

### 3. Ein neues graphisches Verfahren zur Synthese von Akzeptoren aus regulären Ausdrücken

In den folgenden Kapiteln 3.1 - 3.3 wird ein Verfahren zur Synthese von Akzeptoren aus regulären Ausdrücken vorgeschlagen, das sich aus einer inversen Anwendung der Analyse-Methode nach BRZOZOWSKI/McCLUSKEY ergibt unter Erweiterung des Satzes von Regeln zur Transformation graphischer Strukturen. Der modifizierte Satz von Regeln ermöglicht hierbei - im Gegensatz zu bekannten Verfahren - eine rein graphische Durchführung und eine Erreichung der Minimallösung. Die einzelnen Schritte betreffen hierbei Unterstrukturen des Graphen, weshalb die Durchführung jedes einzelnen Transformations-Schrittes auch bei kompliziertem Gesamtproblem stets übersichtlich bleibt. Darüber hinaus gestatten die Hilfsmittel des Verfahrens die Zerlegung von Problemen in Teilaufgaben, sowie zur Lösung von Problemen der Klassifikation und der Informationsreduktion durch Automaten-synthese die Komposition von Akzeptor-Strukturen zu klassifizierenden Automaten und Transduktoren. (Auf Transduktoren und klassifizierende Automaten und deren Synthese wird in Kapitel 4 und 4.2 näher eingegangen.)

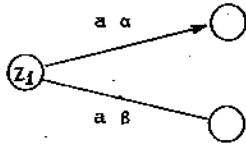
Das Analyse-Verfahren nach BRZOZOWSKI/McCLUSKEY jr. geht von uneigentlichen und deterministischen Zustandsdiagrammen aus. Denn zwei nicht-atomare Zweige  $\alpha a$  und  $\beta a$ , die vom gleichen Knoten  $z_1$  ausgehen und das gleiche einzige Anfangssymbol  $a$  haben (Bild 3.1.a), sind nicht indeterministisch, wenn  $\alpha \neq \beta$  ist, wobei  $\alpha \neq A$  und  $\beta \neq A$ . Der Folgezustand  $\delta(z_1, \alpha a)$  oder  $\delta(z_1, \beta a)$  ist in diesem Fall eindeutig bestimmt. Der entsprechende Graph ist ein uneigentlicher Zustandsgraph. Sind die beiden von  $z_1$  ausgehenden Zweige jedoch atomar (Bild 3.1.b), so stellt die gleiche graphische Struktur einen indeterministischen Übergang dar, da der Folgezustand mit  $z_{f1} = z_{f2} = \delta(z_1, a)$  nicht eindeutig bestimmt ist. Die letztere Struktur ist gleichzeitig infolge der Atomarität ihrer Zweige ein echtes Zustandsdiagramm.

Im Gegensatz zum Verfahren nach BRZOZOWSKI/McCLUSKEY jr. sind bei dem in folgendem Abschnitt vorgeschlagenen Verfahren auch solche Strukturen zugelassen, die dem Schema nach Bild 3.1.b entsprechen.

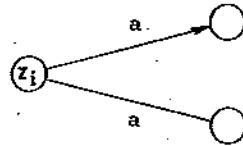
Die hierdurch bedingte Erweiterung des Definitionsbereiches zur Begründung des vorgeschlagenen Verfahrens ermöglicht die Definition einer Relation zwischen Knoten in Graphen, die als *Rückwärts-Äquivalenz* bezeichnet sei und zur bekannten Relation der *Äquivalenz* dual ist. In Kapitel 3.1 wird u.a. gezeigt, daß zwei rückwärts-äquivalente Knoten  $z_1$  und  $z_2$  stets indeterministische Folgezustände zweier gleicher Vorzustandsmengen beschreiben (z.B. Bild 3.2.1), daß aber umgekehrt nicht alle indeterministischen Folgezustände rückwärts-äquivalent sind (z.B. Bild 3.2.b). Zur leichteren Unterscheidung dieser beiden Typen indeterministischer Strukturen seien die folgenden Bezeichnungen vereinbart.

Definition 3.1: Die Knoten zweier indeterministischer Folgezustände und die Gesamtheit ihrer ankommenden Zweige seien als "indeterministische Struktur" bezeichnet. Diese wird "indeterministische Struktur 1. Art" genannt, wenn die beiden Zustände rückwärts-äquivalent sind. Andernfalls wird sie "indeterministische Struktur 2. Art" genannt.

Es wird in Kapitel 3.1 gezeigt werden, daß indeterministische Strukturen 1. Art durch rückwärts-orientierte Anwendung bekannter elementarer Reduktions-Operationen in deterministische Strukturen umgewandelt werden können. Hierzu, sowie für die entsprechende Umwandlung indeterministischer Strukturen 2. Art werden Transformationsregeln für Graphen angegeben. Diese, sowie eine in Kapitel 3.2 eingeführte Regel zur Leerzweigentfernung und eine auf der normalen Reduktions-Operation beruhende Transformationsregel bilden die Grundlage des in Kapitel 3.2 bis 3.3 zusammengestellten Satzes von Regeln zur Transformation von Strukturen in allgemeinen Zustandsgraphen.

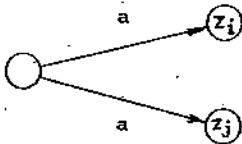


a) uneigentlicher Zustandsgraph: deterministisch

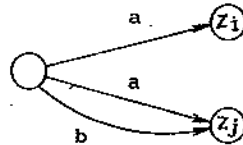


b) echter Zustandsgraph: in-deterministisch

**Bild 3.1** uneigentlicher und echter Zustandsgraph (Beispiele)



a) indeterministische Struktur erster Art



b) indeterministische Struktur zweiter Art

**Bild 3.2** indeterministische Struktur 1. und 2. Art (Beispiele)

Axiom	KLEENE'sche Algebra		lineare Algebra	
	Disjunktion	Produkt	Summe	Produkt
	a.	b.	a.	b.
1. Idempotenz	X	-	-	-
2. Assoziativität	X	X	X	X
3. Kommutativität	X	-	X	X
4. Adjunktivität	-	-	-	-
5. Distributivität	X	-	X	-
6. Komplementarität	-		-	
7. Null-Element-Existenz	-	∅	-	0
8. Identität-Existenz	∅	∧	0	1
9. weitere Operation (abgeleitet)	Iteration		Division	

**Tafel 3.1.1:** Strukturvergleich zwischen KLEENE'scher Algebra und linearer Algebra.

Bei manueller Anwendung beruht die Übersichtlichkeit und Wirksamkeit des Verfahrens auf der menschlichen Fähigkeit zur leistungsfähigen visuellen Struktur-Erkennung. In Kapitel 5 wird ein Vorschlag zur Modifizierung des Verfahrens für maschinelle Anwendung beschrieben, wobei die Methodik auf elementare Strukturen zurückgeführt wird. Diese Modifikation hat den Zweck, unter Verzicht auf komplizierte struktur-erkennende Unterprogramme die hohe Verarbeitungsgeschwindigkeit moderner Rechenanlagen ausnützen zu können.

### 3.1 Voraussetzungen

Zunächst sei anhand eines Vergleiches der Axiomatik von KLEENE'scher Algebra und gewöhnlicher linearer Algebra die Modifikation des Satzes von Regeln nach BRZOWSKI/McCLUSKEY (vgl. Kapitel 2.4) beschrieben und begründet. Tafel 3.1.1 zeigt die Gegenüberstellung der axiomatischen Strukturen dieser beiden Algebren zur Demonstration des Unterschiedes zwischen Signalflußgraphen nach MASON (vgl. Kapitel 2.4) und uneigentlichen Zustandsgraphen. Die Arbeit BRZOWSKI/McCLUSKEY geht nur von *deterministischen* Graphen aus, wobei sich gegenüber Signalflußgraphen nur folgende Unterschiede ergeben (vgl. Tafel 3.1.1 und auch Tafel 2.1.2):

1. Aus Axiom 3.b folgt Ausschluß der Kommutativität. Dieser Unterschied ist wesentlich. (Bild 3.1.1)
2. Aus Axiom 9. folgt Fortfall der Division; stattdessen wird die Iteration als Division interpretiert.

Letzterer Unterschied ist nur interpretativ, betrifft aber nicht die Methodik der Transformation gerichteter Graphen (vgl. Bild 2.4.3 c/d). Ersterer Unterschied bedeutet jedoch eine Einschränkung gegenüber der Signalflußgraphen-Methode, da das für die gewöhnliche lineare Algebra gültige Axiom 3 b für die KLEENE'sche Algebra nicht gilt. Es sei im Gegensatz hierzu vorgeschlagen, den umgekehrten Fall mit einzubeziehen, d.h. auch dasjenige Axiom der KLEENE'schen Algebra zu berücksichtigen, das für die gewöhnliche lineare Algebra nicht gilt. Es handelt sich hierbei um das Idempotenzgesetz der Disjunktion (Axiom 1.a in Tafel 3.1.1; vgl. auch Tafel 2.1.2). Bild 3.1.2\* zeigt die dem Idempotenzgesetz entsprechende graphische Äquivalenz. Deren Anwendung zu einer zusätzlichen Erweiterung der graphischen Methodik führt dazu, daß der Definitionsbereich über deterministische Strukturen hinaus auf das Gebiet der nicht-

\* auf Seite 79

deterministischen Zustandsgraphen ausgedehnt wird. Dies ist insofern vom Syntheseproblem her zulässig, als per definitionem auch der nicht-deterministische Akzeptor in seiner Funktion als erkennendes System eindeutig ist (vgl. Satz 2.2.1). Die Erweiterung des Definitionsbereiches gestattet im Hinblick auf die angestrebte Minimallösung von Syntheseproblemen die Anwendung zusätzlicher Transformationen. Vor einer detaillierteren Beschreibung des hierdurch gewonnenen Instrumentariums werden die folgenden Definitionen eingeführt mit dem Ziel, eine vereinfachte Ausdrucksweise im folgenden Kapitel zu ermöglichen.

Wenn zur Lösung eines Erkennungsproblem es ein Automat mit möglichst wenig Zuständen gesucht wird, so ist die optimale Lösung ein Automat, der gleichzeitig *minimal und rückwärts-minimal* ist. Es soll im Folgenden gezeigt werden, daß das Problem der Umwandlung *nicht-deterministischer Strukturen erster Art* in deterministische Strukturen (Rückwärts-Minimierung) im Prinzip auf das Minimierungsproblem bei Automaten zurückgeführt werden kann über die Definition eines *inversen Akzeptors*. Vor der Definition des *inversen Akzeptors* seien die wichtigsten Sätze für Minimierungsprobleme wiederholt.

Definition 3.1.1: Zwei Zustände  $z_i$  und  $z_j$  eines Akzeptors  $\mathcal{A}$  sind *äquivalent*, wenn deren Folgezustände auf jede beliebige Eingabefolge  $s$  jeweils gleiche Entscheidungen von  $\mathcal{A}$  darstellen /52/.

Unter *Gleichheit der Entscheidung* ist zu verstehen, daß jeweils entweder beide Folgezustände Endzustände sind, oder keiner der Folgezustände ein Endzustand ist. Diese Definition der Gleichheit der Entscheidung gilt wie auch alle folgenden Definitionen und Sätze auch für nicht-deterministische Akzeptoren.

Definition 3.1.4: Zur Übergangsfunktion  $\delta$  eines Akzeptors sei die *inverse Übergangsfunktion*  $\delta^{-1}$  definiert, welche die Menge aller Zustände in die Menge aller Vorzustände abbildet. Jeder spezielle Wert  $\delta^{-1}(z_j, x) = z_i \in Z$  dieser Funktion  $\delta^{-1}$  heißt *Vorzustand*, wobei  $z_j \in Z$  der Folgezustand und  $x \in X$  das Eingabesymbol ist.

Die inverse Übergangsfunktion  $\delta^{-1}$  gestattet die Rückwärtsbeschreibung von Zustandsfolgen des durch  $\delta$  definierten Akzeptors, wobei der Vorzustandsbegriff in der gleichen Weise verallgemeinert werden kann auf die Anwendung von Folgen und regulären Ausdrücken, wie dies vom Folgezustandsbegriff hinsichtlich der Funktion  $\delta$  bereits bekannt ist (vgl. Def. 2.2.1 ff. und Def. 3.1.1b).

Satz 3.1.3: Zu jedem Akzeptor  $\mathcal{N} = (X, Z, z_0, \delta, Z')$ , der das reguläre Ereignis  $\rho$  akzeptiert, existiert ein zu  $\mathcal{N}$  inverser Akzeptor  $\overline{\mathcal{N}} = (X, Z, \overline{z}_0 = z_0, \delta^{-1}, \overline{Z}' = Z')$ , der die Reflexion  $\rho^{-1}$  von  $\rho$  akzeptiert.

Beweis: Der Akzeptor  $\mathcal{N}$  sei in Form eines Graphen beschrieben. Für jede von  $\mathcal{N}$  akzeptierte Folge  $s$  existiert ein (Vorwärts-) Pfad von einem Anfangs-Knoten  $z_i \in Z_0$  zu einem End-Knoten  $z_j \in Z'$  von  $\mathcal{N}$ . Ein solcher Pfad ist definiert durch  $z_j = \delta(z_i, s)$ . Jeder dieser Pfade und kein anderer Pfad ist in Rückwärtsrichtung definiert von Knoten  $z_j$  nach Knoten  $z_i$  durch  $z_i = \delta^{-1}(z_j, s^{-1})$ . Aufgrund von Definition 3.1.3 ist  $\rho^{-1}$  genau die Menge aller Reflexionen von Folgen  $s \in \rho$ , also die Reflexion des von  $\mathcal{N}$  akzeptierten Ereignisses  $\rho$ . Infolge Definition 2.2.4 ist jede Folge  $s^{-1} \in \rho^{-1}$  durch den Akzeptor  $\overline{\mathcal{N}}$  genau dann akzeptiert, wenn bei diesem Akzeptor  $\delta^{-1}$  die Übergangsfunktion,  $Z' (= \overline{Z}_0)$  die Menge aller Anfangszustände, und  $z_0 (= \overline{z}'_0)$  die Menge aller Endzustände ist, q. e. d.

Der Graph eines Akzeptors  $\mathcal{N}$  wird in den Graphen des dazu inversen Akzeptors  $\overline{\mathcal{N}}$  umgewandelt, indem unter Beibehaltung

Definition 3.1.4: Zur Übergangsfunktion  $\delta$  eines Akzeptors sei die *inverse Übergangsfunktion*  $\delta^{-1}$  definiert, welche die Menge aller Zustände in die Menge aller Vorzustände abbildet. Jeder spezielle Wert  $\delta^{-1}(z_j, x) = z_1 \in Z$  dieser Funktion  $\delta^{-1}$  heißt *Vorzustand*, wobei  $z_j \in Z$  der Folgezustand und  $x \in X$  das Eingabesymbol ist.

Die inverse Übergangsfunktion  $\delta^{-1}$  gestattet die Rückwärtsbeschreibung von Zustandsfolgen des durch  $\delta$  definierten Akzeptors, wobei der Vorzustandsbegriff in der gleichen Weise verallgemeinert werden kann auf die Anwendung von Folgen und regulären Ausdrücken, wie dies vom Folgezustandsbegriff hinsichtlich der Funktion  $\delta$  bereits bekannt ist (vgl. Def. 2.2.1 ff. und Def. 3.1.1b).

Satz 3.1.3: Zu jedem Akzeptor  $\mathcal{A} = (X, Z, Z_0, \delta, Z')$ , der das reguläre Ereignis  $\rho$  akzeptiert, existiert ein zu  $\mathcal{A}$  inverser Akzeptor  $\overline{\mathcal{A}} = (X, Z, \overline{Z}_0 = Z', \delta^{-1}, \overline{Z}' = Z_0)$ , der die Reflexion  $\rho^{-1}$  von  $\rho$  akzeptiert.

Beweis: Der Akzeptor  $\mathcal{A}$  sei in Form eines Graphen beschrieben. Für jede von  $\mathcal{A}$  akzeptierte Folge  $s$  existiert ein (Vorwärts-) Pfad von einem Anfangs-Knoten  $z_1 \in Z_0$  zu einem End-Knoten  $z_j \in Z'$  von  $\mathcal{A}$ . Ein solcher Pfad ist definiert durch  $z_j = \delta(z_1, s)$ . Jeder dieser Pfade und kein anderer Pfad ist in Rückwärtsrichtung definiert von Knoten  $z_j$  nach Knoten  $z_1$  durch  $z_1 = \delta^{-1}(z_j, s^{-1})$ . Aufgrund von Definition 3.1.3 ist  $\rho^{-1}$  genau die Menge aller Reflexionen von Folgen  $s \in \rho$ , also die Reflexion des von  $\mathcal{A}$  akzeptierten Ereignisses  $\rho$ . Infolge Definition 2.2.4 ist jede Folge  $s^{-1} \in \rho^{-1}$  durch den Akzeptor  $\overline{\mathcal{A}}$  genau dann akzeptiert, wenn bei diesem Akzeptor  $\delta^{-1}$  die Übergangsfunktion,  $Z'$  (=  $\overline{Z}_0$ ) die Menge aller Anfangszustände, und  $Z_0$  (=  $\overline{Z}'$ ) die Menge aller Endzustände ist, q. e. d.

Der Graph eines Akzeptors  $\mathcal{A}$  wird in den Graphen des dazu inversen Akzeptors  $\overline{\mathcal{A}}$  umgewandelt, indem unter Beibehaltung



der Zustandszuordnung alle Zweige umgekehrt, alle Anfangsknoten in Endknoten und alle Endknoten in Anfangsknoten umgewandelt werden. Es folgt die Definition der zur Unterscheidbarkeit dualen Relation der *Rückwärts-Unterscheidbarkeit*.

Definition 3.1.5: Wenn ein Akzeptor  $\mathcal{A}$  den regulären Ausdruck  $\rho$  akzeptiert und für eine Vorsilbe  $s_{oi}$  von  $\rho$  ein Folgezustand  $z_i = \delta(z_o, s_{oi})$  existiert mit  $z_o \in Z_o$ , dann wird  $s_{oi}$  als *Vorsilbe des Zustandes*  $z_i$  bezeichnet.

Definition 3.1.6: Das *Vor-Ereignis*  $\rho_{oi}$  eines Zustandes  $z_i$  ist die Menge aller Vorsilben  $s$  des Zustandes  $z_i$ , d.h.

$$\rho_{oi} = \{s \mid [z_i = \delta(z_o, s)] \wedge z_o \in Z_o\}$$

Eine andere Ausdrucksweise dieser Definition lautet:

$$\rho_{oi} = \{s \mid \delta^{-1}(z_i, s^{-1}) \in Z_o\}$$

Definition 3.1.7: Zwei Zustände  $z_i$  und  $z_j$  eines Akzeptors  $\mathcal{A}$  sind *rückwärts-äquivalent*, wenn vor jeder beliebigen Folge  $s$  die Vorzustände  $z_{vi} = \delta^{-1}(z_i, s^{-1})$  und  $z_{vj} = \delta^{-1}(z_j, s^{-1})$  jeweils die gleiche Mitgliedschaftsrelation zur Menge  $Z_o$  der Anfangszustände haben.

Satz 3.1.4: Zwei Zustände  $z_i$  und  $z_j$  eines Akzeptors  $\mathcal{A}$  sind *rückwärts-äquivalent*, wenn deren Vor-Ereignisse  $\rho_{oi}$  und  $\rho_{oj}$  gleich sind.

Beweis: folgt aus Definition 3.1.6 und 3.1.7.

Die obenstehend vorgeschlagenen Definitionen und darauf basierenden Relationen gestatten es, Knoten in Zustandsgraphen hinsichtlich ihrer Rückwärts-Umgebung auf die gleiche Weise zueinander in Beziehung zu setzen, wie dies bezüglich der Vorwärts-Umgebung bekannt ist (u.a. /25,26/). Tafel 3.1.2

zeigt die Gegenüberstellung einiger hierbei dualer Eigenschaften und Relationen bei Akzeptoren. Hierdurch ergeben sich Methoden zur Reduktion indeterministischer Strukturen aufgrund folgenden Satzes.

Satz 3.1.5: Die Relation der Rückwärts-Äquivalenz ist bei Akzeptoren dual zur (Vorwärts-) Äquivalenz.

Beweis: Aus Definition 3.1.3 und 3.1.6 folgt

$$\rho_{oi}^{-1} = \{s^{-1} \mid s^{-1}(z_1, s^{-1}) \in Z_0\}.$$

Die Reflexion  $\rho_{oi}$  des Vor-Ereignisses des Zustandes  $z_1$  von  $\mathcal{A}$  wird akzeptiert durch den Zustand  $z_1$  des zu  $\mathcal{A}$  inversen Akzeptors  $\overline{\mathcal{A}}$  dadurch, daß hierbei  $\overline{z^1} = z_0$  ist, q.e.d.

Bekannt ist die Reduktion eines Akzeptors nach folgender Vorschrift /52/.

Regel 3.1.1: Eine Reduktion eines Akzeptors erfolgt durch die Verschmelzung äquivalenter Zustände.

Infolge Satz 3.1.5 ergibt sich die folgende hierzu duale Regel.

Regel 3.1.2: Eine Rückwärts-Reduktion eines Akzeptors erfolgt durch die Verschmelzung rückwärts-äquivalenter Zustände.

Im Zusammenhang mit der Dualität der Äquivalenz-Relation treten die folgenden dualen Begriffs-Paare auf:

Vorzustand	-	Folgezustand
rückwärts-äquivalent	-	äquivalent
rückwärts-unterscheidbar	-	unterscheidbar
ankommendes Bündel	-	abgehendes Bündel
$\rho^{-1}$	-	$\rho$
Anfangszustand	-	Endzustand

(Der Begriff *Bündel* wird im folgenden Abschnitt definiert.)  
Tafel 3.1.2 zeigt einige duale Eigenschaften von Zuständen bei Akzeptoren. In Kapitel 5 ff. wird gezeigt werden, daß bei Computer-Simulation von Transformationen an Zustandsgraphen die Dualität der Äquivalenz-Relation eine erhebliche Vereinfachung des Programmier-Aufwandes gestattet.

### 3.1.1 Schnittstellen in Zustandsgraphen

Zur formalen Beschreibung von Schnittstellen in Zustandsgraphen wird der Begriff *Bündel eines Knotens* eingeführt. Es werden Äquivalenz-Relationen zwischen Bündeln definiert, welche die theoretische Grundlage für einige computergerechte Elementar-Operationen zur Simulation von Transformationen an Zustandsgraphen bilden. In Kapitel 5 ff. wird die Anwendung dieser Elementar-Operationen zur Simulation der graphischen Synthese von Akzeptoren behandelt. Zunächst seien die folgenden Definitionen vereinbart.

Definition 3.1.8: Als *Gesamtzeile eines Zustandes*  $z_j$  wird bezeichnet:

- a) beim Akzeptor die Zeile  $j$  der Übergangstabelle oder die nach  $i$  geordnete Menge  $\{\delta(z_j, x_i) \mid \delta(z_j, x_i) \neq z_w\}$ ,
  - b) beim Moore-Automaten die Zusammenfassung der Zeile  $j$  der Übergangstabelle mit den Ausgabesymbolen der Folgezustände oder die nach  $i$  geordnete Menge  $\{(\delta(z_j, x_i), \lambda(\delta(z_j, x_i))) \mid \delta(z_j, x_i) \neq z_w\}$ ,
  - c) beim Mealy-Automaten die Zusammenfassung der Zeilen  $j$  aus Übergangstabelle und Ausgangstabelle oder die nach  $i$  geordnete Menge  $\{(\delta(z_j, x_i), \lambda(z_j, x_i)) \mid \delta(z_j, x_i) \neq z_w\}$ ,
- wobei  $z_w \in Z$  der Rückweisungszustand gemäß Satz 2.2.2 ist.

Definition 3.1.9: Zwei Gesamtzeilen  $z_i$  und  $z_j$  werden *angleichbar* genannt, wenn sie durch Vertauschung aller  $z_i$  mit  $z_j$  (oder umgekehrt) in den Zeilen der Übergangstabelle gleich werden.

Satz 3.1.6: Zwei Zustände  $z_i$  und  $z_j$  eines Automaten sind äquivalent, wenn deren Gesamtzeilen gleich oder angleichbar sind. Auf diese Weise äquivalente Zustände heißen *einfach-äquivalent*.

(Der Begriff *Bündel* wird im folgenden Abschnitt definiert.)  
Tafel 3.1.2 zeigt einige duale Eigenschaften von Zuständen bei Akzeptoren. In Kapitel 5 ff. wird gezeigt werden, daß bei Computer-Simulation von Transformationen an Zustandsgraphen die Dualität der Äquivalenz-Relation eine erhebliche Vereinfachung des Programmier-Aufwandes gestattet.

### 3.1.1 Schnittstellen in Zustandsgraphen

Zur formalen Beschreibung von Schnittstellen in Zustandsgraphen wird der Begriff *Bündel eines Knotens* eingeführt. Es werden Äquivalenz-Relationen zwischen Bündeln definiert, welche die theoretische Grundlage für einige computergerechte Elementar-Operationen zur Simulation von Transformationen an Zustandsgraphen bilden. In Kapitel 5 ff. wird die Anwendung dieser Elementar-Operationen zur Simulation der graphischen Synthese von Akzeptoren behandelt. Zunächst seien die folgenden Definitionen vereinbart.

Definition 3.1.8: Als *Gesamtzeile eines Zustandes  $z_j$*  wird bezeichnet:

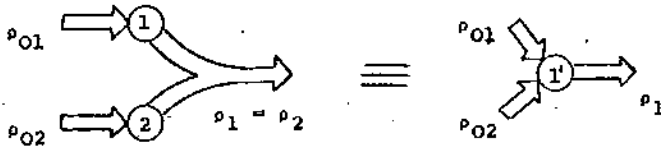
- a) beim Akzeptor die Zeile  $j$  der Übergangstabelle oder die nach  $i$  geordnete Menge  $\{\delta(z_j, x_1) \mid \delta(z_j, x_1) \neq z_w\}$ ,
  - b) beim Moore-Automaten die Zusammenfassung der Zeile  $j$  der Übergangstabelle mit den Ausgabesymbolen der Folgezustände oder die nach  $i$  geordnete Menge  $\{(\delta(z_j, x_1), \lambda(\delta(z_j, x_1))) \mid \delta(z_j, x_1) \neq z_w\}$ ,
  - c) beim Mealy-Automaten die Zusammenfassung der Zeilen  $j$  aus Übergangstabelle und Ausgangstabelle oder die nach  $i$  geordnete Menge  $\{(\delta(z_j, x_1), \lambda(z_j, x_1)) \mid \delta(z_j, x_1) \neq z_w\}$ ,
- wobei  $z_w \notin Z$  der Rückweisungszustand gemäß Satz 2.2.2 ist.

Definition 3.1.9: Zwei Gesamtzeilen  $z_i$  und  $z_j$  werden *angleichbar* genannt, wenn sie durch Vertauschung aller  $z_i$  mit  $z_j$  (oder umgekehrt) in den Zeilen der Übergangstabelle gleich werden.

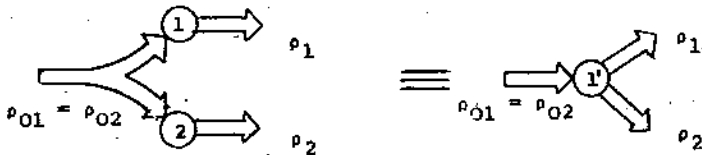
Satz 3.1.6: Zwei Zustände  $z_i$  und  $z_j$  eines Automaten sind äquivalent, wenn deren Gesamtzeilen gleich oder angleichbar sind. Auf diese Weise äquivalente Zustände heißen *einfach-äquivalent*.

definiert durch $\delta^{-1}$	definiert durch $\delta$
Alle Zustände sind paarweise rückwärts-unterscheidbar + Akzeptor ist rückwärts-minimal ( $\delta$ -Abbildung ist eindeutig)	Alle Zustände sind paarweise unterscheidbar + Akzeptor ist minimal ( $\delta^{-1}$ -Abbildung ist eindeutig)
Vorzustand $z_v$ eines Zustandes $z_f$ : $z_v = \delta^{-1}(z_f, s^{-1})$	Folgestand $z_f$ eines Zustandes $z_v$ : $z_f = \delta(z_v, s)$
Vor-Ereignis $\rho_{0i}$ eines Zustandes $z_i$ : $\rho_{0i} = \{s \mid \delta^{-1}(z_i, \rho_{0i}^{-1}) \in Z_0\}$	Durch Zustand $z_i$ akzeptiertes Ereignis $\rho_i'$ : $\rho_i' = \{s \mid \delta(z_i, \rho_i') \in Z^1\}$
Rückwärts-Reduktion möglich durch: Verschmelzung rückwärts-äquivalenter Zustände	Reduktion möglich durch: Verschmelzung äquivalenter Zustände

**Tafel 3.1.2:** Zur Demonstration der Dualität der Eigenschaften unterscheidbar und rückwärts-unterscheidbar bei Zuständen von Akzeptoren



a) graphische Darstellung von Regel 3.1.1



b) graphische Darstellung von Regel 3.1.2

**Bild 3.2.1:** Graphische Transformationen zur Verschmelzung äquivalenter oder rückwärts-äquivalenter Zustände

Zustände, die nicht einfach-äquivalent sind, heißen *einfach-unterscheidbar* /26/.

Auf diesem Satz beruht das Reduktionsverfahren durch Verschmelzen einfach-äquivalenter Zustände /26/. Gemäß Satz 3.1.5 ist dieser Satz auch auf rückwärts-einfach-äquivalente Zustände anwendbar, wovon Gebrauch gemacht wird, bei der in Kapitel 5 ff. beschriebenen Simulationstechnik.

Zur Vereinfachung der verbalen und graphischen Formulierung von Beziehungen in Zustandsgraphen werden die folgenden Definitionen eingeführt.

Definition 3.1.10: Ein *ankommendes (abgehendes) Bündel eines Knoten* ist eine Zusammenfassung ankommender (abgehender) Zweige dieses Knotens. Dieser Knoten wird *Bezugsknoten* des Bündels genannt. Die graphische Darstellung erfolgt nach Bild 3.1.3.

Definition 3.1.11: Zwei ankommende (abgehende) Gesamtbündel werden *gleich* genannt, wenn die Gesamtzeilen ihrer Bezugszustände in der  $\delta^{-1}$ -Funktion ( $\delta$ -Funktion) gleich sind.

Definition 3.1.12: Zwei ankommende (abgehende) Gesamtbündel werden *angleichbar* genannt, wenn die Gesamtzeilen ihrer Bezugszustände in der  $\delta^{-1}$ -Funktion ( $\delta$ -Funktion) angleichbar sind.

Definition 3.1.13: Zwei ankommende (abgehende) Gesamtbündel, die gleich oder angleichbar sind, werden als *(rückwärts) einfach-äquivalent* bezeichnet. Das Bündelsymbol für einfach-äquivalente Gesamtbündel wird gemäß Bild 3.1.4 vereinbart.

Satz 3.1.7: Die Bezugszustände zweier (rückwärts-) einfach-äquivalenter Bündel sind (rückwärts-) einfach-äquivalent. Der Beweis ergibt sich aus der Dualität der Äquivalenz-Relation und aus Satz 3.1.6.

Aus obigem Satz folgt, daß die (Rückwärts-) Einfach-Äquivalenz von Zuständen durch Vergleich der Bündel ihrer Knoten ermittelt werden kann. Hieraus können die beiden folgenden Regeln abgeleitet werden, die als Elementar-Operationen für die Transformation von Zustandsgraphen wichtig sind.

Regel 3.1.3: Eine Reduktion eines Zustandsgraphen erfolgt durch Verschmelzung von Knoten, deren abgehende Gesamtbündel einfach-äquivalent sind (Bild 3.2.1 a).

Regel 3.1.4: Eine Rückwärts-Reduktion eines Zustandsgraphen erfolgt durch Verschmelzung von Knoten, deren ankommende Gesamtbündel rückwärts-einfach-äquivalent sind (Bild 3.2.1 b).

Bild 3.2.1 zeigt die graphische Darstellung der beiden obigen Regeln unter Verwendung des Bündelsymbol. In Kapitel 5 ff. wird eine als *Suchlistendarstellung* bezeichnete Listenstruktur zur formalen Darstellung von gerichteten Graphen eingeführt werden, wobei jedes Gesamtbündel des Graphen durch einen als *Zeilenblock* bezeichneten Listenbestandteil beschrieben wird. Die Regeln 3.1.3/4 lassen sich daher unmittelbar auf diese Listenstruktur anwenden.

Mit Hilfe des eingeführten Bündelsymbol werden bei künftigen graphischen Darstellungen von Unterstrukturen die Schnittstellen charakterisiert. Hierbei wird jeder Schnittstellenknoten durch Bündelsymbole gekennzeichnet. Andernfalls handelt es sich um interne Knoten der jeweiligen Unterstruktur. Bild 3.1.5 veranschaulicht die Anwendung der eingeführten Bündelsymbole für Schnittstellen durch ein Beispiel.

### 3.2 Ein Satz graphischer Substitutionsregeln

Unter der Voraussetzung der in Kapitel 3.1 eingeführten Definitionen und der darauf basierenden Relationen wird in diesem Kapitel ein Satz von Regeln zur graphischen Synthese von Akzeptoren aus regulären Ausdrücken vorgeschlagen. Durch Erweiterung des Vorrates an bekannten graphischen Substitutionsregeln und z. T. verallgemeinerte Formulierung solcher Regeln wird der folgende Satz von Regeln zusammengestellt.

1. zur allgemeinen Transformation graphischer Strukturen  
(Kap. 3.2)
  - a) graphische Anwendung des Dekompositionsprinzips  
(OTT/FEINSTEIN, Bild 3.2.2),
  - b) Verallgemeinerung der Transformationen nach  
BRZOZOWSKI/McCLUSKEY jr. für Strukturen mit mehr  
als 2 Variablen (Bild 3.2.1),
  - c) Transformationen nach OTT/FEINSTEIN zur Entfer-  
nung von Leerzweigen durch Knotenverschmelzung  
(Bild 3.2.5),
  - d) eine neue Transformation zur Entfernung von Leer-  
zweigen bei nicht verschmelzbaren Knoten (Bild  
3.2.6);
  - e) eine neue Transformation zur Determinisierung  
zweiter Art (Bild 3.1.6, sowie Kapitel 3.2);
2. zur Transformation von Strukturen mit Iterationen  
(Kap. 3.2)
  - a) Transformation zur Bildung von Schlingen  
(BRZOZOWSKI/McCLUSKEY jr., Bild 3.2.9 a),
  - b) neu formulierte Transformationen zur Umwandlung  
von Schlingen in Schleifen, sowie zur Reduktion  
von Schleifen (Bild 3.2.10),
  - c) Transformation von Strukturen mit rekursiver  
Iteration (Bild 3.2.11);
3. zur Transformation von Strukturen mit Potenzen  
(Kap. 3.3)

Unter Verwendung des in Kapitel 3.1 definierten Bündelsym-  
boles (vgl. Definitionen 3.1.8/9) lassen sich die Regeln  
3.1.1 und 3.1.2 graphisch darstellen, wie in Bild 3.2.1 ge-  
zeigt wird. Auch zur vereinfachten Darstellung eines Tei-  
les der von BRZOZOWSKI/McCLUSKEY jr. angegebenen graphi-  
schen Äquivalenzen eignet sich das Bündelsymbol.



### 3.2.1 Strukturen ohne Iteration

Zur Ermittlung von Zustandsgraphen aus regulären Ausdrücken sind die folgenden graphischen Transformationsregeln bekannt für den Fall, daß keine Iteration vorhanden ist und somit keine Leerzweige entstehen. Von OTT/FEINSTEIN wurden angegeben (Bild 3.2.2):

- 1) Multiplikative Dekomposition
- 2) Disjunktive Dekomposition

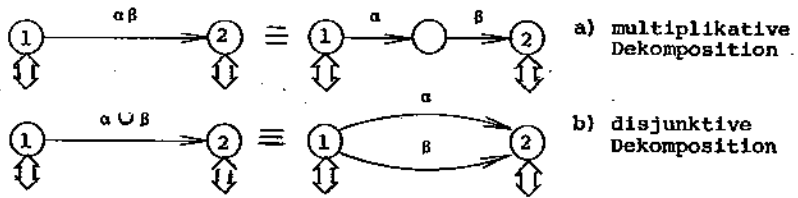
Bild 3.2.3 zeigt die von BRZOWSKI/McCLUSKEY jr. angegebenen Transformationsregeln für Strukturen mit drei Variablen:

- 3) Regel für Pfad-Vereinigung (Vereinigungsregel)
- 4) Regel für Pfad-Verzweigung (Verzweigungsregel)
- 5) Maschen-Stern-Umwandlung (Maschenregel)

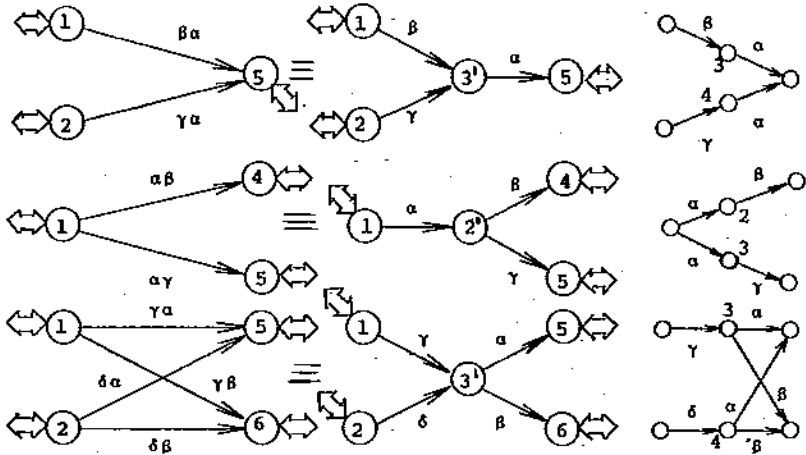
Die drei letzteren Regeln können ersetzt werden durch die Anwendung zweier anderer Regeln. Die in Bild 3.2.3 a und b auf der linken Seite gezeigten Strukturen werden durch die multiplikative Dekomposition zunächst transformiert. Das Zwischenergebnis wird auf der rechten Seite von Bild 3.2.3 a und b gezeigt. Der zweite Schritt ist eine (Rückwärts-) Einfach-Reduktion gemäß Bild 3.2.4 a bzw. b. Das Ergebnis dieser beiden Schritte ist identisch mit dem Ergebnis der Anwendung der Vereinigungs- oder Verzweigungsregel. Somit ist gezeigt, daß die Vereinigungs- und die Verzweigungsregel auf die elementaren Operationen der Dekomposition und der Reduktion zurückgeführt werden können.

Dies gilt auch für die Maschenregel nach Bild 3.2.3 c. Eine zweimalige Anwendung der Verzweigungsregel ergibt die in Bild 3.2.3 auf der rechten Seite gezeigte Struktur. Durch Reduktion gemäß Bild 3.2.4 c folgt daraus die in Bild 3.2.3 in der Mitte gezeigte Stern-Struktur. Somit ist erwiesen, daß die drei folgenden elementaren Regeln für die Transformation beliebiger Zustandsgraphen ohne Iteration und Leerzweige genügen.

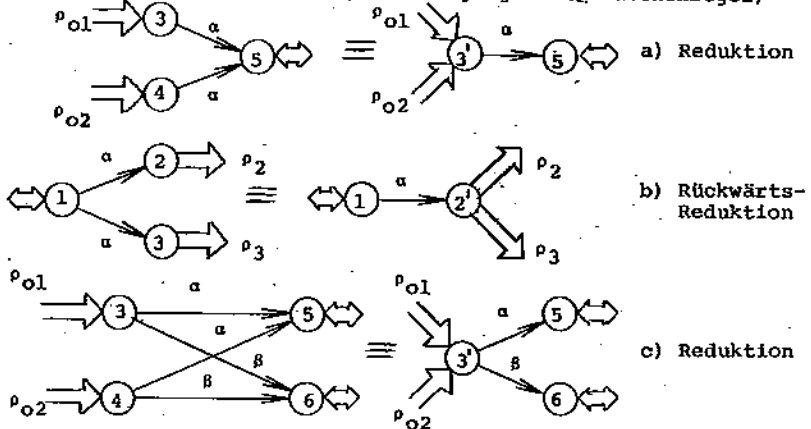
- 1) Multiplikative Dekomposition
- 2) Disjunktive Dekomposition
- 3) Einfach-Reduktion (vorwärts oder rückwärts)



**Bild 3.2.2** Transformationen ohne Iteration n.OTT/FEINSTEIN.



**Bild 3.2.3** Transformationen ohne Iteration nach BRZOZOWSKI/McC. (Vereinigungs-, Verzweigungs- und Maschenregel)



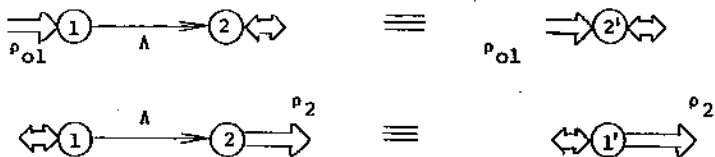
**Bild 3.2.4** Reduzierende Transformationen als Ersatz für Vereinigungs-, Verzweigungs- und Maschenregel.

### 3.2.2 Entfernung von Leerzweigen

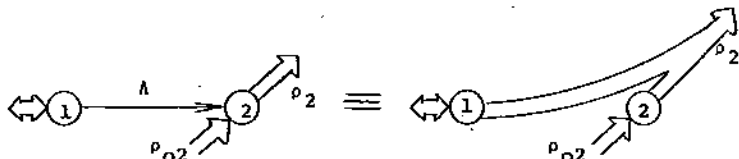
Bei der graphischen Synthese von Akzeptoren aus regulären Ausdrücken mit Iterationen werden Leerzweige eingeführt. Das Syntheseproblem ist jedoch erst dann gelöst, wenn ein echter Zustandsgraph, d.h. ein Graph ohne Leerzweige vorliegt. Von OTT/FEINSTEIN wurde für solche Fälle eine Regel zur graphischen Leerzweig-Entfernung angegeben, in denen dies durch Verschmelzen der beiden Knoten möglich ist, welche der betreffende Leerzweig verbindet. Diese als "verschmelzende Leerzweigentfernung" bezeichnete Regel ist in Bild 3.2.5 dargestellt.

Neben dieser Regel ist keine Vorschrift bekannt zur Entfernung von Leerzweigen zwischen solchen Knoten, die durch die Vorschrift nach Bild 3.2.5 nicht verschmolzen werden können. Für diesen Fall wird eine Transformation vorgeschlagen (Bild 3.2.6), welche die Leerzweig-Entfernung graphisch ohne Verschmelzung von Knoten ermöglicht. Diese Transformation gestattet den Verzicht auf die hierzu bei bekannten Syntheseverfahren notwendige Aufstellung von Tabellen, wodurch bessere Homogenität erreicht wird. Der Beweis für die in Bild 3.2.6 gezeigte Transformation wird mit Hilfe von Bild 3.2.7 wie folgt erbracht.

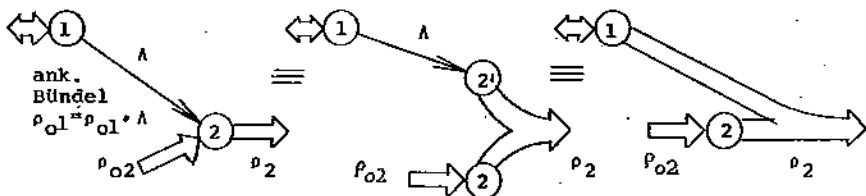
Beweis: Der Leerzweig in Bild 3.2.7 a ist ein ankommendes Bündel des Knoten 2. Durch inverse Anwendung der Transformation nach Bild 3.2.1 a wird Knoten 2 aufgespalten in die Knoten 2 und 2' (Bild 3.2.7 b). Knoten 2' wird entfernt durch Verschmelzung mit Knoten 1 zu Knoten 1 nach Transformation in Bild 3.2.5, woraus Bild 3.2.7 c folgt, q.e.d.



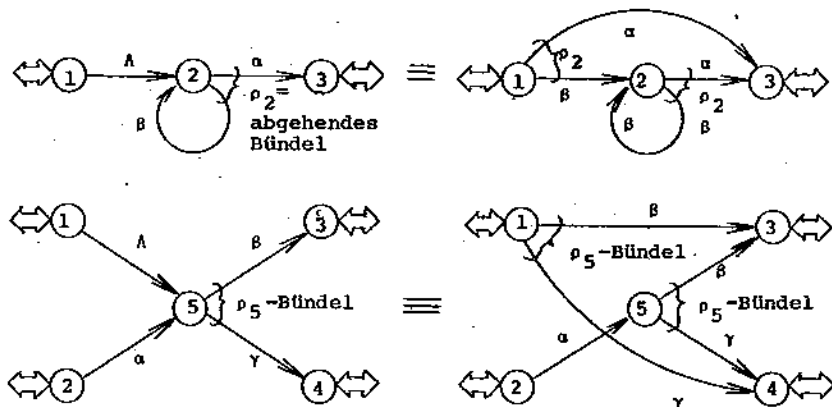
**Bild 3.2.5** Transformationen zur Leerzweig-Entfernung durch Knoten-Verschmelzung nach OTT/FEINSTEIN.



**Bild 3.2.6** Transformation zur Leerzweig-Entfernung bei nicht-verschmelzbaren Knoten.



**Bild 3.2.7** Zum Beweis der Transformation nach Bild 3.2.6.



**Bild 3.2.8** Beispiele für verschmelzungsfreie Entfernung von Leerzweigen.

Bild 2.3.8 zeigt zwei Beispiele für die Entfernung von Leerzweigen nach der hier bewiesenen Regel.

Bei allen bekannten Syntheseverfahren auf der Grundlage regulärer Ausdrücke wird zumindest das Vorliegen eines Moore-Automaten vorausgesetzt, weshalb auch das Ausgangssignal ("1") der Endzustände mitbearbeitet werden muß. Auch der Anfangsknoten erfährt bei diesen Methoden irgendeine Sonderbehandlung. Da bei der hier vorliegenden Arbeit jedoch vom Akzeptor-Modell ausgegangen wird, ist im Zusammenhang mit solchen Leerzweigen, die mit Anfangs- oder Endknoten verbunden sind, die folgende Regel zu beachten.

Regel 3.2.1: Ein Leerzweig eines Akzeptors überträgt die Endzustandseigenschaft von seinem Folgezustandsknoten  $z_f \in Z'$  auf seinen Vorzustandsknoten  $z_v$  und die Anfangszustandseigenschaft von seinem Vorzustandsknoten  $z_v \in Z_0$  auf seinen Folgezustandsknoten  $z_f$ .

Dies bedeutet:

1) bei verschmelzender Leerzweig-Entfernung:

Ist einer der beiden durch einen Leerzweig verbundenen Knoten ein Endzustands- (Anfangszustands-) Knoten, so ist der durch Verschmelzung entstehende neue Knoten ein Endzustands- (Anfangszustands-) Knoten.

2) bei verschmelzungsfreier Leerzweig-Entfernung:

Ist an einem Leerzweig der Folgezustandsknoten ein Endknoten (der Vorzustandsknoten ein Anfangszustandsknoten), so wird der Vorzustandsknoten in einen Endknoten (der Folgezustandsknoten in einen Anfangsknoten) umgewandelt. Bei entgegengesetzter Kombination

findet keine solche Umwandlung statt.

Beweis: Ist  $\rho_{OV}$  das Vorereignis von  $z_V$  und  $\rho_{OF}$  das Vorereignis von  $z_F$ , und  $\rho$  das vom vorliegenden Akzeptor akzeptierte Gesamt ereignis, so ist bei Fall 1)

$\rho_{OV} = \rho_{OV} \cdot A = \rho_{OF}$ . Wegen  $z_F \in Z'$  ist  $\rho_{OF} \subseteq \rho$ , und somit ist  $\rho_{OV} \subseteq \rho$  und  $z_V$  ein Endzustandsknoten, q.e.d. Der Beweis für den Fall mit Anfangsknoten ist hierzu dual.

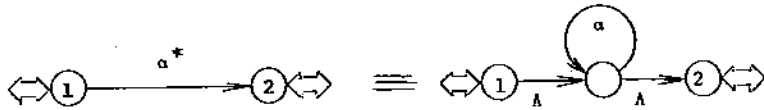
Der Beweis gilt auch für den Fall 2).

Der Beweis für das Verbot einer Knotenumwandlung bei entgegengesetzter Kombination mit  $z_V \in Z'$  und  $z_F \notin Z'$  ist folgender. Da im Fall 2) der Leerzweig nicht der einzige ankommende Zweig von Knoten  $z_F$  ist, gilt  $\rho_{OF} \neq \rho_{OV}$  und somit  $\rho_{OF} \not\subseteq \rho_{OV}$ . Daraus folgt nicht notwendigerweise  $\rho_{OF} \subseteq \rho$ . (Eine evtl. in Sonderfällen gegebene Relation  $\rho_{OF} \subseteq \rho$  kann nur von der Vorwärts-Umgebung des Knoten  $z_F$  herrühren.)

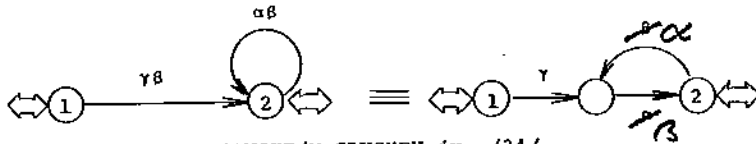
Der Beweis für die Struktur mit Anfangszustand ist hierzu dual.

### 3.2.3 Strukturen mit Iterationen

Im Folgenden soll die bei den bisherigen Betrachtungen ausgeklammerte Iteration behandelt werden. Bild 3.2.9 zeigt die beiden bekannten graphischen Transformationen für diesen Fall. Zu diesem Satz von Regeln wird mit dem Ziel einer Verallgemeinerung und der Erreichung eines Reduktions-Effektes folgende Modifikation vorgeschlagen. Die Regel nach Bild 3.2.9 b wird ersetzt durch die vier Regeln nach Bild 3.2.10. Die Richtigkeit dieser vier Transformationen wird

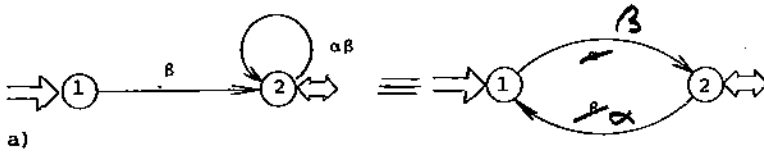


a) nach OTT/FEINSTEIN /46/, (Iterations-Dekomposition)

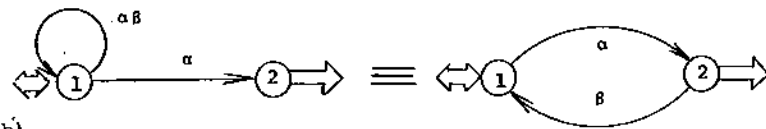


b) invers zu BRZOZOWSKI/MCCLUSKEY jr. /34/

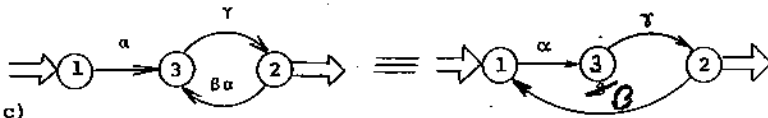
Bild 3.2.9 Transformationen für Iterationen



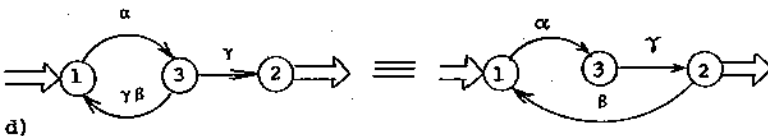
a)



b)



c)



d)

Bild 3.2.10 Transformationen zur Reduktion deterministischer und nicht-deterministischer Schleifen und Schlingen.

anschließend bewiesen. Hierbei ist die Regel nach Bild 3.2.10 a eine andere Schreibweise für die Regel nach Bild 3.2.9 b (BRZOZOWSKI/McCLUSKEY jr.) zur Umwandlung von Schlingen in Schleifen. Bild 3.2.10 b zeigt die hierzu duale neue Regel für indeterministische Strukturen. Die ebenfalls neuen Regeln in Bild 3.2.10 c und d haben einen reinen Reduktionseffekt bei Schleifen-Strukturen. Es folgen die Beweise für die Regeln nach Bild 3.2.10.

Beweis zu Bild 3.2.10 a: Man setze  $a = Aa$  und führe Transformation nach Bild 3.2.9 durch. Nach Entfernung von A (Regel nach Bild 3.2.5 a) ergibt sich die vorgeschlagene Lösung, q.e.d.

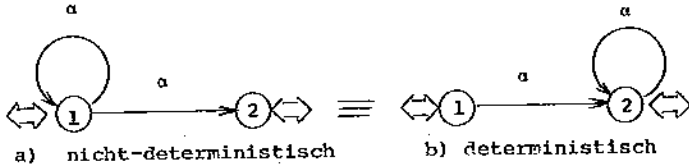
Beweis zu Bild 3.2.10 b: Man zerlege die Kette  $a\beta$  der Schlinge des Knotens 1. Der dadurch entstandene Knoten 3 ist nicht-rückwärts-unterscheidbar von Knoten 2, weshalb diese beiden Knoten verschmolzen werden zu Knoten 2 (Regel nach Bild 3.2.1 b), q.e.d.

Beweis zu Bild 3.2.10 c: Man zerlege die Kette  $\beta a$  zwischen Knoten 2 und 3 und erhält einen neuen Knoten 4. Knoten 1 und 4 sind nicht unterscheidbar und werden deshalb verschmolzen zu Knoten 1 (Regel 3.2.1 a), q.e.d.

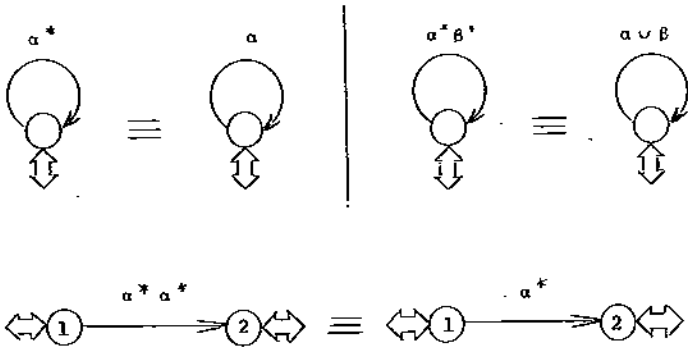
Beweis zu Bild 3.2.10 d: Man zerlege die Kette  $\gamma\beta$  zwischen Knoten 3 und 1 und erhält einen neuen Knoten 4. Knoten 2 und 4 sind nicht-rückwärts-unterscheidbar und werden daher verschmolzen zu Knoten 2 (Regel nach Bild 3.2.1 b), q.e.d.

Eine spezielle und wichtige iterative Struktur und ihre Transformation zeigt Bild 3.2.11. Diese Transformation ist eine Auswirkung der folgenden Gleichung der KLEENE'schen Algebra.

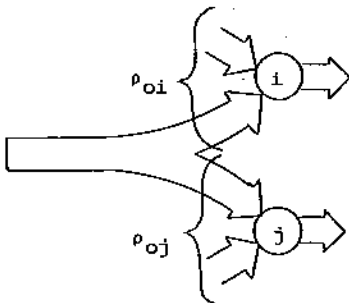




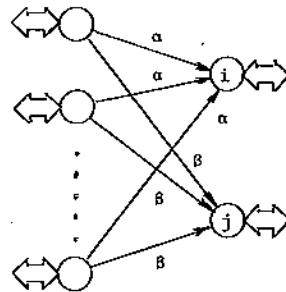
**Bild 3.2.11** Zwei äquivalente Realisierungen des Ereignis  $\alpha^* \alpha = \alpha \alpha^*$ .



**Bild 3.2.13** Transformationen für rekursive Iterationen.



**Bild 3.2.14** rückwärts teil-  
unterscheidbare Bündel  $\rho_{oi}$   
und  $\rho_{oj}$  zweier Knoten i und j



$$l(\alpha) = l(\beta) = 1$$

**Bild 3.2.15** indeterministische  
Struktur zweiter Art

Regel 3.2.2:

$$\alpha \alpha^* = \alpha^* \alpha$$

Beweis:

$$\begin{aligned}
\alpha \alpha^* &= \alpha (\Lambda \cup \alpha \cup \alpha^1 \cup \alpha^2 \cup \dots) \\
&= \alpha \Lambda \cup \alpha^2 \cup \alpha^3 \cup \dots \\
\text{wegen } \Lambda \alpha &= \alpha \Lambda: \\
&= \Lambda \alpha \cup \alpha^2 \cup \alpha^3 \cup \dots \\
&= (\Lambda \cup \alpha \cup \alpha^1 \cup \alpha^2 \cup \dots) \alpha \\
&= \alpha^* \alpha
\end{aligned}$$

q.e.d.

Für die Auflösung von rekursiv-iterativen Strukturen werden zusätzliche Transformationen vorgeschlagen, die in Bild 3.2.13 gezeigt und durch die Regeln 2.1.3, 2.1.5 und 2.1.6 bewiesen sind.

### 3.2.4 Indeterministische Strukturen zweiter Art

Wie bereits erwähnt wurde, gibt es indeterministische Strukturen, die nicht durch Rückwärts-Minimisierung deterministisch werden können (vgl. Regel 3.1.2). Ein Beispiel für eine solche indeterministische Struktur zweiter Art zeigt Bild 3.2.b. Im Rahmen einer Synthese-Aufgabe besteht das Ziel darin, solche Strukturen und auch indeterministische Strukturen erster Art in deterministische Strukturen umzuwandeln. Hierzu wird untenstehender Algorithmus zur Ergänzung des graphischen Instrumentariums angegeben. Zur leichteren Formulierung des Problemes und des vorgeschlagenen Lösungsweges nebst Beweisen werden folgende Definitionen eingeführt.

Definition 3.2.1: Zwei Bündel  $\rho_{oi}$ " und  $\rho_{oj}$ " sind *rückwärts-vollunterscheidbar*, wenn sie keine rückwärts-äquivalenten Unterbündel enthalten.

Definition 3.2.2: Zwei Zustände  $z_i$  und  $z_j$  sind *rückwärts-teilunterscheidbar*, wenn deren ankommende Gesamtbündel  $\rho_{oi}$  und  $\rho_{oj}$  rückwärts-unterscheidbar sind und sich zerlegen lassen in jeweils zwei zueinander exklusive Unterbündel  $\rho_{oi}'$  und  $\rho_{oi}''$  bzw.  $\rho_{oj}'$  und  $\rho_{oj}''$  derart, daß  $\rho_{oi}'$  und  $\rho_{oj}'$  rückwärts-voll-

unterscheidbar, sowie  $\rho_{oi}$  und  $\rho_{oj}$  rückwärts-äquivalent sind (Bild 3.2.14).

**Definition 3.2.4:** Zwei rückwärts-teilunterscheidbare Bündel  $\rho_{oi}$  und  $\rho_{oj}$  sind eine indeterministische Struktur zweiter Art, wenn sie rückwärts-einfach-äquivalente Unterbündel enthalten (Bild 3.2.15 und beispielsweise Bild 3.2.b).

Zur Determinisierung nicht-deterministischer Strukturen zweiter Art wird der folgende Algorithmus angegeben, wobei in Bild 3.2.16a die Aufgabenstellung graphisch dargestellt ist. Knoten 1 und 2 sind hierbei Knoten mit rückwärts-einfach-äquivalenten Unterbündeln. Die ankommenden Gesamtbündel der Knoten 1 und 2 sind somit indeterministisch nach zweiter Art. Der vorgeschlagene Algorithmus besteht aus folgenden Schritten.

**1. Schritt:** Transformation der Struktur (Bild 3.2.16 a) in eine Struktur nach Bild 3.2.16 c.

**2. Schritt:** Man entferne die durch Schritt 1 eingeführten Leer-Zweige gemäß Regel nach Bild 3.2.5/6.

Zur Determinisierung eines Zustandsgraphen wende man diese Prozedur auf jede nicht-deterministische Unterstruktur zweiter Art, und Regel 3.1.2 auf jede nicht-deterministische Struktur erster Art an.

**Beweis:** Schritt 1 a (Bild 3.2.16 a/b) erfüllt Axiom 8 b nach Tafel 2.1.2. Schritt 1 b (Bild 3.2.1 b) erfüllt Regel 3.1.2. Die atomaren Übergänge von den ehemaligen Vorzustandsknoten der Knoten 1 und 2 zu Knoten 3 sind nunmehr determi-

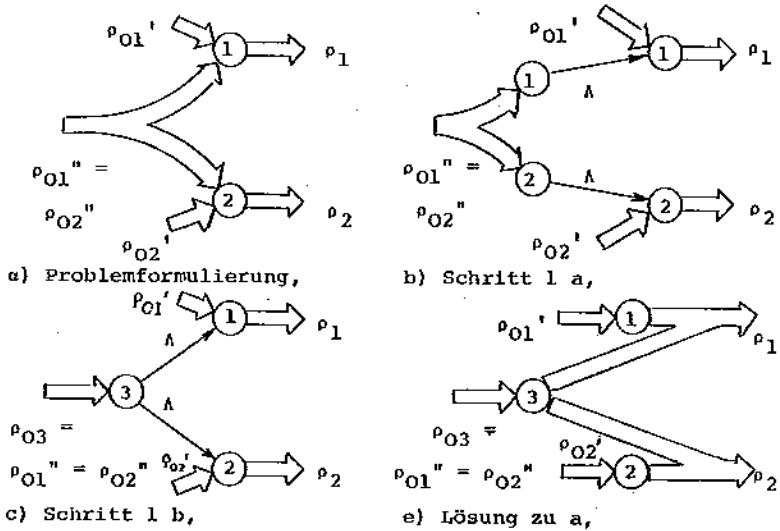


Bild 3.2.16 Algorithmus zur Determinisierung von indeterministischen Strukturen zweiter Art

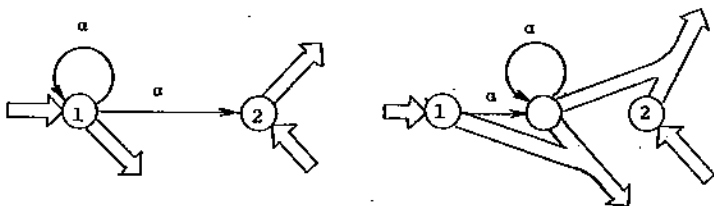


Bild 3.2.17 Beispiel für die Determinisierung einer indeterministischen Struktur zweiter Art.

nistisch. Die Regel nach Bild 3.2.6 ist anwendbar, da die vorausgesetzte Aufgabenstellung vorliegt. q.e.d.

Konvergenz-Beweis für den nicht-iterativen Fall: Wenn die obige Prozedur auf die Bündel der Knoten 1 und 2 angewandt wurde, sind die Übergänge von allen Vorzuständen  $z_v = \delta^{-1}(z_1, x_1^{-1}) = \delta^{-1}(z_2, x_1^{-1})$  deterministisch geworden, während die Knoten 1 und 2 rückwärts-vollunterscheidbar geworden sind (Bild 3.2.1.3 e). Da hierbei Bündel  $\rho_3$  durch die Vereinigung zweier Bündel entstanden ist, können zu den Folgezuständen von  $z_1$  und  $z_2$  indeterministische Übergänge von  $z_3$  entstanden sein. Eine solche neuentstandene indeterministische Struktur würde sich rechts der Knoten 1 und 2 befinden, während die ursprüngliche vorhandene indeterministische Struktur sich links von Knoten 1 und 2 befunden hat. Es ist somit bei wiederholter Anwendung obiger Prozedur gesichert, daß sich die Orte der Anwendung im Graphen sukzessive nach rechts bewegen. Auf diese Weise werden in einer endlichen Anzahl von Schritten die Endzustände erreicht, q.e.d.

Konvergenzbeweis für den iterativen Fall: Das Problem ist für die Abgangsbündel von Schleifen oder Schlingen lösbar aufgrund der Transformationen nach Bild 3.2.10 b und d, 3.2.11, oder 3.3.2 a. Das Fortschreiten des Verfahrens von Knoten mit Eintritts-Bündel durch die Schleife oder Schlinge hindurch ist aufgrund des Beweises für den nicht-iterativen Fall gesichert, q.e.d.

Bild 3.2.17 und 4.1.3 zeigt Anwendungsbeispiele für den vorgeschlagenen Determinisierungs-Algorithmus.

### 3.3 Spezielle graphische Transformation von Strukturen mit Potenzen

In Syntheseproblemen ist oft implizit die Aufgabenstellung enthalten, Folgen oder Teilfolgen gleicher Symbole hinsicht-

lich ihrer Länge zu separieren. Solche Aufgaben liegen beispielsweise dann vor, wenn der als Problemformulierung gegebene reguläre Ausdruck Potenzen enthält (vgl. Schreibweise zu Def. 2.1.2). Strukturen, in denen Potenzen und Iterationen des gleichen Ausdruckes verkettet sind, bieten einen Ansatzpunkt für die Durchführung von Reduktionsprozessen. In diesem Kapitel werden hierzu typische graphische Transformationen eingeführt. Zunächst seien zur Ermöglichung einfacherer Formulierungen die folgenden Definitionen und Relationen eingeführt.

Definition 3.3.1: Die *beschränkte Iteration*  $a^{*n}$  von  $a$  ist definiert durch

$$a^{*n} = \bigcup_{i=0}^n a^i$$

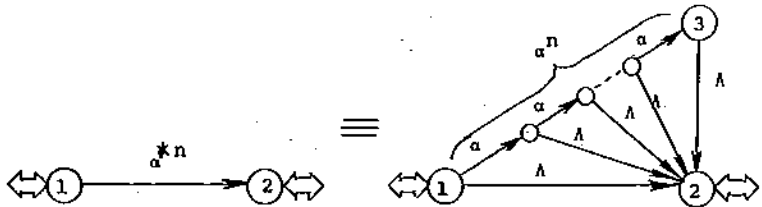
Bild 3.3.1 zeigt die graphische Darstellung einer solchen beschränkten Iteration unter Verwendung von Leerzweigen. Die Pfade zwischen Knoten 1 und 2 (*Fächer*) beschreiben die beschränkte Iteration  $a^{*n}$ , während der Pfad von Knoten 1 nach Knoten 3 (*Potenz-Kette*) die Potenz  $a^n$  beschreibt. Diese Tatsache ist die Grundlage der untenstehend begründeten *Auffächerung* der Iteration (Bild 3.3.3 b). Zwei wichtige Regeln zur Begründung der in diesem Kapitel eingeführten Transformationen lauten wie folgt.

$$a^* = A \vee a \ a^* \quad (\text{Regel 2.1.7})$$

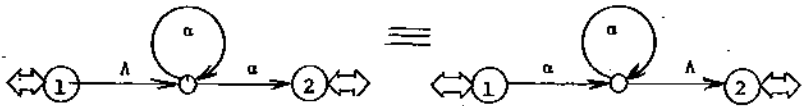
$$a^* \ a = a \ a^* \quad (\text{Regel 3.2.2})$$

Diese Regeln sind in Bild 3.3.2 graphisch dargestellt. Durch rekursive Anwendung dieser Regeln ergeben sich als Folgerung die Regeln 3.3.1 und 3.3.2.

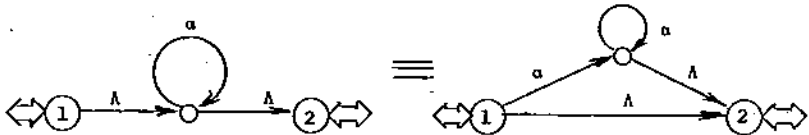
Regel 3.3.1:  $a^n \ a^* = a^* \ a^n$



**Bild 3.3.1** Graphen der beschränkten Iteration

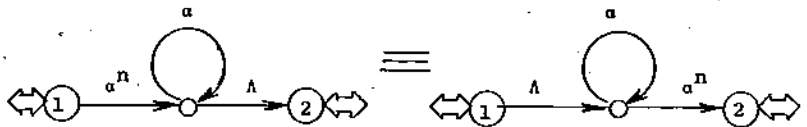


a) graphische Darstellung von Regel 3.2.2

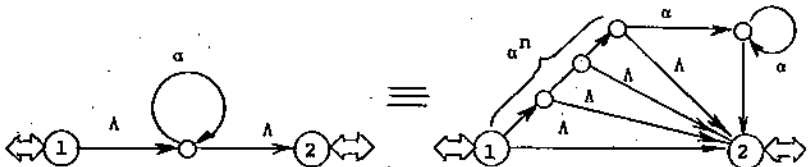


b) graphische Darstellung von Regel 2.1.7

**Bild 3.3.2** Zwei elementare Transformationen für Iterationen



a) graphische Darstellung von Regel 3.3.1



b) Auffächerung der Iteration (Regel 3.3.2)

**Bild 3.3.3** Verallgemeinerte elementare Transformationen der Iteration

Beweis:

aus Regel 3.2.1 folgt:

$$\begin{aligned}
 a^n a^* &= a^{n-1} a^* a \\
 a^{n-1} a^* a &= a^{n-2} a^* a^2 \\
 a^{n-2} a^* a^2 &= a^{n-3} a^* a^3 \\
 a^{n-i} a^* a^i &= a^{n-i-1} a^* a^{i+1} \\
 a^1 a^* a^{n-1} &= a^* a^n \\
 &= a^* a^n \quad \text{q.e.d.}
 \end{aligned}$$

Regel 3.3.2:

$$a^* = a^* \cup a^{n+1} a^*$$

Beweis:

aus Regel 2.1.7 folgt:

$$\begin{aligned}
 a^* &= A \cup a a^* \\
 a^* &= A \cup a (A \cup a a^*) \\
 &= A \cup a A \cup a^2 a^* \\
 a^* &= A \cup a A \cup a^2 (A \cup a a^*) \\
 &= A \cup a A \cup a^2 A \cup a^3 a^* \\
 a^* &= A \cup a A \cup a^2 \cup \dots \cup a^{n+1} a^* \\
 &= a^* \cup a^{n+1} a^* \quad \text{q.e.d.}
 \end{aligned}$$

Die graphische Darstellung dieser beiden obigen Regeln zeigt Bild 3.3.3. Bild 3.3.4 zeigt vier darauf basierende Transformationen zur Auflösung uneigentlicher Zustandsgraphen mit Potenzen, Iterationen und beschränkten Iterationen. Diese Transformationen können mit Hilfe der Regeln 3.3.1 und 3.3.2 leicht bewiesen werden.

In Kapitel 4.2.1 wird ein Beispiel für die Anwendung obiger Transformationsregeln gezeigt.



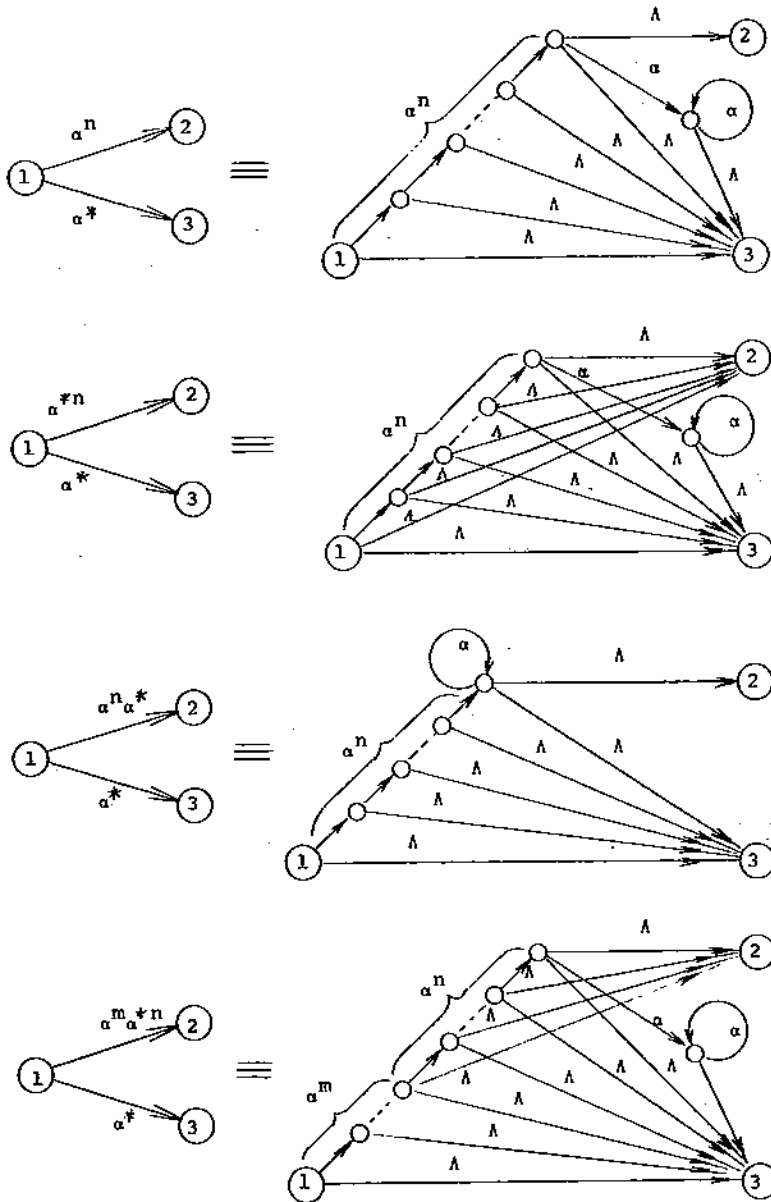


Bild 3.3.4 Misch-Transformationen für Potenzen und Iterationen