Über die Anwendung des Endlichen Akzeptors auf das Zeichenerkennungsproblem

Zur Erlangung des akademischen Grades
eines Doktor-Ingenieurs
von der Fakultät für Elektrotechnik
der Universität Karlsruhe
genehmigte
D i s s e r t a t i o n

vorgelegt
von
Dipl.-Ing. Reiner Hartenstein
aus
Berlin

Tag der mündlichen Prüfung: 19. Juni 1969

Hauptreferent: Prof. Dr.-Ing. K. Steinbuch
Korreferent: Prof. Dr. rer. nat. K. Nickel

Inhaltsverzeichnis

Zusammenfassung

Die Grundlagen der vorliegenden Arbeit sind die *Algebra der regulären Mengen* nach KLEENE und der aus der Automatentheorie bekannte *Akzeptor*. Der Akzeptor ist ein auch unter der Bezeichnung *Detektor* oder *Erkennender Automat* bekannter abstrakter Automat. Ausdrücke der Algebra der regulären Mengen werden *reguläre Ausdrücke* genannt.

Die vorliegende Arbeit behandelt

Die Synthese von Akzeptoren aus regulären Ausdrücken mit Hilfe gerichteter Graphen,

Die Simulation von Operationen an gerichteten Graphen auf Digitalrechnern,

Die Verallgemeinerung des Akzeptor-Modelles und

Methoden zur Lösung von Klassifikationsproblemen und Problemen der Informationsreduktion auf der Grundlage des abstrakten Akzeptors.

Nach einer Zusammenfassung der Grundlagen zur Algebra der regulären Mengen und zur Theorie des abstrakten Akzeptors wird eine Übersicht über bekannte Methoden zur abstrakten Automaten-Synthese aus regulären Ausdrücken gegeben. Hierbei wurde in der Darstellung eine Vereinheitlichung der in den Originalarbeiten sehr individuellen Terminologie angestrebt.

Es folgt die Einführung einer rein graphischen Methode zur Manipulation regulärer Ausdrücke, sowie echter und uneigentlicher Zustandsgraphen, welche die Erzielung der Minimallösung bei der abstrakten Synthese aus regulären Ausdrücken ermöglicht. Die Methode wird für den Fall der Akzeptor-Syn-

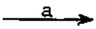
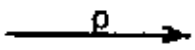
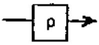
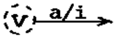
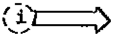
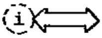
these auf der Theorie des abstrakten Akzeptors begründet. Darüber hinaus wird gezeigt, daß die eingeführte Methode auch auf solche speziellen Moore- und Mealy-Automaten angewandt werden kann, die zur Lösung von Zeichenerkennungs-Problemen geeignet sind. Zur Begründung des erweiterten Anwendungsbereiches wird die Definition des Akzeptor-Begriffes verallgemeinert.

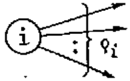
Für Zwecke der Simulation auf Digitalrechnern wird eine allgemeine Listenstruktur zur Darstellung gerichteter Graphen vorgeschlagen. Zusätzlich wird hierzu ein Satz elementarer Operationen zur computer-gerechten Transformation graphischer Unterstrukturen angegeben. Die eingeführte Simulationstechnik ermöglicht beispielsweise zur Minimisierung von Akzeptor-Strukturen gegenüber der Anwendung der konventionellen Automatentafel erhebliche Rechenzeitersparnisse. Die relative Ersparnis steigt dabei mit der Anzahl der Zustände und der Länge des Eingabealphabetes, wobei Unterschiede von mehreren Größenordnungen erreicht werden können.

Neben automatentheoretischen Problemen können durch die angegebene Simulationstechnik auch andere an gerichteten Graphen darstellbare Prozesse simuliert werden.

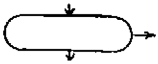
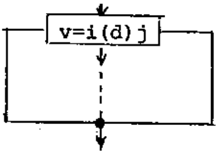
Verwendete Symbole und Abkürzungen

Graphen:

	Knoten für einen Zustand z_i ohne Ausgabe (beim Moore-Akzeptor: $\lambda(z_i) = 0$)
	Knoten für einen Anfangszustand z_0
	Knoten für einen Endzustand z_f gemäß Akzeptor-Definition (beim Moore-Akzeptor: $\lambda(z_f) > 0$)
	Endzustands-Knoten eines Moore-Akzeptors mit $\lambda(z_f) = i > 0$
	Knoten für einen Zustand, der gleichzeitig Anfangs- und Endzustand ist
	echter Übergangszweig (a ist atomar) ohne Ausgabe
	uneigentlicher Zweig (ρ ist nicht-atomar)
	uneigentlicher Zweig in black box-Graph-Darstellung
	End-Zweig eines Mealy-Akzeptors mit $\lambda(z_v, a) = i > 0$ (a ist atomar)
	Zweig in Signalflußgraphen
	abgehendes Bündel eines Knotens i
	ankommendes Bündel eines Knotens i
	gemischtes Bündel eines Knotens i
	nicht-unterscheidbare Bündel zweier Knoten i,j
	nicht-rückwärts-unterscheidbare Bündel zweier Knoten i,j

	Zusammenfassung von Zweigen zu einem Bündel

Strukturdiagramme:

	Anfang eines (Unter-) Programmes
	Ende eines (Unter-) Programmes
	Vereinigung von Programmzweigen
	Operation(en) innerhalb eines Programmes
	Programm-Verzweigung
	Programm-Zyklus mit v = i, i+d, i+2d,...j

Logik:

$\wedge$	Konjunktion *(und)*
$\vee$	Adjunktion *(oder)*
$\rightarrow$	Subjunktion *(folgt)*
$\leftrightarrow$	Bijunktion *(dann und nur dann folgt)*
$:$	Definition *(ist definiert über)*
$\exists_x$	Partikularisator *(es existiert ein x)*
$\forall_x$	Generalisator *(für alle x gilt)*

Mengen
Abbildungen:

$\in$	*ist Element von*	Mitgliedschafts-Relation
$\notin$	*ist nicht Element von*	
$\subset$	Inklusion *(ist Untermenge von)*	
$\subseteq$	Inklusion *(ist echte Untermenge von)*	
ϕ	leere Menge	

$\{.....\}$	Menge der Elemente ....
$\{ x \mid q \}$	Menge aller x mit der Eigenschaft q
$\Pi\ (M)$	Partition (Menge paarweise fremder Untermengen) der Menge M
$\Pi_k\ (M)$	Partition aus k Untermengen der Menge M
$\bigcup M_i$	Vereinigungsmenge aller Mengen M_i

Folgen,
reguläre Mengen,
reguläre Ausdrücke:

$\langle.....\rangle$	Folge der Elemente ....
$\{\langle...\rangle,\langle...\rangle,....\langle..\rangle\}$	Ereignis (Menge von Folgen)
$\cup$	Disjunktion *(oder)*
$\cdot$	Kettenprodukt *(gefolgt von)*
$*$	Iteration
$\bigcup E_i$	Disjunktion aller Ereignisse E_i
Λ	Leerfolge (Folge der Länge null)
$\equiv$	Äquivalenz

Variablen-Namen,
Abkürzungen:

a, b, c ...	Variablen für Elemente von Folgen, und für atomare reguläre Ausdrücke
AD(ρ)	Adresse eines Wortes der Sekundär-Hilfsliste
$\mathfrak{A}$	Akzeptor
BA(h)	Basis-Adresse von Suchlistenblock Nr. h
$D_s\rho$	Ableitung des regulären Ausdruckes ρ bezüglich der Folge s
ϑ	endlicher Automat ohne Ausgangszuordner
E	Ereignis (Menge von Folgen)
f	Folgezustands-Index

h	Hilfslisten-Index (Zeilen-Nummer)
i, j	Indices oder Lauf-Variablen
k	Anzahl von Klassen
$\mathfrak{K}$	klassifizierender Automat
l(s)	Länge der Folge s
l(X)	Länge des Alphabetes X
l(h)	Länge des Suchlistenblockes (der Zeile) h
m	Länge des Eingabealphabetes X
M	Menge
$\mathfrak{M}$	Mealy-Klassifikator
n	Länge des Ausgabealphabetes Y
$\mathfrak{N}$	nicht-deterministischer Akzeptor
p	Positions-Index
q	Eigenschaft
r	Anzahl der Elemente einer Zustandsmenge Z
$\mathfrak{R}$	Moore-Klassifikator
s, t	Folgen
s^{-1}, t^{-1}	Reflexion der Folge s, bzw. t
Sw	Schwarz
τ	Transduktor
$U(\mathfrak{A})$	Menge aller von $\mathfrak{A}$ akzeptierten Ereignisse
Ws	Weiß
Ws/1	Weiß, erstmalig auftretend
x	Eingabevariable
x_i	i-tes Symbol des Eingabealphabetes X
X	Eingabealphabet
y	Ausgabevariable
y_i	i-tes Symbol des Ausgabealphabetes Y

Y	Ausgabealphabet
z	Zustandsvariable
z_i	i-tes Element der Zustandsmenge Z
z_f	Folgezustand oder Endzustand
z_o	Anfangszustand
z_v	Vorzustand
Z	Zustandsmenge
Z'	Menge aller Endzustände, $Z' \subseteq Z$
Z_o	Menge aller Anfangszustände, $Z_o \subseteq Z$
Z'', Z''_k	Partition der Endzustände, $Z'' = \Pi(Z')$
Z_i	i-tes Element der Partition Z'', $Z_i \subset Z''$
α, β, γ	Variablen für reguläre Ausdrücke
δ	Übergangsfunktion
δ^{-1}	inverse Übergangsfunktion
$\delta(z_i, x)$, $\delta(z_i, s)$, $\delta(z_i, \rho)$	Folgezustand von z_i
$\Delta(\rho)$	Delta-Funktion des regulären Ausdruckes ρ
λ	Ausgabefunktion
Λ	Leerfolge (Folge der Länge null)
$\Pi''_k(Z')$	k-Partition (Menge von k paarweise fremden Untermengen) der Menge $Z' \subseteq Z$
$\Pi''_k(\rho)$	Akzeptierungs-Partition (Menge paarweise fremder Unterereignisse) von ρ
ρ	regulärer Ausdruck oder reguläres Ereignis
ρ_{oi}	Vor-Ereignis des Zustandes z_i eines Akzeptors
ρ_i	durch Zustand z_i akzeptiertes Ereignis
$\rho(i)$	i-tes Element der Partition $\Pi_k(\rho)$, $i \leq k$
ρ^{-1}	Reflexion des Ereignisses ρ

Erläuterung einiger Begriffe

Ableitung: (Definition 2.3.9) *Die Ableitung* $D_s\rho$ *eines regulären Ausdruckes* ρ *bezüglich einer Folge* s ist die Menge aller Nachsilben t des Ausdruckes ρ, deren Kettenprodukte st in ρ enthalten sind, d.h. $D_s\rho = \{t|st \in \rho\}$.

abstrakte Synthese: Syntheseprozeß mit dem Ziel, einen abstrakten Automaten als Lösung zu gewinnen, wenn entweder eine Menge von Eingabefolgen (oder ggf. die Partition einer solchen Menge) oder einer oder mehrere reguläre Ausdrücke gegeben sind.

ähnlich: (Definition 2.3.11) Zwei reguläre Ausdrücke sind *ähnlich*, wenn sie ineinander umgewandelt werden können unter ausschließlicher Anwendung der drei folgenden Regeln: $\alpha \cup \alpha = \alpha$, $\alpha \cup \beta = \beta \cup \alpha$, $(\alpha \cup \beta) \cup \gamma = \alpha \cup (\beta \cup \gamma)$.

äquivalent: Zwei Zustände eines Automaten sind *äquivalent*, wenn dieser in jedem dieser Zustände bei beliebiger Eingabefolge die gleiche Ausgabefolge abgibt (oder bei Akzeptoren: die gleiche Entscheidung trifft). Zwei Automaten sind *äquivalent*, wenn zu jedem Zustand des einen Automaten ein äquivalenter Zustand des anderen Automaten existiert.

akzeptieren: s. unter *Akzeptor*

Akzeptor: (Synonyme: Detektor, Erkennender Automat): diskreter Automat ohne Ausgangszuordner mit drei Kategorien von Zuständen: Anfangszustände, Endzustände, gewöhnliche Zustände. Folgen oder Ereignisse gelten als *akzeptiert durch einen Akzeptor*, wenn sie diesen von einem Anfangszustand in einen Endzustand überführen (Def. 2.2.4). Nicht akzeptierte Folgen oder Ereignisse werden als *rückgewiesen* bezeichnet. Folgen oder Ereignisse gelten als *akzeptiert durch Zustand* z_i *eines Akzeptors*, wenn sie diesen von Zustand z_i in einen Endzustand überführen (Def. 2.3.10).

Analyseproblem bei Akzeptoren: Die Aufgabe, denjenigen regulären Ausdruck zu finden (oder diejenige Menge aller Folgen), der (die) das von einem gegebenen Akzeptor insgesamt akzeptierte Ereignis beschreibt (s. auch: *Akzeptor*).

atomar: hat bei Zweigen in Graphen oder bei Ausdrücken die Bedeutung: nicht zusammengesetzt, d.h. nicht zerlegbar.

Auffächerung: graphische Transformation entsprechend der KLEENE'schen Gleichung $\alpha^* = \alpha^{*n} \cup \alpha^{n+1}\alpha^*$ (Regel 3.3.2). Die graphische Darstellung der *beschränkten Iteration* α^{*n} hat hierbei die Gestalt eines Fächers (vgl. Bild 3.3.1).

beschränkte Iteration: α^{*n} ist definiert durch $\alpha^{*n} = \bigcup_{i=0}^{n} \alpha^i$ im Gegensatz zu $\alpha^* = \bigcup_{i=0}^{\infty} \alpha^i$, welches die Definition der ("unbeschränkten") Iteration ist.

black-box-Graph: (Bild 2.3.1) Der *black-box-Graph* eines einfachen Akzeptors ist dessen uneigentlicher Zustandsgraph, der nur je einen Anfangs- und Endknoten in Form eines Extremalknoten besitzt.

Bündel: Ein Bündel eines Knotens j ist eine Menge von Zweigen, die mit j verbunden sind. Je nach Richtung dieser Zweige werden unterschieden: *abgehendes, ankommendes, gemischtes Bündel.* Ein *Gesamtbündel* umfaßt sämtliche mit j verbundene Zweige der angegebenen Richtung(en).
Bei Suchlistendarstellung (Kap. 5.1) von Automaten ist ein *Zeilenblock* der Deskriptor eines nicht-gemischten Gesamtbündels.

charakteristische Ableitungen: (Satz 2.3.1) Jeder reguläre Ausdruck ρ hat eine endliche Anzahl *Typen* von *Ableitungen*. Die Menge aller Ableitungen unterschiedlichen Typs von ρ heißt *Satz charakteristischer Ableitungen von* ρ.

Darstellung von Ereignissen: Ein Ereignis ist durch einen Automaten in Zustand z_v *dargestellt*, wenn zu jeder Folge s dieses Ereignisses ein Folgezustand $\delta(z_v, s)$ existiert. Der Begriff gilt sinngemäß auch für Folgen (Ereignisse, die aus nur einer Folge s bestehen).

Dekomposition: die Zerlegung von Graphen in Unter-Graphen als Abbildung der Zerlegung von Ausdrücken einer Algebra in Unter-Ausdrücke.

Δ-Funktion: (Definition 2.3.8) Die *Δ-Funktion* Δ(ρ) *eines regulären Ausdruckes* ρ, der das reguläre Ereignis R beschreibt, ist wie folgt definiert: $\Lambda \in R \rightarrow \Delta(\rho) = \Lambda$, und: $\Lambda \notin R \rightarrow \Delta(\rho) = \Phi$.

δ-Funktion: Übergangsfunktion diskreter Automaten. Bei gegebenem Vorzustand z_v und Eingabesymbol x_i gibt die δ-Funktion den Folgezustand z_f an mit $z_f = \delta(z_v, x_i)$. Durch Verallgemeinerung unter Zulassung von Folgen s der Länge $l(s) > 1$ und Mengen von Folgen (Ereignissen) ρ sind auch die Formen $z_f = \delta(z_v, s)$ und $z_f = \delta(z_v, \rho)$ definiert.

Deskriptor: Ein Ausdruck einer Sprache oder einer Algebra, der ein Ereignis oder eine Aussage beschreibt, ist *Deskriptor* dieses Ereignisses, bzw. dieser Aussage.

Determinisierung: Umwandlung eines nicht-deterministischen diskreten Automaten $\mathfrak{A}$ in einen dazu äquivalenten deterministischen Automaten $\mathfrak{A}'$.

deterministisch: Ein diskreter Automat ist deterministisch, wenn zu jeder Eingabefolge s und jedem Zustand z_v höchstens ein Folgezustand $\delta(z_v, s)$ existiert.

Duplex-Darstellung: Die *Duplex-Darstellung* eines endlichen Automaten $\mathfrak{E}$ besteht aus dessen numerischer Spezifikation, verbunden mit der numerischen Spezifikation des dazu inversen Automaten $\overline{\mathfrak{E}}$.

einfach: Ein Akzeptor $(X, Z, \delta, \{z_o\}, \{z_f\})$ ist *einfach*, wenn er genau je einen Anfangszustand z_o und Endzustand z_f besitzt, wobei $z_o \neq z_f$, sowie z_o <u>nie</u> und z_f <u>nur</u> Folgezustand ist.

Ereignis: eine Menge, deren Elemente Folgen sind. (Spezialfälle von Ereignissen sind: die Folge, das Eingabesymbol, die Leerfolge (Λ), das Leerereignis (Φ).)

Extremalknoten: Ein Extremalknoten in einem gerichteten Graphen ist ein Knoten, der nur mit ankommenden oder nur mit abgehenden Zweigen verknüpft ist.

Fächer: Der Graph der *beschränkten Iteration* (Bild 3.3.1) wird *Fächer* genannt.

Folgezustand: Bei Zustandsübergang der Zustand <u>nach</u> dem auslösenden Ereignis.

Graph: Netzwerk aus Zweigen, die Knoten miteinander verbinden.

graphische Transformation: Substitution eines (Unter-)Graphen durch einen äquivalenten anderen (Unter-)Graphen als Abbildung der Substitution eines (Unter-)Ausdruckes a einer Algebra durch einen äquivalenten (Unter-)Ausdruck b = a.

Hilfsliste: Eine Automatentafel in *Suchlistendarstellung* /59/ besteht aus *Suchliste*, *Hilfsliste*, und *Ausgabeliste*. Die *Hilfsliste* enthält für jeden Zustand ein Wertepaar, welches Position und Länge des zugehörigen Zeilenblockes innerhalb der Suchliste angibt.

idempotent: Eine zweistellige algebraische Verknüpfung ist *idempotent*, wenn für diese das Idempotenzgesetz gilt. Dies lautet für eine durch das Konnektivsymbol "·" bezeichnete Verknüpfung: $a \cdot a = a$

indeterministisch: Ein diskreter Automat ist indeterministisch, wenn er mindestens einen Zustand z_v hat, der für das gleiche Eingabesymbol x_i mehr als einen Folgezustand $\delta(z_v, x_i)$ hat.

indeterministische Folgezustände: z_i und z_j eines Vorzustandes z_v sind infolge Mehrdeutigkeit der Übergangsfunktion δ Folgezustände von z_v auf die gleiche Eingabefolge s.

indeterministische Struktur 1. Art: Unterstruktur eines Zustandsdiagrammes mit *indeterministischen Folgezuständen* (s. dort), die *rückwärts-äquivalent* sind (vgl. Bild 3.2.a)

indeterministische Struktur 2. Art: Unterstruktur eines Zustandsdiagrammes mit *indeterministischen Folgezuständen* (s. dort), die *rückwärts-unterscheidbar* sind (vgl. Bild 3.2 b)

inverse Übergangsfunktion: δ^{-1} bildet das Kartesische Produkt $Z \times X$ der Zustandsmenge Z und des Eingabe-Alphabetes X eines diskreten Automaten in die Zustandsmenge Z ab derart, daß Folgezustände als Argumente und Vorzustände als abhängige Variablen auftreten.

isoliert: (Definition 5.4.1) Ein Knoten eines gerichteten Graphen sei *isoliert* genannt, wenn er mit keinem einzigen abgehenden Zweig verknüpft ist. Ein Zustand sei *isoliert* genannt, wenn er durch einen isolierten Knoten dargestellt wird.

Iteration: Operation der KLEENE'schen Algebra der regulären Mengen (Definition 2.1.3). Die *Iteration* α^* des regulären Ausdruckes α ist definiert durch: $\alpha^* = \Lambda \vee \alpha^1 \vee \alpha^2 \vee \alpha^3 \vee \alpha^4 \ldots$

Kaskade: Serienschaltung von Automaten

klassifizierender Akzeptor: Akzeptor, dessen Endzustandsmenge eine Partition ist. (Vgl. Def. 4.1.1/2/3.)

Komposition: Verknüpfung von Graphen als Abbildung der operativen Verknüpfung von Ausdrücken einer Algebra.

Leerfolge: Folge der Länge null (wird bezeichnet durch das Symbol Λ)

Leer-Ereignis: Menge von Folgen, wobei die Anzahl der Folgen null ist (wird bezeichnet durch das Symbol Φ).

Leermenge: Menge, deren Anzahl der Elemente null ist.

Leerpfeil: Synonym für *Leerzweig*

Leerzweig: graphischer Zweig, dem das Symbol Λ zugeordnet ist (s. Leerfolge).

Mealy-Akzeptor: (angegeben durch Def. 4.2.2) endlicher Transduktor mit Anfangszustand(-zuständen), wobei zwei Arten von Übergängen unterschieden werden: solche mit und solche ohne Ausgabesymbol.

Menge: Ensemble von Elementen (wie beispielsweise Symbole)

minimal (Synonym: vollständig reduziert): Ein diskreter Automat $\mathfrak{A}$ mit r Zuständen ist minimal (ein Minimal-Automat), wenn kein zu $\mathfrak{A}$ äquivalenter Automat $\mathfrak{A}'$ existiert, der weniger als r Zustände hat.

Minimisierung (Synonym: vollständige Reduktion) bei diskreten Automaten: Umwandlung eines Automaten $\mathfrak{A}$ in einen dazu äquivalenten *minimalen* Automaten.

Mitgliedschaftsrelation: Die durch $\in$ (*Element von*) oder $\notin$ (*nicht Element von*) ausgedrückte Relation zwischen einem Element und einer Menge wird *Mitgliedschaftsrelation* genannt.

Moore-Akzeptor: Spezialfall des *Moore-Klassifikators* (s. dort) mit nur einer einzigen Klasse von Endzuständen.

Moore-Klassifikator: (angegeben durch Def. 4.1.4) klassifizierender Akzeptor, wobei nur die Klassenzugehörigkeit der Endzustände oder die Rückweisung durch einen Ausgangszuordner angezeigt wird.

Partition: Die Partition einer gegebenen Menge M ist eine Menge paarweise fremder (d.h. sich nicht überschneidender) Untermengen von M.

Potenz-Kette: graphische Darstellung einer Potenz der KLEENE'schen Algebra der regulären Mengen.

rechts-eindeutige Abbildung: Abbildung einer Menge M_1 (Vorbereich) in eine Menge M_2 (Nachbereich), wobei mindestens ein Element von M_1 einem einzigen Element von M_2 zugeordnet wird.

Reduktion: Umwandlung eines Automaten in einen dazu äquivalenten Automaten unter Verringerung der Zustands-Anzahl durch Verminderung der Zahl der Paare *äquivalenter* Zustände.

Reflexion: Die *Reflexion s^{-1} einer Folge s* ist die Rückwärts-Folge der Elemente von s (Def. 3.1.2). Die *Reflexion ρ^{-1} eines regulären Ausdruckes* ρ ist die Menge der Reflexionen s^{-1} aller Folgen s des regulären Ereignisses ρ (Definition 3.1.3), d.h. $\rho^{-1} = \{s^{-1} | s \in \rho\}$.

reguläre Menge: s. unter *reguläres Ereignis*

regulärer Ausdruck: Ausdruck der KLEENE'schen Algebra der regulären Mengen

reguläres Ereignis: (Synonym: *reguläre Menge*) Ereignis, das durch einen endlichen Automaten oder einen regulären Ausdruck darstellbar ist.

Residuen-Graph: uneigentlicher Zustandsgraph, der bei Moore-Klassifikatoren oder -Transduktoren nur markierte Knoten i (mit $\lambda(z_i) \neq \Lambda$) oder bei Mealy-Klassifikatoren oder -Transduktoren nur (i.allg. nicht-atomare) markierte Zweige ρ_{ij} besitzt (mit $\lambda(z_i, \rho_{ij}) \neq \Lambda$).

rückwärts-äquivalent: Zwei Zustände z_i und z_j eines Automaten sind *rückwärts-äquivalent*, wenn aufgrund der inversen Übergangsfunktion δ^{-1} bei beliebiger inverser Eingabefolge s^{-1} der Automat die gleiche Entscheidung trifft unabhängig davon, ob er sich in Zustand z_i oder z_j befand. Unter "gleicher Entscheidung" wird hierbei verstanden: <u>beim Transduktor</u> die Gleichheit der inversen Ausgabefolge aufgrund der inversen Ausgabefunktion λ^{-1}, <u>beim Akzeptor</u> die Gleichheit der Mitgliedschaftsrelation ($\in$ oder $\notin$) der Vorzustände $\delta^{-1}(z_i,s^{-1})$ und $\delta^{-1}(z_j,s^{-1})$ zur Menge Z_o der Anfangszustände (s. unter *Akzeptor*).

rückwärts-minimal: Ein Automat ist *rückwärts-minimal*, wenn er keine *rückwärts-äquivalenten* Zustände besitzt.

Rückwärts-Minimisierung: Umwandlung eines Automaten in einen äquivalenten *rückwärts-minimalen* Automaten.

Rückwärts-Reduktion: Umwandlung eines Automaten in einen dazu äquivalenten Automaten mit verringerter Zahl der Zustände durch Verringerung der Zahl der Paare *rückwärts-äquivalenter* Zustände.

rückwärts-unterscheidbar: Zwei Zustände eines Automaten sind *rückwärts-unterscheidbar*, wenn sie nicht *rückwärts-äquivalent* sind.

Rückweisung: s. unter *Akzeptor*

Signalflußgraph: /37,38/ linearer gerichteter Graph als Deskriptor linearer algebraischer Ausdrücke. Mit jedem Knoten i wird ein *Signal* u_i verknüpft. Mit jedem Zweig ij, der Knoten i mit Knoten j verbindet, wird ein Koeffizient t_{ij} *(Transmission)* verknüpft. (vgl. *Signalflußgraphen-Methode*)

Signalflußgraphen-Methode: /37,38/ Methode zur Lösung von linearen algebraischen Gleichungen oder Gleichungssystemen mittels graphischer Transformationen an *Signalflußgraphen*. Anwendungsbeispiel: Berechnung der Übertragungseigenschaften linearer elektrischer Netzwerke.

Silbe: Eine *Silbe einer Folge* s ist eine Teilfolge von s (Definition 2.3.2). Eine *Silbe eines regulären Ausdruckes* ρ, der das Ereignis R beschreibt, ist eine Silbe einer Folge $s \in R$.

stabil: Ein Zustand z_v eines Automaten über dem Alphabet X ist *stabil*, wenn er für jedes Symbol $x_i \in X$ gleichzeitig Folgezustand ist mit $z_v = \delta(z_v, x_i)$. Er wird graphisch dargestellt durch einen Knoten mit Rückkehrschleife für alle $x_i \in X$.

Suchliste: Eine Automatentafel in *Suchlistendarstellung* (s. dort) besteht aus *Suchliste*, *Hilfsliste* und *Ausgabeliste*. Die *Suchliste* ist hierbei die Gesamtheit aller *Zeilenblöcke*.

Suchlistendarstellung: von Automatentafeln (vgl. Kapitel 5.1): Die δ-Funktion ist nicht in Matrixform spezifiziert, sondern durch je einen *Zeilenblock* pro Zustand. Jeder Zeilenblock enthält die Spezifikation jedes Überganges in Form einer *Übergangsregel* (s. dort). Der Index des Zeilenblockes ist der Vorzustand z_v. Im Zeilenblock befindet sich die Menge aller Paare (x, z_f). Bei unvollständig definierter oder mehrdeutiger δ-Funktion wird die Übergangsregel für ein bestimmtes x durch Absuchen des betr. Zeilenblockes gefunden, was die Bezeichnung *Suchliste* erklärt.

Syntheseproblem: s. unter *abstrakte Synthese*

Transduktor: diskreter Automat mit Ausgangszuordner

Typ: reguläre Ausdrücke, die unterschiedliche Form haben, aber das gleiche Ereignis beschreiben, sind *äquivalent* oder *vom gleichen Typ*.

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

Übergangsregel: Eine algebraische Struktur kann als Akzeptor-Zustandsgraph dargestellt werden. Axiome werden als Anfangszustände dargestellt in der Form (z_o). Abgeleitete Regeln werden als Übergänge *(Übergangsregeln)* dargestellt in der Form $(z_v, x\ z_f)$ für $z_f = \delta(z_v, x)$. Die *Suchlisten-Darstellung* einer Automatentafel beruht auf der Beschreibung von Übergängen durch *Übergangsregeln.*

uneigentlich: Ein Zustandsgraph ist *uneigentlich,* wenn er mindestens einen Zweig besitzt, dem ein nicht-atomarer regulärer Ausdruck oder das Symbol für "Leerfolge" zugeordnet ist.

unterscheidbar: (s. auch unter *rückwärts-unterscheidbar);* zwei Zustände eines Automaten sind *unterscheidbar,* wenn sie nicht *äquivalent* sind.

unzugänglich: (Definition 5.4.1) Ein Knoten eines gerichteten Graphen sei *unzugänglich* genannt, wenn er mit keinem einzigen ankommenden Zweig verknüpft ist. Ein Zustand sei *unzugänglich* genannt, wenn er durch einen unzugänglichen Knoten dargestellt wird.

Verschmelzung: Zwei nicht (rückwärts-)unterscheidbare Zustände z_i und z_j eines Automaten werden durch den Zustand z_i ersetzt. Hierbei müssen in der gesamten δ-Funktion des Automaten alle Folgezustände z_j durch z_i ersetzt werden.

Vorzustand: Bei Zustandsübergang der Zustand vor dem auslösenden Ereignis

Zeilenblock: Bei einer Automatentafel in *Suchlisten-Darstellung* ist jedem Zustand z_h ein *Zeilenblock* zugeordnet, der für jeden explizit definierten Übergang von z_h durch Eingabesymbol x_i in Folgezustand z_f ein Wertepaar (x_i, z_f) enthält, das *Übergangsregel* genannt wird. Ein *Zeilenblock* ist die Gesamtheit aller Übergangsregeln des gleichen Vorzustandes z_h

zurückweisen: s. unter *Akzeptor*

Zweig: in Graphen Verbindungslinie zwischen zwei Knoten.

1. Einleitung

Bei theoretischer Behandlung von Erkennungsproblemen wird davon ausgegeangen, daß von einem zu erkennenden Muster durch einen Wandler eine formale Beschreibung aus Symbolen über einem endlichen Alphabet gewonnen wird, welche als Ausgabewort des Wandlers oder als "W-Wort" bezeichnet sei (Bild 1.1a). Da ein Muster im allgemeinen auch dann erkannt werden soll, wenn es von seiner Idealgestalt etwas abweicht, wird jeweils eine Menge M_i von Worten einer einzigen Bedeutung i zugeordnet. Eine solche Menge M_i wird Klasse i genannt. Die Klassenzuordnung erfolgt durch einen Klassifikator, wofür Bild 1.1a ein Beispiel mit k Klassen zeigt. Jeder der k Mengen M_i ist ein Indikator i zugeordnet. Illegale Worte, d.h. Worte, die keiner der k Klassen angehören werden zurückgewiesen, wobei keiner der Indikatoren etwas anzeigt.

Beim Zeichenerkennungsproblem sind die Mengen M_i meist sehr groß. Deshalb wird vor der eigentlichen Klassifikation eine Informationsreduktion vorgenommen (Bild 1.1b), wobei bedeutungsmäßig irrelevante Information ausgeschieden wird. Hierdurch wird die Anzahl der zu klassifizierenden Worte verringert. Ein Ausgabewort des Systems zur Informations-Reduktion sei als "R-Wort" bezeichnet.

Bei Parallel-Klassifikatoren müssen alle Eingabeworte des Klassifikators gleiche Länge und gleiche Wortstruktur haben, da für jede Stelle f_i des Wortes ein spezieller Eingang vorgesehen ist. Bild 1.2b zeigt ein zur Demonstration erfundenes Beispiel für eine Klasse M_i mit der Wortlänge 2, wobei das Alphabet X der Worte $w \in M_i$ nebst Interpretation der Symbole $x \in X$ in Bild 1.2a dargestellt ist.

Die vorliegende Arbeit befaßt sich mit ausschließlich sequentiellen Methoden der Informationsverarbeitung bei Erkennungsproblemen. Hierbei sind die Worte Folgen, weshalb im

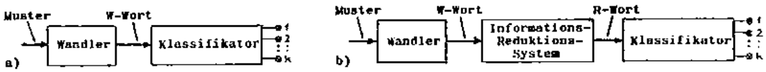


Bild 1.1 Prinzipieller Aufbau von Erkennungssystemen

Falle der Klassifikation eine Klasse durch eine Menge E_i von Folgen gebildet wird. Mengen von Folgen heißen in der Mengenlehre "Ereignisse". Bild 1.2c zeigt auf der Grundlage des Alphabetes X nach Bild 1.2a ein Beispiel für eine aus einem Ereignis E_i gebildete Klasse. Die Folgen $s \in E_i$ dürfen hierbei verschiedene Länge haben (vgl. Bild 1.2c), da der Klassfikator nur einen Eingang für ein einziges Symbol $x \in X$ hat.

Zur Klassifikation von Folgen liefert die Automatentheorie ein Hilfsmittel zur formalen Beschreibung diskreter Systeme für den Spezialfall mit nur einer einzigen Klasse, wobei die "Rückweisungs-Klasse" nicht mitgezählt ist. Es handelt sich hierbei um den als "Endlicher Akzeptor" oder "Erkennender Automat" bekannten Typ des Endlichen Automaten.

Auch bei Mehrklassenproblemen ist eine sequentielle Klassifikation mit Hilfe Endlicher Automaten möglich /32,59,61/. Darüber hinaus sind auch Operationen der Informationsreduktion auf sequentiellem Wege über geeignet entworfene Endliche Automaten durchführbar /27,29-32,47/.

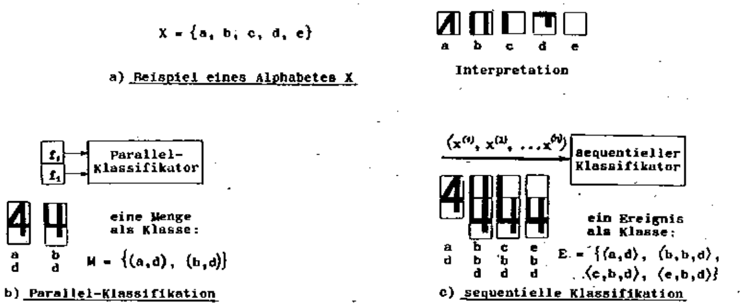


Bild 1.2 Zur Veranschaulichung des Klassifikationsbegriffes

Obwohl einige Autoren darauf hinweisen, daß das Automatenmodell des *Akzeptor* auf das Zeichenerkennungsproblem anwendbar ist (z.B. /48,49/), klafft eine Lücke zwischen der Automatentheorie und ihren Anwendung auf Probleme der Automatischen Zeichenerkennung. Es sind zwar einige Veröffentlichungen bekannt, in welchen meist bei einfachen Teilproblemen die Anwendung von Schaltwerken für Erkennungszwecke beschrieben wird /27,39,30,31,32/. Diese Schaltwerke wurden jedoch auf heuristische Weise entworfen ohne die Anwendung automatentheoretischer formaler Mittel, die über die Benützung von Zustandsdiagrammen zur Funktionsbeschreibung hinausgehen. FOGEL et. al. /48/ beschreiben ein automatisierbares Verfahren zur Synthese abstrakter Automaten, wobei auch die Automatische Zeichenerkennung als mögliches Anwendungsgebiet erwähnt wird. Dieses als *evolutionary programming* bezeichnete Verfahren beruht auf der Mutationsmethode (nach I. RECHENBERG /12/) und ist vom Prinzip her nicht auf der Automatentheorie begründet.

In einer Vorstufe zur hier vorliegenden Arbeit wurde eine als *Suchlistendarstellung* bezeichnete Listenstruktur zur Darstellung abstrakter Automaten verwendet zur automatischen Synthese von Automaten für Klassifikationszwecke /59/. Ausgehend von nicht abgeschlossener Problemformulierung sieht der Algorithmus Lernfähigkeit des Systemes durch sukzessives Hinzufügen einzelner Übergänge in die Liste vor. Ein speziell hierfür entwickeltes Programm ermöglicht die rechenzeit-effektive Reduktion dieses in Suchlistendarstellung gegebenen Automaten /59/. Zu einem späteren Zeitpunkt wird von BERNSTEIN/ WILLIAMS /20/ die Anwendung einer ähnlichen wachsenden Listenstruktur für Klassifikationszwecke beschrieben, die von den Autoren jedoch nicht als Automatentafel interpretiert und behandelt wird. In beiden Arbeiten sind die zugrundeliegenden automatentheoretischen Vorstellungen trivial, abgesehen vom erwähnten Reduktions-Algorithmus.

Die beiden bei der Automatischen Zeichenerkennung angewandten oben erwähnten Algorithmen zur Automatensynthese gehen nicht von einer abgeschlossenen Aufgabenstellung aus, sondern von lernfähigen Systemen, die bereits bei noch unvollkommener Problemformulierung die Durchführung von Syntheseschritten erfordern. Bei Vorliegen geschlossener Probleme ist auf diesem Gebiet jedoch keine formal durchgeführte Automatensynthese bekannt. Für das Bestehen dieser Lücke zwischen Automatentheorie und Zeichenerkennung können u.a. zwei Gründe genannt werden.

1). Es sind keine konkreten Vorschläge zur Anwendung des relativ einfachen Akzeptor-Modelles auf die komplexeren Probleme der Automatischen Zeichenerkennung bekannt.

2.) Bekannte Verfahren zur Akzeptor-Synthese aus regulären Ausdrücken /33,34,36,41,46,52/ sind bei Anwendung ohne Hilfe eines Digitalrechners relativ kompliziert, und haben (mit Ausnahme von /52/) keine eindeutige Lösung.

In der vorliegenden Arbeit wird ein Verfahren zur Akzeptor-Synthese aus regulären Ausdrücken beschrieben, das auf der Anwendung gerichteter Graphen beruht, einfach ist, und das Erreichen der Minimallösung gestattet. Darüber hinaus wird auf der Grundlage dieser Synthesemethode ein Ansatz zur Lösung der komplizierteren Probleme der Zeichenerkennung beschrieben, was Probleme der Klassifikation und Informationsreduktion umfaßt. Ein weiterer Abschnitt der Arbeit beschreibt eine Modifikation des beschriebenen Synthese-Verfahrens, die auf einfache Weise eine Simulation des Verfahrens auf Digitalrechnern ermöglicht.

Im Anschluß an das Einleitungs-Kapitel wird eine Einführung in die theoretischen Grundlagen gegeben. Zunächst wird die KLEENE'sche Algebra der regulären Mengen und der Begriff des Akzeptors eingeführt. Daran anschließend werden bekannte Verfahren zur Akzeptor-Synthese aus regulären Ausdrücken beschrieben. Da die Algebra der regulären Mengen auf dem Gebiet der Nachrichtentechnik relativ wenig bekannt ist, nimmt der Abschnitt über die Theoretischen Grundlagen innerhalb dieser Arbeit einen breiten Raum ein.