusammen und ordne diesen Teilmengen jeweils einen Folgezustands-Knoten zu. Man trage die derart definierten Knoten in den Graphen ein.
- 3) Man trage zu jedem dieser neuen Knoten einen Übergang von z_v ausgehend ein, welcher der Alphabet-Zugehörigkeit der Symbolmenge des neuen Knotens entspricht.

4) Alle diejenigen Knoten, in denen mindestens ein Symbol aus der Liste der Endsymbole enthalten ist, sind Knoten von Endzuständen. Diese werden durch einen Doppelkreis gekennzeichnet.

Bild 2.3.12 a zeigt den Zustandsgraphen des über Tafel 2.3.1 ermittelten Akzeptors, der in den folgenden Schritten aufgestellt wurde:

$$\delta(z_0, 1) = z_1 \hat{=} \{1_1\},$$

$$\delta(z_1, 0) = z_2 \hat{=} \{0_1, 0_4, 0_6\},$$

$$\delta(z_2, 0) = z_3 \hat{=} \{0_3\}, \quad \delta(z_2, 1) = z_4 \hat{=} \{1_5\},$$

$$\delta(z_3, 0) = z_2,$$

$$\delta(z_4, 1) = z_2$$

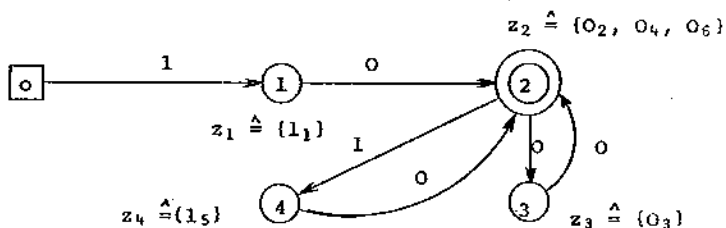
Der an die Ermittlung der Übergangspaare anschließende Teil der Syntheseprozedur kann auf einfache Weise modifiziert werden, indem die mittels obiger Regeln 1 - 4 ermittelten Übergänge und Folgezustände nicht graphisch, sondern in einer Übergangstabelle erfaßt werden. Tafel 2.3.2 zeigt die entsprechende Übergangstabelle für das oben verwendete Beispiel.

Das gewählte Beispiel stellt einen für die beschriebene Methode günstigen Fall dar mit nur 5 Zuständen, gegenüber theoretisch möglichen 32 Zuständen. Das theoretische Maximum ist 2^p , wenn p die Zahl der Stellen des regulären Ausdrucks ist /33/. Der Minimal-Akzeptor für den gegebenen regulären Ausdruck hat jedoch nur 3 Zustände (Bild 2.3.12 b). Diese Minimallösung ist bei dem vorliegenden Beispiel durch Anwendung der Methode nach McNAUGHTON und YAMADA nicht erreichbar. Der Grad an Approximation der Minimallösung ist bei dieser Methode sehr unterschiedlich und stark von der zufällig vorliegenden Form des vorgegebenen regulären Ausdrucks abhängig.

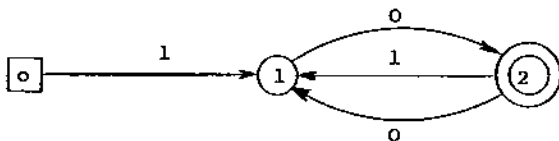
Anfangssymbol der Paare	Übergangspaare
z_0	$\langle z_0, 1_1 \rangle$
1_1	$\langle 1_1, 0_2 \rangle, \langle 1_1, 0_4 \rangle, \langle 1_1, 0_6 \rangle$
0_2	$\langle 0_2, 0_3 \rangle$
0_4	$\langle 0_4, 1_5 \rangle$
0_3	$\langle 0_3, 0_2 \rangle, \langle 0_3, 0_4 \rangle, \langle 0_3, 0_6 \rangle$
1_5	$\langle 1_5, 0_2 \rangle, \langle 1_5, 0_4 \rangle, \langle 1_5, 0_6 \rangle$

Anfangssymbole: 1_1	Endsymbole: 0_4
-----------------------	-------------------

Tafel 2.3.1 Liste der Übergangspaare, Anfangssymbole und Endsymbole des regulären Ausdrucks
 $\rho = 1(00 \cup 01)^* 0 = 1_1(0_2 0_3 \cup 0_4 1_5)^* 0_6$



a). Lösung nach der Methode McNAUGHTON / YAMADA.



b). Minimallösung.

Bild 2.3.12 Lösungen des Syntheseprobles $\rho = 1(00 \cup 01)^* 0$.

2.3.5 Synthese-Algorithmus nach GLUSCHKOW

Ein weiteres Verfahren zur abstrakten Synthese aus regulären Ausdrücken auf der Grundlage der Übergangspaare wird von GLUSCHKOW angegeben /22/. Die Lösungen bei diesem Verfahren sind bei gleicher Form der Aufgabenstellung identisch mit Lösungen, die mit Hilfe des Verfahrens nach McNAUGHTON/YAMADA ermittelt werden. Dies bedeutet, daß bei einem gegebenen Problem beide Verfahren Automaten mit gleichvielen Zuständen und gleichguter (oder -schlechter) Approximation des Minimalautomaten liefern. Der Algorithmus von GLUSCHKOW unterscheidet sich von dem nach McNAUGHTON/YAMADA durch seine wesentlich kompliziertere Indizierungsvorschrift.

Eine Modifikation des Syntheseverfahrens nach GLUSCHKOW wurde von KUSNEZOW angegeben /21/. Die Modifikation besteht darin, daß der als Problemstellung vorgegebene reguläre Ausdruck nach der Dekompositions-Methode in Form eines gerichteten Graphen dargestellt wird, und anschließend die Indizierung an diesem Graphen vorgenommen wird.

2.3.6 Synthese-Algorithmus nach BRZOZOWSKI

Der erste Schritt eines Syntheseverfahrens, daß sich von den bisher beschriebenen sehr stark unterscheidet, beruht auf der Ermittlung der *charakteristischen Ableitungen* des gegebenen regulären Ausdruckes. Im folgenden soll zunächst der Begriff der Ableitung für reguläre Ausdrücke definiert werden und ein Verfahren zur Ermittlung dieser Ableitungen beschrieben werden /52/.

Definition 2.3.9: Die Ableitung $D_{s,p}$ eines regulären Ausdruckes p bezüglich einer Folge s ist die Menge aller Nachsilben t des Ausdruckes p , deren Kettenprodukte st in p

enthalten sind, d.h.

$$D_s \rho = \{t \mid st \in \rho\}$$

Beispiel: Gegeben sei ein regulärer Ausdruck

$$\rho = \{\langle a, b, c, a \rangle, \langle a, b, d \rangle, \langle a, b \rangle, \langle b, e, f \rangle\}$$

Die Ableitung von ρ bezüglich der Folge $s = \langle a, b \rangle$ ist

$$D_s \rho = \{\langle c, a \rangle, \langle d \rangle, \Lambda\}$$

Die Ableitung von ρ bezüglich der Folge $s = \langle b \rangle$ ist

$$D_s \rho = \{\langle e, f \rangle\}$$

Die Ableitung eines regulären Ausdruckes ist wiederum ein regulärer Ausdruck. Die Ableitung eines regulären Ausdruckes bezüglich einer Folge $s = \langle a, b, c, \dots \rangle$ kann rekursiv ermittelt werden durch sukzessive Ermittlung der Ableitungen von Ableitungen bezüglich einzelner Symbole des Alphabetes X nach dem Schema $D_{x_1} \alpha \rho = D_a (D_{x_1} \rho)$.

In der Menge sämtlicher Ableitungen eines regulären Ausdruckes befinden sich in der Regel auch äquivalente Ausdrücke, wobei die Äquivalenz wegen unterschiedlicher Form der Ausdrücke oft nicht sofort erkennbar ist. Solche äquivalenten Ausdrücke werden als Ausdrücke vom gleichen Typ bezeichnet, wobei folgender Satz gilt /52/.

Satz 2.3.1: Jeder reguläre Ausdruck ρ hat eine endliche Anzahl d_ρ von Typen von Ableitungen. Die Menge aller Ableitungen verschiedenen Typs von ρ heißt *Satz charakteristischer Ableitungen* von ρ .

Zwei weitere wichtige Sätze für die Synthese mit Hilfe charakteristischer Ableitungen sagen folgendes aus /52/.

z \ x	0	1
z ₀		{1 ₁ }
z ₁ = {1 ₁ }	{0 ₂ , 0 ₄ , 0 ₆ }	
z ₂ = {0 ₂ , 0 ₄ , 0 ₆ }	{0 ₃ }	{1 ₅ }
z ₃ = {0 ₃ }	{0 ₂ , 0 ₄ , 0 ₆ }	
z ₄ = {1 ₅ }	{0 ₂ , 0 ₄ , 0 ₆ }	

z \ x	0	1
z ₀	-	z ₁
z ₁	z ₂	-
z ₂	z ₃	z ₄
z ₃	z ₂	-
z ₄	z ₂	-

a) ursprüngliche Übergangstabelle

b) umbenannte Übergangstabelle

Tafel 2.3.2: Beispiel für Synthese nach McNAUGHTON/YAMADA aus dem regulären Ausdruck $\rho = 1(00 \cup 01)^*0$.

Problem : $\rho = 1(00 \cup 01)^*0$ Alphabet : $X = \{0, 1\}$		
Ableitung $D_{s\rho}$	zugeord. Zustand	Bemerkung
$D_{\Lambda\rho} = 1(00 \cup 01)^*0$	z_0	Anfangszust., da $s = \Lambda$
$D_{0\rho} = \emptyset$		
$D_{1\rho} = (00 \cup 01)^*0$	z_1	Endzust., da $\Lambda \in D_{10}$ für $l(s) = 3$ keine Ableitungen neuen Typs
$D_{00\rho} = \emptyset$		
$D_{01\rho} = \emptyset$		
$D_{10\rho} = (0 \cup 1)(00 \cup 01)^*0 \cup \Lambda$	z_2	
$D_{11\rho} = \emptyset$		
$D_{000\rho} = \emptyset$		
$D_{001\rho} = \emptyset$		
$D_{010\rho} = \emptyset$		
$D_{011\rho} = \emptyset$		
$D_{100\rho} = (00 \cup 01)^*0 = D_{1\rho}$	(z_1)	
$D_{101\rho} = (00 \cup 01)^*0 = D_{1\rho}$	(z_1)	
$D_{110\rho} = \emptyset$		
$D_{111\rho} = \emptyset$		

Tafel 2.3.3: Beispiel für die Ermittlung der charakteristischen Ableitungen eines regulären Ausdruckes.

Satz 2.3.2: Die Anzahl der Zustände desjenigen Minimal-Automaten, der ein reguläres Ereignis E akzeptiert, das durch den regulären Ausdruck $\rho(E)$ beschrieben wird, ist gleich der Anzahl d_ρ der charakteristischen Ableitungen von $\rho(E)$.

Bei Vorliegen der Aufgabenstellung als regulärer Ausdruck ρ_0 werden nach diesem Verfahren die folgenden Schritte durchgeführt.

- 1) Ermittlung der Menge der charakteristischen Ableitungen von ρ_0 (Beispiel: Tafel 2.3.3).
- 2) Bildung der Zustandsmenge durch eindeutige Zuordnung der Ableitungen (vgl. Bild 2.3.13).
- 3) Ermittlung der Übergänge aus der Liste der Ableitungen.

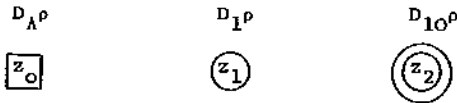
Das Verfahren nach BRZOWSKI hat den Nachteil, daß bei komplizierten regulären Ausdrücken die Erkennung gleicher Typzugehörigkeit bei Ableitungen oft schwierig ist. Daher ist trotz der prinzipiellen Möglichkeit hierzu bei manueller praktischer Anwendung nicht immer die Minimallösung eines vorgegebenen Syntheseprobemes erreichbar. Die gesicherte Erzielung von Minimallösungen ist bei komplizierteren Problemen mit diesem Verfahren nur bei geeigneter Anwendung einer Datenverarbeitungsanlage denkbar.

2.3.7 Vergleich der Synthese-Verfahren

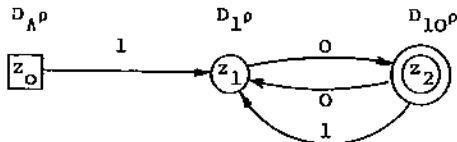
Tafel 2.3.4 gibt einen zusammenfassenden Überblick über die bekannten und vorstehend beschriebenen Verfahren zur abstrakten Synthese aus regulären Ausdrücken. Dabei zeigt sich, daß es nur ein einziges Verfahren gibt, bei dem die Erzielung der Minimallösung gesichert ist. Es handelt sich hierbei um das in Kapitel 2.3.6 beschriebene algebraische Verfahren nach BRZOWSKI /52/. Die Erzielung der Minimallösung ist bei diesem Verfahren ohne die Anwendung

Synthese-Verfahren nach	Art der Problem-Aufbereitung	weiterer Verlauf des Algorithmus	zusätzlicher Reduktions-Algorithmus erforderlich?
McNAUGHTON/ YAMADA (1960)	Ermittlung der Übergangspaare (Listenverarbgt.)	rekursiver Aufbau eines determinist. Zustandsgraphen	ja
OTT/ FEINSTEIN (1961)	graphische Dekomposition	Beseitigung von Leerübergängen (graphisch und listenverarb.)	ja
GLUSCHKOW (1961)	Erfassung der Übergangspaare (implizit durch Listenverarbgt.)	rekursiver Aufbau eines determinist. Zustandsdiagrammes	ja
KUSNEZOW (1963)	listenmäßige Erfassung der Übergangspaare mit graphischer Dekomposition (teils heuristisch)	rekursiver Aufbau eines determinist. Zustandsdiagrammes	ja
BRZOZOWSKI (1964)	algebraische Ermittlung der charakteristischen Ableitungen	rekursiver Aufbau eines determinist. Zustandsgraphen	bei Automatisierung: nein

Tafel 2.3.4: Übersicht über bekannte Verfahren zur abstrakten Synthese aus regulären Ausdrücken



a) Zustandszuordnung der charakteristischen Ableitungen von ρ .



b) Minimallösung

Bild 2.3.13 Synthesebeispiel nach BRZOZOWSKI (Tafel 2.3.3), Ergebnis des vorletzten (a) und des letzten (b) Schrittes

eines Digitalrechners aus Zeitaufwandsgründen sehr erschwert /52/, wie in Kapitel 2.3.6 gezeigt wird. In Kapitel 3 ff. wird eine neue Synthesemethode entwickelt, die auch bei komplexeren Problemen noch relativ übersichtlich bleibt und die Erzielung der Minimallösung auch dann bei relativ geringem Zeitaufwand ermöglicht, wenn kein Digitalrechner eingesetzt wird.

2.4 Ein Analyseverfahren für Akzeptoren auf der Grundlage regulärer Ausdrücke

MASON /37,38/ gibt eine Methode zur Lösung von linearen Gleichungen und Systemen linearer Gleichungen mit Hilfe linearer gerichteter Graphen an, die beispielsweise zur Berechnung der Übertragungseigenschaften linearer elektrischer Netzwerke geeignet ist. Das MASON'sche Verfahren besteht darin, daß eine zu lösende Gleichung oder ein zu berechnendes Netzwerk zunächst in Form eines *Signal-Fluß-Graphen* dargestellt wird. Die einzelnen Koeffizienten des Systemes werden durch gerichtete Zweige in diesem Graphen dargestellt. Zur Ermittlung des Übertragungsfaktors von einem Knoten i zu einem Knoten j werden bei diesem Verfahren mit Hilfe eines Satzes von Regeln sukzessive Knoten aus dem Signalflußgraphen entfernt solange, bis nur noch die Knoten i und j übrig sind. Der Koeffizient des einzigen dann noch übrigen und diese beiden Knoten verbindenden Zweiges ist die Lösung des Problems. BRZOZOWSKI und McCLUSKEY /34/ geben entsprechende Methoden zur Ermittlung regulärer Ausdrücke aus Zustandsgraphen an, wobei der Zustandsgraph als modifizierter Signalflußgraph interpretiert wird. Die Modifikation ist erforderlich wegen der in den Axiomen begründeten Unterschiede zwischen der gewöhnlichen linearen Algebra und der KLEENE'schen Algebra der regulären Mengen.

Zur Einführung seien kurz einige Elemente der Darstellung von Signalflußgraphen angegeben /37/. Ein Signalflußgraph ist ein Netz gerichteter Zweige, die Knoten verbinden. Zweig ij entspringt Knoten i und endet an Knoten j . Ein Quellenknoten ist nur mit abgehenden Zweigen verbunden, während ein Senkenknoten nur mit ankommenden Zweigen verbunden ist. Mit jedem Zweig ij wird ein Koeffizient t_{ij} verknüpft (graphisch: Beschriftung am Pfeil), der als Transmission t_{ij} von Knoten i nach Knoten j bezeichnet wird. Mit jedem Knoten i wird ein Signal u_i verknüpft. Die Abbildung

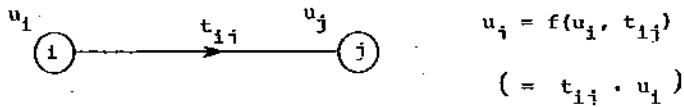


Bild 2.4.1 Elementare Äquivalenz zwischen Signalflußgraph und linearer algebraischer Gleichung.

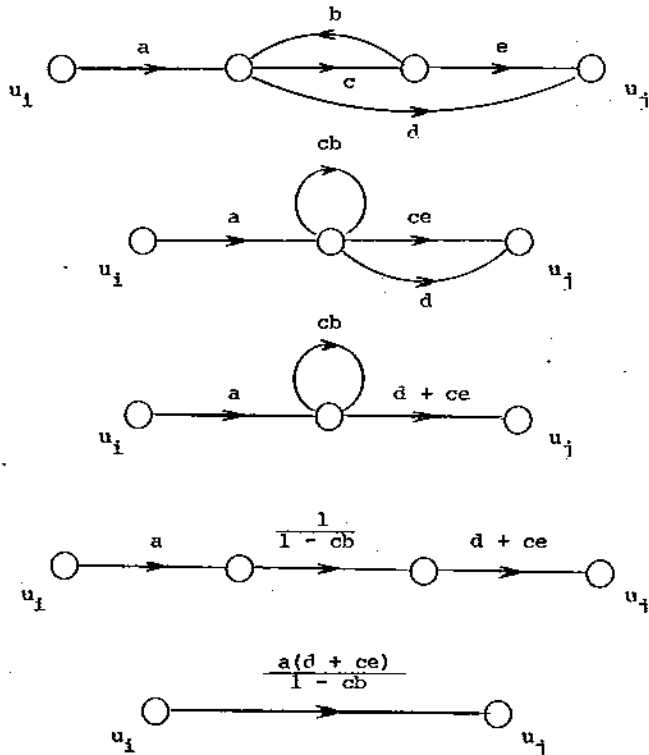


Bild 2.4.2 Beispiel für die Ermittlung der Transmission t_{ij} nach der Signalflußgraphen-Methode.

des aus diesen Elementen dargestellten elementaren Signalflußgraphen (Bild 2.4.1), bestehend aus einem einzigen Zweig t_{ij} , in die lineare Algebra führt zur Bedeutung $u_j = f(u_i, t_{ij})$, wobei $f(u_i, t_{ij}) = t_{ij} \cdot u_i$ ist.

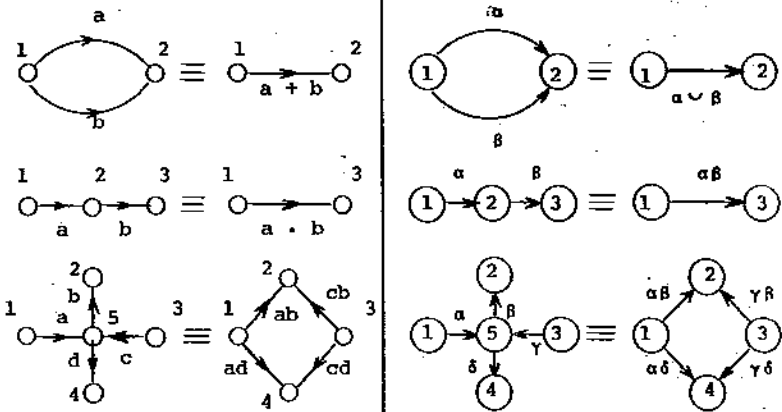
Unter der Voraussetzung, daß die Abbildung der vorgegebenen Beziehungen in einen Graphen richtig gelöst worden ist, zeigt Bild 2.4.2 ein Beispiel für die Ermittlung der Transmission t_{ij} zwischen Knoten i und Knoten j durch schrittweise Entfernung von Knoten eines Signalflußgraphen mit Hilfe der MASON'schen Äquivalenzen. Diese Methode besteht in der mehrfachen Anwendung der folgenden drei Elementaroperationen:

- a) Entfernung von Knoten gemäß den Äquivalenzen nach Bild 2.4.3 a (linke Seite).
- b) Reduktion von Schleifen auf Schlingen gemäß Äquivalenz nach Bild 2.4.3 b (linke Seite).
- c) Entfernung von Schlingen gemäß Äquivalenz nach Bild 2.4.3 c und d (linke Seite).

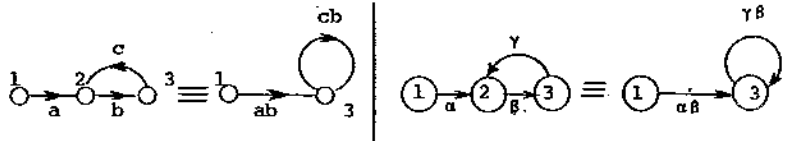
BRZOWSKI hat gezeigt, daß jedes deterministische Zustandsdiagramm eines Automaten als linearer Signalfluß-Graph interpretiert werden kann, und daß Techniken zur Manipulation linearer Signalflußgraphen auch zur Ermittlung regulärer Ausdrücke aus Zustandsgraphen verwendet werden können /34/. Schon rein äußerlich sind die Graphen-Darstellungen völlig einander entsprechend als Netz gerichteter Zweige, die Knotenpunkte miteinander verbinden. Die Beschriftung von Zweigen ist in einem Fall die "Transmission" zwischen den beiden verbundenen Knoten und im Falle des Zustandsgraphen das den entsprechenden Zustands-Übergang auslösende Ereignis. Dem Signal an einem Knoten entspricht der reguläre Ausdruck, dessen Folgezustand durch diesen Knoten dargestellt wird. Die Einheitstransmission (vom Wert 1) entspricht dem Λ , der Folge der Länge Null. Einer Transmission vom Wert null entspricht \emptyset , die Leermenge der Folgen.

Signalflußgraph

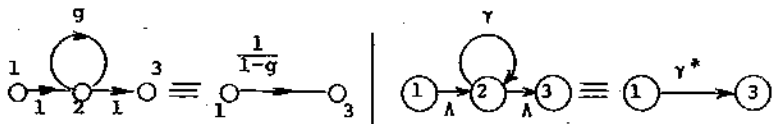
Zustandsgraph



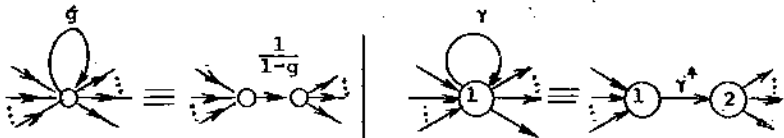
a). Elementare Äquivalenzen zur Zusammenfassung von: parallelen und seriellen Pfaden, sowie von Stern zu Masche.



b). Äquivalenz zur Reduktion von Schleifen zu Schlingen.



c). Äquivalenz zur Beseitigung von Schlingen.



d). Allgemeines Beispiel für die Beseitigung von Schlingen.

Bild 2.4.3 Elementare Äquivalenzen in Signalflußgraphen und Automaten-Graphen.

Es gibt weitere elementare Äquivalenzen, die sowohl auf lineare Signalflußgraphen als auch auf deterministische Zustandsgraphen angewandt werden können unter der Einschränkung, daß das Produkt nicht kommutativ ist. Bild 2.4.3 zeigt auf der rechten Seite eine Zusammenstellung solcher Äquivalenzen für Zustandsgraphen. Die Schlinge (Bild 2.4.3 c) entspricht bei konventionellen Signalfluß-Graphen der Division, wobei die Transmission $1/(1-g)$ ist. Für $|g| < 1$ ergibt sich die Analogie zur Iteration der regulären Algebra $g^* = 1 \cup g \cup g^2 \cup g^3 \cup g^4 \cup \dots$ mit

$$\frac{1}{1-g} = 1 + g + g^2 + g^3 + g^4 + \dots$$

Entsprechend der Methode nach MASON wird bei BRZOZOWSKI durch sukzessive Entfernung aller Knoten eines einfachen deterministischen Zustandsgraphen außer zwei Knoten i und j der reguläre Ausdruck ρ_{ij} des von Zustand z_i nach Zustand z_j überführenden Ereignisses ermittelt. Zu einer solchen Reduktion eines Graphen werden die in Bild 2.4.3 rechts dargestellten Äquivalenzen graphischer Unterstrukturen ausgenutzt. Zur Demonstration dieser Methode sei die Aufgabe gestellt, zu demjenigen Akzeptor den akzeptierten regulären Ausdruck zu ermitteln, dessen Graph in Bild 2.3.12 b gezeigt wird. Bild 2.4.4 zeigt hierzu die folgenden Schritte der Reduktion des Graphen:

- a) Aufgabenstellung
- b) ergibt sich aus Äquivalenz nach Bild 2.4.3 a (oben)
- c) ergibt sich aus Äquivalenz nach Bild 2.4.3 b
- d) ergibt sich aus Äquivalenz nach Bild 2.4.3 c
- e) ergibt sich aus Äquivalenz nach Bild 2.4.3 a (Mitte) und ist die Lösung mit dem regulären Ausdruck:

$$\rho = 10((0 \cup 1)0)^*$$

Nach den Regeln der KLEENE'schen Algebra folgt daraus:

$$\rho = 1(0(0 \cup 1))^* 0 = 1(00 \cup 01)^* 0$$

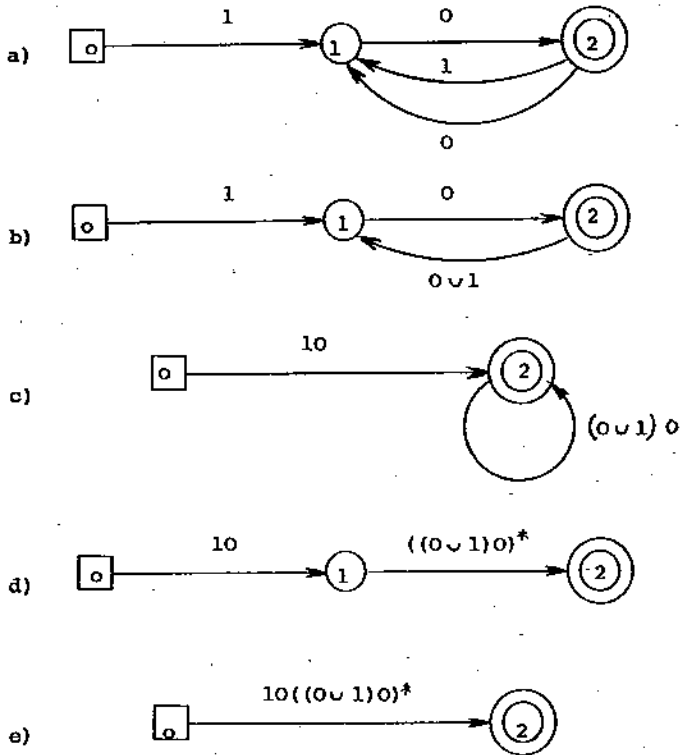


Bild 2.4.4 Beispiel für die Analyse eines Akzeptors nach BRZOZOWSKI/McCLUSKEY jr.

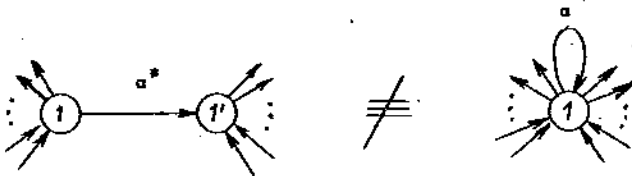


Bild 2.4.5 Zur Begründung der Notwendigkeit von Leerzweigen bei Synthese-Prozeduren

Letzterer Ausdruck hat die Form der Aufgabenstellung des Syntheseproblem, das in Kapitel 2.3 die Lösung nach Bild 2.3.12 a (bzw. b) hatte. Die Schlußfolgerung daraus ist folgende: der echten Mehrdeutigkeit des Syntheseproblem steht eine unechte Mehrdeutigkeit des Analyseproblem gegenüber. Die Mehrdeutigkeit bezieht sich beim Analyseproblem auf die Form des ermittelten regulären Ausdruckes, jedoch nicht auf dessen Typ. Da die verschiedenen möglichen Lösungen untereinander gleich sind, handelt es sich um keine echte Mehrdeutigkeit.

Ein weiterer Unterschied des Analyseproblem gegenüber dem Syntheseprobem besteht darin, daß die Einführung von Leerzweigen vermeidbar ist. Ein Knoten mit Schlinge läßt sich bei Transformation nach Bild 2.4.3 d immer so zerlegen, daß der entstehende Iterationszweig einen Knoten mit nur ankommenden Zweigen und einen Knoten mit nur abgehenden Zweigen verbindet. Beim Syntheseprobem ist bei der Umwandlung eines Iterationszweiges in eine Schlinge diese Struktur nicht die einzig mögliche Aufgabenstellung, wie Bild 2.4.5 zeigt. Daraus ergibt sich die Folgerung, daß die Analysemethode nach BRZOZOWSKI nicht ohne weiteres durch inverse Anwendung zur Lösung von Syntheseprobemen anwendbar ist. In Kapitel 3 werden Vorschläge zur Modifizierung dieser Methode für die Anwendung auf Syntheseprobeme beschrieben.