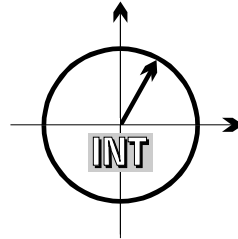


■ *Forschungsberichte aus dem  
Institut für Nachrichtentechnik  
der Universität Karlsruhe (T.H.)*



Karl Lütjen

■ **Systeme und Verfahren  
für strukturelle  
Musteranalysen mit  
Produktionsnetzen**

■ Band 5

**Forschungsberichte aus dem Institut für Nachrichtentechnik  
der Universität Karlsruhe (T.H.)**

---

Herausgeber: Prof. Dr. rer. nat. Friedrich Jondral

- Band 1 Marcel Kohl  
**Simulationsmodelle für die Bewertung von Satellitenübertra-  
gungstrecken im 20/30 GHz Bereich**
- Band 2 Christoph Delfs  
**Zeit-Frequenz-Signalanalyse: Lineare und quadratische Ver-  
fahren sowie vergleichende Untersuchungen zur Klassifikation  
von Klaviertönen**
- Band 3 Gunnar Wetzker  
**Maximum-Likelihood Akquisition von Direct Sequence  
Spread-Spectrum Signalen**
- Band 4 Anne Wiesler  
**Parametergesteuertes Software Radio für Mobilfunksysteme**
- Band 5 Karl Lütjen  
**Systeme und Verfahren für strukturelle Musteranalysen mit  
Produktionsnetzen**

## Vorwort des Herausgebers

Mustererkennungs- und -analyseaufgaben erscheinen einem unbedarften Beobachter häufig als recht einfach zu lösende Probleme. Dabei wird von Feuilletonisten angesehener Zeitungen häufig, manchmal aber auch von Wissenschaftlern, übersehen, dass die Beschreibung einer Fragestellung noch lange nicht deren Lösung ist und dass eine wissenschaftlich begründete Lösung sich nicht unbedingt als wirklich brauchbar erweisen muss. Andererseits besitzen rein "praktische Lösungen" in der Regel den Nachteil, dass sie nicht automatisch zur Bearbeitung verwandter oder gar nur ähnlich gelagerter Fragestellungen herangezogen werden können.

Die Mustererkennung beschäftigt sich sowohl mit einfachen (d. h. durch Klassifikationsverfahren in einem Schritt zuzuordnenden) Mustern als auch mit komplexen (d. h. aus einfachen Mustern aufgebauten) Objekten. Hierbei ist zu beachten, dass Mustererkennungsverfahren eigentlich präziser als Wiedererkennungsverfahren für bekannte Muster bezeichnet werden sollten, da wenigstens ansatzweise bekannt sein muss, was (wieder-) erkannt werden soll.

Die Ausgangsaufgabe der vorliegenden Dissertation ist die Analyse von aus einem Flugobjekt aufgenommenen Schwarz/Weiß-Bildern. Sie wird im Anhang A am Beispiel bildgestützter Navigation dargestellt: Das Flugobjekt "kennt" z. B. markante Objekte im Gelände aus vorliegendem Kartenmaterial und soll auf dieser Grundlage die Objekte während des Überflugs wiedererkennen, um aus Lageabweichungen Kurskorrekturen vornehmen zu können. Markante Objekte sind z. B. Straßenkreuzungen, Gebäudeanordnungen oder Ähnliches, also komplexe Muster, die in den zugehörigen Konturbildern aus einfachen Mustern wie Geraden, Streifen, Kreisbögen usw. aufgebaut sind. Die Kunst der Wiedererkennung liegt einerseits in der Vorverarbeitung der Bilder, in der Extraktion und Klassifikation der einfachen Muster und andererseits darin, aus diesen die komplexen Muster beispielsweise mit einem auf Produktionsnetzen beruhenden Verfahren zu rekonstruieren. Die Aufgabe der automatischen Bildanalyse besteht darin, festgelegte Referenzobjekte in aktuellen Aufnahmen wiederzufinden und zu positionieren.

Die ausgezeichnete Dissertation von Herrn Lütjen stellt die von ihm entwickelten bzw. verwendeten Bildanalysemethoden auf eine tragfähige theoretische Grundlage, so dass davon ausgegangen werden kann, dass die Analyseergebnisse robust, reproduzierbar und nachvollziehbar erscheinen. Sie ist ein Beispiel dafür, wie einer auf praktischen Erfahrungen eines Ingenieurs basierenden Arbeit ein theoretisch einwandfreies Fundament gegeben wird.

Karlsruhe im Dezember 2001

Friedrich Jondral

Copyright: Institut für Nachrichtentechnik  
Universität Karlsruhe, 2001

ISSN: 1433-3821

# **Systeme und Verfahren für strukturelle Musteranalysen mit Produktionsnetzen**

Zur Erlangung des akademischen Grades eines

**DOKTOR-INGENIEURS**

von der Fakultät für  
Elektrotechnik und Informationstechnik  
der Universität Fridericiana Karlsruhe

genehmigte

**DISSERTATION**

von

Dipl.-Ing. Karl Lütjen

aus

Oldenburg in Oldenburg

Tag der mündlichen Prüfung:  
Hauptreferent:  
Korreferenten:

27.11.2001  
Prof. Dr. rer. nat. F. Jondral  
Prof. Dr. Ing. C.-E. Liedtke  
Prof. Dr. rer. nat. M. Tacke



## Danksagung

Die vorliegende Arbeit entstand während meiner Tätigkeit als wissenschaftlicher Mitarbeiter am Forschungsinstitut für Informationsverarbeitung und Mustererkennung (FIM) in Ettlingen bzw. am Forschungsinstitut für Optronik und Mustererkennung (FOM), in dem die Abteilungen des FIM und des Forschungsinstitutes für Optik seit Juni 1999 zusammengefasst sind.

Ich danke den Direktoren des FIM, Herrn Prof. Dr. H. Kazmierczak und des FOM, Herrn Prof. Dr. M. Tacke für die Unterstützung der Arbeit und die hervorragenden Arbeitsmöglichkeiten. Herrn Prof. Dr. M. Tacke danke ich darüber hinaus für die Übernahme des zweiten Korreferates.

Herrn Prof. Dr. F. Jondral danke ich für die Betreuung als "Externer" an der Universität Karlsruhe und die Übernahme des Hauptreferates. Herrn Prof. Dr. C.-E. Liedtke danke ich für die Übernahme des ersten Korreferates.

Weiterhin gilt mein Dank all denjenigen, die auf vielfältige Weise zum Gelingen der Arbeit beigetragen haben: Meiner Frau Bärbel sowie Frau I. Dönges und Frau G. Potzy für redaktionelle Korrekturen, Herrn Dr. R. Geibel, Herrn Dr. U. Stilla, Herrn Dr. E. Michaelsen sowie Herrn Dr. E. Mauer für inhaltliche Diskussionen und Hinweise, Frau B. Euscher für die Anfertigung der Zeichnungen und die sorgfältigen Fotoarbeiten, Herrn Dipl.-Inform. H. Füger und Herrn Dipl.-Inform. H.-J. Greif für die Implementierung der Module des BPI-Programms zur Analyse der Navigationsbilder.

Schließlich gilt mein besonderer Dank Herrn Dipl.-Ing. K.-H. Bers und Herrn Dipl.-Inform. U. Thönnessen, die mir viele Arbeiten und Dienstreisen abnahmen; ohne sie hätte ich nicht die Zeit gehabt, die vorliegende Arbeit fertig zu stellen.





## Kurzfassung

Die vorliegende Arbeit befasst sich mit der automatischen Musteranalyse und bearbeitet einen strukturellen Ansatz mit Konzepten, die auf Transformations- bzw. Reduktionssystemen aus dem Bereich der theoretischen Informatik sowie Tafel- und Produktionssystemen aus dem Bereich der sog. künstlichen Intelligenz aufbauen.

Der Ansatz arbeitet mit Produktionsnetzen, die symbolische Modelle sind und u. a. Teil-von-Relationen beschreiben. Er erlaubt, mehrstufige Musteranalysen auf Parallelrechnersystemen effizient auszuführen. Die detaillierte formale Beschreibung dieses Ansatzes ist ein Schwerpunkt der vorliegenden Arbeit. Hierfür werden u. a. spezielle Ersetzungs-, Zuordnungssysteme und -Automaten sowie Erzeugungs- und Erkennungsverfahren definiert. Darauf aufbauend werden Aufwandsabschätzungen diskutiert und wichtige Eigenschaften des Ansatzes bewiesen.

Zur Verdeutlichung der betrachteten Bildanalyseaufgabe sowie zur Diskussion spezieller Aspekte der Analyse mit Produktionsnetzen werden für die Anwendung bildgestützter Navigation im Anhang Ergebnisse der Analyse von Objekten des Typs *Kreuzung*, *Brücke* und *Häuser* in mehreren Luftbildern dargestellt. Diese Referenzobjekte wurden in ungestörten Bildern sicher wiedergefunden. Fehlten in ungestörten Bildern die Referenzobjekte, so wurden sie in keinem Fall unzulässigerweise zugeordnet. Außerdem werden Ergebnisse der Analyse gestörter Bilder erläutert. Damit wird gezeigt, dass mit Produktionsnetzen nachvollziehbare und reproduzierbare Analyseergebnisse entstehen.

Neue Beiträge der vorliegenden Arbeit sind:

1. Eine auf Transformations- bzw. Reduktionssystemen aufbauende formale Definition von Systemen, Automaten und Verfahren zur Ersetzung, Zuordnung, Erzeugung und Erkennung von Mengen mit Instanzen für die strukturelle Musteranalyse;
2. Beweise für die Analyse mit Zuordnungsautomaten zur Konfluenz, Konvergenz und Parallelverarbeitung sowie zur Analyse in der Schleife mit einer Datenerfassung, die auf vorläufigen Analyseergebnissen aufbaut;
3. eine Erläuterung von Größen, die den Analyseaufwand bei Einsatz von Produktionsnetzen beeinflussen.



# Inhaltsverzeichnis

<b>Danksagung</b>	<b>i</b>
<b>Kurzfassung</b>	<b>iii</b>
<b>Inhaltsverzeichnis</b>	<b>v</b>
<b>Verzeichnis der Abbildungen</b>	<b>viii</b>
<b>Verzeichnis der Tabellen</b>	<b>x</b>
<b>Verzeichnis der Formelzeichen</b>	<b>xi</b>
<b>Verzeichnis der Abkürzungen und Symbole</b>	<b>xv</b>
<b>Verzeichnis der Funktionen und Relationen</b>	<b>xix</b>
<b>Verzeichnis weiterer Begriffe</b>	<b>xx</b>
<b>1 Einleitung</b>	<b>1</b>
1.1 Bildanalyse . . . . .	3
1.2 Überblick . . . . .	5
1.3 Ausgangspunkte: Ansätze und Systeme . . . . .	6
1.4 Motivation: Erweiterung der BPI-Architektur für Parallelrechner . . .	8
1.4.1 Ablaufprinzip der BPI-Architektur . . . . .	8
1.4.2 Demonstration eines Analyseablaufs mit der BPI-Architektur .	10
<b>2 Systeme und Verfahren für die Musteranalyse mit Produktionsnetzen</b>	<b>13</b>
2.1 Einleitender Überblick . . . . .	13
2.2 Transformationssysteme . . . . .	14
2.2.1 Abstrakte Transformationssysteme . . . . .	15
2.2.2 Beispiele für Transformationssysteme . . . . .	16
2.2.3 Transformationssysteme für Konfigurationen . . . . .	17
2.2.4 Automaten zur Zuordnung von Konfigurationen . . . . .	26
2.3 Erzeugungs- und Erkennungsverfahren . . . . .	30
2.3.1 Erzeugungs- und Erkennungsverfahren für Zeichenketten . . . .	31
2.3.2 Erzeugungs- und Erkennungsverfahren für Konfigurationen . .	33
2.4 Erkennung erzeugter Konfigurationen . . . . .	39
2.4.1 Ersetzende Erkennung erzeugter Konfigurationen . . . . .	39
2.4.2 Analyse mit Konfigurationen-Erkennungsverfahren . . . . .	41
2.4.3 Zuordnende Erkennung erzeugter Konfigurationen . . . . .	48

<b>3</b>	<b>Diskussion</b>	<b>50</b>
3.1	Ersetzende und zuordnende Erkennung . . . . .	50
3.2	Kontextfreie und kontextsensitive Erkennung . . . . .	51
3.3	Aufwand für die Musteranalyse mit Produktionsnetzen . . . . .	52
3.3.1	Algorithmen ohne Bewertungen . . . . .	54
3.3.2	Algorithmen mit bewerteten Analysezuständen . . . . .	70
3.3.3	Algorithmen mit bewerteten Zwischenergebnissen . . . . .	72
3.4	Mehrfachgenerierungen . . . . .	73
3.4.1	Mehrfachgenerierungen bei der Analyse mit Zuordnungsautomaten . . . . .	73
3.4.2	Mehrfachgenerierung von Primitivobjekten . . . . .	74
3.4.3	Mehrfachgenerierung von Zwischenergebnissen . . . . .	75
3.5	Begrenzung des Analyseaufwandes . . . . .	75
3.6	Minimierung des Analyseaufwandes bei Zuordnungssystemen . . . . .	78
3.7	Minimierung des Analyseaufwandes bei Zuordnungsautomaten . . . . .	80
<b>4</b>	<b>Vergleich mit anderen Ansätzen und Systemen</b>	<b>82</b>
4.1	Syntaktische Ansätze . . . . .	82
4.1.1	Vergleich mit syntaktischen Bildanalyseverfahren . . . . .	82
4.1.2	Vergleich mit erweiterten syntaktischen Bildanalyseverfahren . . . . .	83
4.1.3	Vergleich mit Koordinatengrammatiken . . . . .	86
4.2	Weitere Ansätze . . . . .	89
4.2.1	Vergleich mit PANDEMONIUM-Ansatz . . . . .	89
4.2.2	Vergleich mit tafelbasierten Systemen . . . . .	90
4.2.3	Vergleich mit regelbasierten Systemen . . . . .	90
4.2.4	Vergleich mit wissensbasierten Systemen . . . . .	91
4.2.5	Vergleich mit semantischen Netzen . . . . .	92
<b>5</b>	<b>Ausblick</b>	<b>94</b>
5.1	Verbesserung der Beschreibung der Ergebnisse . . . . .	95
5.2	Verbesserung der Modellierung . . . . .	96
5.2.1	Syntaktische Beschreibung der Eigenschaften von Assoziationen . . . . .	96
5.3	Reduzierung des Analyseaufwandes . . . . .	97
5.3.1	Detailanalyse nach Einweisung . . . . .	97
5.3.2	Verzögerter Einsatz aufwändiger Analyseprogramme . . . . .	97
5.3.3	Vermeidung von Mehrfachgenerierungen mit Ballungsanalysen . . . . .	98
5.3.4	Analyse mit erforderlichen und optionalen Zwischenergebnissen . . . . .	99
<b>6</b>	<b>Zusammenfassung</b>	<b>100</b>
<b>7</b>	<b>Literaturverzeichnis</b>	<b>101</b>
	<b>Anhänge</b>	<b>107</b>

<b>A</b>	<b>Erläuterung der Anwendung bildgestützter Navigation</b>	<b>107</b>
A.1	Bildmaterial und Vorverarbeitung der Bilddaten . . . . .	107
A.2	Extraktion der Primitivobjekte . . . . .	108
A.3	Analyseergebnisse für die bildgestützte Navigation . . . . .	111
A.3.1	Erläuterung der Verarbeitung . . . . .	118
A.3.2	Diskussion der Verarbeitungsergebnisse . . . . .	131
<b>B</b>	<b>Formale Darstellung der Systeme und Automaten</b>	<b>134</b>
B.1	Zuordnungssysteme . . . . .	134
B.1.1	Zuordnungssysteme mit Instanzen . . . . .	134
B.1.2	Parallele Zuordnungen . . . . .	141
B.1.3	Vollständige Zuordnungen . . . . .	150
B.1.4	Zuordnungssysteme mit Konzepten . . . . .	152
B.1.5	Zuordnungssysteme mit Ableitungen . . . . .	154
B.1.6	Eigenschaften von Assoziationen . . . . .	157
B.2	Zuordnungsautomaten . . . . .	162
B.2.1	Zuordnungsautomaten mit Konzepten . . . . .	162
B.2.2	Analysierbare Zustände . . . . .	167
B.2.3	Sequenzen von Teilanalysen und parallele Teilanalysen . . . . .	175
B.2.4	Zuordnungsautomaten mit Ableitungen . . . . .	179
B.3	Erkennung erzeugter Konfigurationen . . . . .	180
B.4	Aussagen für Aufwandsabschätzungen . . . . .	188
B.5	Projektionen von Instanzen, Konfigurationen und Assoziationen . . . . .	195
	<b>Stichwortverzeichnis</b>	<b>197</b>
	<b>Lebenslauf</b>	<b>203</b>

## Verzeichnis der Abbildungen

1-1	Vexierbild: Gesicht oder Vase . . . . .	1
1-2	Karlsruher Schloss mit markierter Statue . . . . .	2
1-3	Erkennung mit mehrdeutigen Segmenten . . . . .	2
1-4	Beispiel für ein Produktionsnetz . . . . .	4
1-5	Struktur eines Tafelsystems . . . . .	7
1-6	Ablaufzyklus eines BPI-Analyseprogramms . . . . .	9
1-7	Ein Anwendungsbeispiel zur Demonstration des Ablaufzyklus . . . . .	10
1-8	Zwischenergebnisse zu Abbildung 1-7 nach mehreren Auswahlzyklen . . . . .	11
2-1	Konfluente Transformationssysteme . . . . .	16
2-2	Ersetzen und zuordnen von Konfigurationen . . . . .	19
2-3	Parallel und asynchron zuordnen . . . . .	21
2-4	Parallel, verteilt und kaskadiert zuordnen . . . . .	22
2-5	Explorative Analyse mit "Nachschieben" der Daten während der Analyse . . . . .	28
2-6	Parallelverarbeitung mit parallelen und asynchronen Zuordnungen . . . . .	30
2-7	Erzeugungs- und Erkennungsverfahren . . . . .	32
2-8	Erzeugung von Konfigurationen . . . . .	34
2-9	Ersetzende Erkennung von Konfigurationen . . . . .	35
2-10	Zuordnende Erkennung von Konfigurationen . . . . .	36
2-11	Ausdehnung und Bereich von Instanzen sowie anlagernde Assoziationen . . . . .	38
2-12	Analyse mit Konfigurationen-Erkennungsverfahren . . . . .	42
2-13	Beispiel für Objekte einer Szene . . . . .	44
2-14	Ersetzende und zuordnende Erkennung . . . . .	49
3-1	Teilbäume bei kontextfreien Assoziationen . . . . .	51
3-2	rbb-Ableitungsbaum . . . . .	57
3-3	Zur Anzahl der Überprüfungen mit Algorithmus 3-10 . . . . .	60
3-4	Generierung von Ableitungsgraphen bei Speicherung nicht terminaler Instanzen bzw. von Analysezuständen . . . . .	71
3-5	Rechteck aus orientierten bzw. nicht orientierten Teilobjekten . . . . .	76
3-6	Durch Schatten nicht mehr detektierbare Dachbegrenzung . . . . .	78
4-1	Ergebnisse der Analyse mit einer Chromosomen-Grammatik . . . . .	84
4-2	Verbindung von PDL-Zeichenketten . . . . .	86
4-3	Das Zeichen "A" als PDL-Zeichenkette . . . . .	86
5-1	Top-down-Verbesserung von Ergebnissen . . . . .	96
5-2	Detailanalyse mit Einweisung . . . . .	98
A-1	Darstellung von Grauwertbildern . . . . .	108
A-2	Generierung der Primitivobjekte . . . . .	109
A-3	Vorverarbeitetes Grauwertbild mit Wegkreuzung . . . . .	110

A-4	Ungestörte Bilder (Autobahnbrücke, Wegkreuzung) . . . . .	112
A-5	Ungestörte Bilder (Wendehammer, Häuserformation, Schlossbild) . . .	113
A-6	Gestörte Bilder der Autobahnbrücke (gedreht und verschoben, 10% vergrößert, 20% verkleinert) . . . . .	114
A-7	Gestörte Bilder der Autobahnbrücke (verrauscht) . . . . .	115
A-8	Gestörte Bilder der Autobahnbrücke (mit Zeilenversatz, mit Wolken- schatten) . . . . .	116
A-9	Überlagerte Bilder . . . . .	117
A-10	Verarbeitungsergebnisse mit ungestörten Bildern (Autobahnbrücke, Wegkreuzung, Wendehammer) . . . . .	119
A-11	Verarbeitungsergebnisse mit ungestörten Bildern (Häuserformation, Schlossbilder 1. Teil) . . . . .	120
A-12	Verarbeitungsergebnisse mit ungestörten Bildern (Schlossbilder 2. Teil) . . . . .	121
A-13	Verarbeitungsergebnisse mit gestörten Bildern der Autobahnbrücke (gedreht und verschoben, 10% vergrößert, 20% verkleinert) . . . . .	122
A-14	Verarbeitungsergebnisse mit gestörten Bildern der Autobahnbrücke (verrauscht) . . . . .	123
A-15	Verarbeitungsergebnisse mit gestörten Bildern der Autobahnbrücke (mit Zeilenversatz, mit Wolken-schatten) . . . . .	124
A-16	Verarbeitungsergebnisse mit zweifach überlagerten Bildern (Brücke und Wendehammer in Brücke-Wendehammer-Überlagerung) . . . . .	125
A-17	Verarbeitungsergebnisse mit zweifach überlagerten Bildern (Häuser- formation und Kreuzung in Brücke-Wendehammer-Überlagerung) . . .	126
A-18	Verarbeitungsergebnisse mit dreifach überlagerten Bildern (Brücke und Kreuzung in Brücke-Kreuzung-Wendehammer-Überlagerung) . . .	127
A-19	Verarbeitungsergebnisse mit dreifach überlagerten Bildern (Wende- hammer und Häuserformation in Brücke-Kreuzung-Wendehammer- Überlagerung) . . . . .	128
A-20	Verarbeitungsergebnisse mit überlagerten Bildern (Brücke und Kreuzung in vierfach überlagertem Bild) . . . . .	129
A-21	Verarbeitungsergebnisse mit überlagerten Bildern (Wendehammer und Häuserformation in vierfach überlagertem Bild) . . . . .	130
A-22	Vergleich der Verarbeitungsergebnisse . . . . .	131
B-1	Mehrfachgenerierungen durch geschachtelte Suchschleifen (1. Ebene) .	189
B-2	Mehrfachgenerierungen durch geschachtelte Suchschleifen . . . . .	189
B-3	Fortpflanzung von Mehrfachgenerierungen . . . . .	190
B-4	Zum Beweis von Lemma B.91 . . . . .	191

## Verzeichnis der Tabellen

3-1	Mehrfachgenerierungen mit dem ersetzenden Algorithmus 3-10 und rbb-Ableitungsbäumen . . . . .	61
3-2	Ohne bzw. mit Speicherung notwendige Anzahl der Überprüfungen mit dem ersetzenden Algorithmus 3-10 bzw. dem zuordnenden Algorithmus 3-19 und rbb-Ableitungsbäumen . . . . .	67
3-3	Anzahl der Überprüfungen mit Suchbereichen . . . . .	69



## Verzeichnis der Formelzeichen

$\square$	: Ende einer Definition, eines Satzes, eines Beweises, eines Beispiels, eines Abschnittes mit Verweisen
$\emptyset$	: leere Menge
$\neg a$	: nicht $a$
$a := b$	: $a$ wird durch $b$ definiert
$a = b$	: $a$ gleich $b$
$a \approx b$	: $a$ näherungsweise gleich $b$
$a \stackrel{\wedge}{=} b$	: $a$ entspricht $b$
$a \stackrel{>}{\approx} b$	: $a \approx b \vee a > b$
$a \stackrel{<}{\approx} b$	: $a \approx b \vee a < b$
$a \wedge b$	: $a$ und $b$
$\left\{ \begin{array}{l} a_1 \\ a_2 \\ \vdots \\ a_n \end{array} \right.$	: $a_1 \wedge a_2 \wedge \cdots \wedge a_n$
$a \vee b$	: $a$ oder $b$
$a \rightarrow b, b \leftarrow a$	: aus $a$ folgt $b$ : $\neg a \vee b$
$a \leftrightarrow b$	: $a$ genau dann, wenn $b$ : $(a \rightarrow b) \wedge (a \leftarrow b)$
$f : X \rightarrow Y$	: $f$ ist eine Funktion der Menge $X$ in die Menge $Y$
$f : x \mapsto f(x)$	: $x$ wird mit der Funktion $f$ auf $f$ von $x$ abgebildet
$f(\mathcal{B} \mathcal{A} \mapsto \mathcal{X})$	: $f$ von $\mathcal{B}$ mit $\mathcal{A}$ für $\mathcal{X}$ (Seite 24)
$\{x \in X   P(x)\}$	: Menge aller $x$ der Menge $X$ , für die $P$ von $x$ gilt
$\forall x \in X : P(x)$	: für alle $x$ der Menge $X$ gilt $P$ von $x$
$\forall x, y \in X : P(x, y)$	: $\forall x \in X : [\forall y \in X : P(x, y)]$
$\exists x \in X : P(x)$	: für ein $x$ der Menge $X$ gilt $P$ von $x$
$\exists! x \in X : P(x)$	: für genau ein $x$ der Menge $X$ gilt $P$ von $x$
$\exists x, y \in X : P(x, y)$	: $\exists x \in X : [\exists y \in X : P(x, y)]$
$x \in X, X \ni x$	: $x$ ist Element von $X$
$x \notin X, X \not\ni x$	: $x$ ist kein Element von $X$
$X \subseteq Y, Y \supseteq X$	: $X$ ist Teilmenge von $Y$ : $\forall x \in X : x \in Y$
$X \subset Y, Y \supset X$	: $X$ ist echte Teilmenge von $Y$ ( $X \subseteq Y$ und $X \neq Y$ )
$X \cup Y$	: Vereinigung der Mengen $X$ und $Y$ : $\{x   x \in X \vee x \in Y\}$
$X \cap Y$	: Schnitt der Mengen $X$ und $Y$ : $\{x   x \in X \wedge x \in Y\}$
$X \times Y$	: kartesisches Produkt der Mengen $X$ und $Y$ : $\{(x, y)   x \in X \wedge y \in Y\}$
$X \setminus Y$	: Differenz der Mengen $X$ und $Y$ : $\{x   x \in X \wedge x \notin Y\}$
$\prod_{i=1}^n X_i$	: $X_1 \times X_2 \times \cdots \times X_n$
$\bigcup_{i \in N} \mathcal{X}_i$	: $\{\gamma   \exists i \in N : \gamma \in \mathcal{X}_i\}$
$\bigcap_{i \in N} \mathcal{X}_i$	: $\{\gamma   \forall i \in N : \gamma \in \mathcal{X}_i\}$

$X^n$	: $\underbrace{X \times X \times \cdots \times X}_{n\text{-mal}}$
$ X $	: Anzahl der Elemente (Mächtigkeit) der Menge $X$
$\wp(X)$	: Menge aller Teilmengen (Potenzmenge) der Menge $X$
$R^{-1}$	: invertierte Relation: $\{(b, a)   (a, b) \in R\}$
$Q \circ R$	: Produkt der binären Relationen $Q$ und $R$ (mit $Q, R \subseteq B \times B$ gilt: $Q \circ R := \{(a, c)   \exists b \in B : (a, b) \in Q \wedge (b, c) \in R\}$ )
$a \Rightarrow b$	: $a$ wird direkt nach $b$ transformiert (Seite 15)
$a \xrightarrow{n} b$	: $a$ wird in $n$ Schritten nach $b$ transformiert
$a \xrightarrow{*} b$	: $a$ wird nach $b$ transformiert
$a \xrightarrow{+} b$	: $a$ wird mit einem Schritt oder mehreren Schritten nach $b$ transformiert
$a \xrightarrow{\epsilon} b$	: $a$ wird mit maximal einem Schritt nach $b$ transformiert
$\mathcal{A} \overset{\mathbf{A}}{\sim} \mathcal{B}$	: $\mathcal{A}$ assoziiert $\mathcal{B}$ : $(\mathcal{A}, \mathcal{B}) \in \mathbf{A}$
$\mathcal{A} \overset{Z}{ } \mathcal{B}$	: die Konfiguration $\mathcal{B}$ wird der Konfiguration $\mathcal{A}$ mit dem $Z$ -System $Z$ direkt zugeordnet (Seite 134)
$\mathcal{A} \overset{\mathbf{A}}{\underset{z}{ }} \mathcal{B}$	: $\mathcal{A} \overset{Z}{ } \mathcal{B}$ ; $Z = (\mathcal{I}, \mathbf{A})$ (geschrieben mit der Transformationsrelation $\Rightarrow$ , Seite 19)
$\mathcal{A} \overset{*Z}{ } \mathcal{B}$	: die Konfiguration $\mathcal{B}$ wird $\mathcal{A}$ mit $Z$ zugeordnet
$\mathcal{A} \overset{*A}{\underset{z}{ }} \mathcal{B}$	: $\mathcal{A} \overset{*Z}{ } \mathcal{B}$ ; $Z = (\mathcal{I}, \mathbf{A})$ (geschrieben mit der Transformationsrelation $\Rightarrow$ )
$\mathcal{A} \overset{\epsilon Z}{ } \mathcal{B}$	: die Konfiguration $\mathcal{B}$ wird $\mathcal{A}$ mit dem $Z$ -System $Z$ mit maximal einem Schritt zugeordnet
$\mathcal{A} \overset{+Z}{ } \mathcal{B}$	: das $Z$ -System $Z$ ordnet der Konfiguration $\mathcal{A}$ mit einer Zuordnung oder mehreren direkten Zuordnungen $\mathcal{B}$ zu
$\mathcal{A} \overset{\parallel Z}{ } \mathcal{B}$	: die Konfiguration $\mathcal{B}$ wird $\mathcal{A}$ mit dem $Z$ -System $Z$ direkt parallel zugeordnet
$\mathcal{A} \overset{\mathbf{A}}{\underset{\parallel z}{ }} \mathcal{B}$	: $\mathcal{A} \overset{\parallel Z}{ } \mathcal{B}$ ; $Z = (\mathcal{I}, \mathbf{A})$ (geschrieben mit der Transformationsrelation $\Rightarrow$ )
$\mathcal{A} \overset{* \parallel Z}{ } \mathcal{B}$	: die Konfiguration $\mathcal{B}$ wird $\mathcal{A}$ mit dem $Z$ -System $Z$ parallel zugeordnet
$\mathcal{A} \overset{*A}{\underset{\parallel z}{ }} \mathcal{B}$	: $\mathcal{A} \overset{* \parallel Z}{ } \mathcal{B}$ ; $Z = (\mathcal{I}, \mathbf{A})$ (geschrieben mit der Transformationsrelation $\Rightarrow$ )
$\mathcal{A} \overset{v Z}{\underset{W}{ }} \mathcal{B}$	: die Konfiguration $\mathcal{B}$ wird $\mathcal{A}$ mit dem Tupel $W$ durch das $Z$ -System $Z$ unabhängig verteilt zugeordnet (Seite 142)
$\mathcal{A} \overset{v \parallel Z}{\underset{W}{ }} \mathcal{B}$	: die Konfiguration $\mathcal{B}$ wird $\mathcal{A}$ mit der Menge $W$ durch das $Z$ -System $Z$ verteilt zugeordnet

$\mathcal{A} \xrightarrow{kZ} \mathcal{B}$	: die Konfiguration $\mathcal{B}$ wird $\mathcal{A}$ mit dem Z-System $Z$ kaskadiert zugeordnet
$\mathcal{A} \xrightarrow[kz]{\mathbf{A}} \mathcal{B}$	: $\mathcal{A} \xrightarrow{kZ} \mathcal{B}$ ; $Z = (\mathcal{I}, \mathbf{A})$ (geschrieben mit der Transformationsrelation $\Rightarrow$ )
$\mathcal{A} \xrightarrow{a  Z} \mathcal{B}$	: die Konfiguration $\mathcal{B}$ wird $\mathcal{A}$ mit dem Z-System $Z$ direkt asynchron zugeordnet
$\mathcal{A} \xrightarrow{*a  Z} \mathcal{B}$	: die Konfiguration $\mathcal{B}$ wird $\mathcal{A}$ mit dem Z-System $Z$ asynchron zugeordnet
$\mathcal{A} \xrightarrow{\dagger Z} \mathcal{B}$	: die Konfiguration $\mathcal{B}$ wird $\mathcal{A}$ mit dem Z-System $Z$ abschließend zugeordnet
$\mathcal{A} \xrightarrow{mZ} \mathcal{B}$	: die Konfiguration $\mathcal{B}$ wird $\mathcal{A}$ mit dem Z-System $Z$ direkt mehrfach zugeordnet
$\mathcal{A} \xrightarrow{*mZ} \mathcal{B}$	: die Konfiguration $\mathcal{B}$ wird $\mathcal{A}$ mit dem Z-System $Z$ mehrfach zugeordnet
$\mathcal{A} \xrightarrow{\forall Z} \mathcal{B}$	: die Konfiguration $\mathcal{B}$ wird $\mathcal{A}$ mit dem Z-System $Z$ direkt vollständig zugeordnet
$\mathcal{A} \xrightarrow{\forall  Z} \mathcal{B}$	: die Konfiguration $\mathcal{B}$ wird $\mathcal{A}$ mit dem Z-System $Z$ direkt vollständig parallel zugeordnet
$\mathcal{A} \xrightarrow{\forall a  Z} \mathcal{B}$	: die Konfiguration $\mathcal{B}$ wird $\mathcal{A}$ mit dem Z-System $Z$ direkt vollständig asynchron zugeordnet
$p \xrightarrow{A} q$	: der Zustand $p$ des Automaten $A$ geht direkt über in den Zustand $q$ (Seite 163)
$p \xrightarrow[A]{\Rightarrow} q$	: $p \xrightarrow{A} q$ (geschrieben mit der Transformationsrelation $\Rightarrow$ ) (Seite 27)
$p \xrightarrow{  A} q$	: der Zustand $p$ des Automaten $A$ geht parallel direkt über in den Zustand $q$
$p \xrightarrow{aA} q$	: der Zustand $p$ des Automaten $A$ geht asynchron direkt über in den Zustand $q$
$p \xrightarrow{*A} q$	: der Zustand $p$ des Automaten $A$ geht über in den Zustand $q$
$p \xrightarrow[A]{*} q$	: $p \xrightarrow{*A} q$ (geschrieben mit der Transformationsrelation $\Rightarrow$ )
$p \xrightarrow{+A} q$	: der Zustand $p$ des Automaten $A$ geht mit einem Übergang oder mehreren direkten Übergängen in den Zustand $q$ über
$p \xrightarrow{\dagger A} q$	: der Zustand $p$ des Automaten $A$ geht abschließend über in den Zustand $q$
$\alpha^{(X)}$	: Projektion des Symbols $\alpha$ auf die Menge $X$ (Seite 180)

$\mathcal{K}^{(X)}$	: Projektion der Konfiguration $\mathcal{K}$ auf die Menge $X$
$\mathbf{A}^{(X)}$	: Projektion der Relation $\mathbf{A}$ auf die Menge $X$
$W^i(V)$	: Menge aller Zeichenketten der Länge $i$ mit Zeichen der Menge $V$ (Seite 17)
$W^*(V)$	: Menge aller Zeichenketten mit Zeichen der Menge $V$
$W^+(V)$	: $W^*(V) \setminus \{\lambda\}$
$\mathcal{R} \boxtimes_{\mathbf{A}} U_r$	: die Konfiguration $\mathcal{R}$ ist mit der Menge der Überprüfungsaufträge $U_r$ und mit der Menge der Assoziationen $\mathbf{A}$ analysierbar (Seite 27, 168)
$ld(i)$	: Logarithmus zur Basis 2

## Verzeichnis der Abkürzungen und Symbole

$(V_T, V_N, \Sigma, P_i)$	: Chomsky-Grammatik (Gl. 2-13)
$(\mathcal{I}_T, \mathcal{I}_N, \Sigma, \mathbf{A})$	: Szenario-Beschreibung (Gl. 2-15 ff.)
$\alpha, \beta, \dots, \omega$	: Instanz (Gl. B-1)
$a, b, \dots, z$	: Zustand eines Zuordnungsautomaten (Gl. B-86)
$A, \dots, I, K, L, \dots, Z$	: Menge von Überprüfungsaufträgen (Gl. B-83)
$\mathcal{A}, \dots, \mathcal{I}, \mathcal{K}, \mathcal{L}, \dots, \mathcal{Z}$	: Konfiguration (Menge von Instanzen)
$A, B, C, D, \dots$	: Konfiguration von Produktionen
$a$	: Auswahlfunktion (Gl. B-82)
$A(\gamma)$	: Ausdehnung einer Instanz $\gamma$ (Gl. B-76)
$A$	: Zuordnungsautomat (Gl. B-86)
$A_A$	: Zuordnungsautomat mit Ableitungen (Gl. B-125)
$A_M$	: Zuordnungsautomat mit Merkmalvektoren
$AF$	: Analysefunktion (Gl. B-15)
$\mathbf{A}$	: Menge der Instanz-Assoziationen (Gl. B-1)
$\mathbf{A}_g$	: Menge der Instanz-Assoziationen einer Menge von Überprüfungsaufträgen (Gl. B-86)
$\mathbf{A}_K$	: Menge der Instanz-Assoziationen eines Zuordnungssystems mit Konzepten (Gl. B-46)
$\mathbf{A}_r$	: reduzierte Menge der Instanz-Assoziationen (Seite 78)
$\mathbf{A}_u$	: Menge der Instanz-Assoziationen eines Überprüfungsauftrages (Gl. B-85)
$\mathbf{A}_X$	: Menge aller Instanz-Assoziationen mit $X$ (Gl. B-48)
$B(\gamma)$	: Bereich einer Instanz $\gamma$ (Gl. B-85)
BPI	: Blackboard Based Production System for Image Interpretation
CFAR	: Constant False Alarm Rate
CN	: Hilfsfunktion zur Berechnung von Beschreibungsgraphen (Gl. B-61)
CYK	: Cocke-Younger-Kasami (Parser)
FGAN	: Forschungsgesellschaft für Angewandte Naturwissenschaften e.V.
FIM	: Forschungsinstitut für Informationsverarbeitung und Mustererkennung der FGAN (wurde 1999 Teil des FOM)
FN	: Hilfsfunktion zur Berechnung von Beschreibungsgraphen (Gl. B-62)
FOM	: Forschungsinstitut für Optronik und Mustererkennung der FGAN (entstand 1999 aus den Instituten FfO und FIM)
$G$	: Zeichenkettengrammatik (Gl. 2-13)
$i$	: Instanziierungsfunktion (Gl. B-42)
I-Assoziation	: Instanz-Assoziation
$I_X$	: Menge der Indices von Ableitung $\gamma \in \mathcal{I}_A$ (Gl. B-52)

$\mathcal{I}$	: Menge der Instanzen
$\mathcal{I}_\alpha$	: Konzeptmenge von $\alpha$ (Gl. B-45)
$\mathcal{I}_a$	: Menge der assoziierten Instanzen (Gl. B-2)
$\mathcal{I}_A$	: Menge der Ableitungen (Gl. B-54)
$\mathcal{I}_i$	: Menge der instanziierten Konzepte (Gl. B-47)
$\mathcal{I}_{iA}$	: Menge der instanziierten Konzepte mit Ableitungen (Gl. B-65)
$\mathcal{I}_K$	: Menge der Konzept-Repräsentanten (Gl. B-43)
$\mathcal{I}_l$	: Menge der lokal überprüfbaren Instanzen (Gl. B-122)
$\mathcal{I}_M$	: Menge der Merkmalvektoren (Gl. B-51)
$\mathcal{I}_{Ma}$	: Menge der Merkmalvektoren mit ausdehnungsbeschreibenden Merkmalen (Gl. B-76)
$\mathcal{I}_N$	: Menge der nichtterminalen Instanzen (Seite 31)
$\mathcal{I}_r$	: reduzierte Menge der Instanzen (Seite 78)
$\mathcal{I}_T$	: Menge der terminalen Instanzen (Seite 31)
$\mathbf{I}$	: Tupel von Attributen bzw. Merkmalen (Gl. B-50)
$\mathbf{I}_a$	: Tupel ausdehnungsbeschreibender Merkmale (Gl. B-76)
$\lambda$	: Zeichenkette der Länge 0
$l$	: Approximationslänge (Abschnitt A.2)
$L_C^a$	: analysierte Sprache (Gl. 2-14)
$L_C^g$	: generierte Sprache (Gl. 2-13)
$M_i$	: i-tes Merkmal (Gl. B-50)
$m_{i,j}$	: Anzahl der Mehrfachgenerierungen der Zwischenergebnisse $\nu_{i,j}$ (Gl. 3-13)
MIps	: Mega Instruktionen pro Sekunde
$\mathbb{N}$	: Menge der natürlichen Zahlen
$\mathbb{N}_0$	: Menge der nicht negativen ganzen Zahlen
$\mathbb{N}_x$	: $\{i \in \mathbb{N}_0 \mid i \geq x\}$ , $x \in \mathbb{N}_0$
$n$	: Anzahl Binärisierungsschnitte (Abschnitt A.2)
$n_{tbn}$ bzw. $n_{tbn4}$ , $n_{tbn8}$ , $n_{tbn16}$	: Anzahl der Überprüfungen bei Ableitungsbäumen mit Speicherung der Zwischenergebnisse bzw. diese Größe mit 4, 8, 16 zusammenzufassenden terminalen Instanzen (Gl. 3-27)
$n_{tbn4}$ bis $n_{tbn16}$	: Näherungen für $n_{tbn4}$ bis $n_{tbn16}$ (Gl. 3-30)
$n_{tbn}$	: Anzahl der Überprüfungen bei Ableitungsbäumen (Gl. 3-23) mit Speicherung der Zwischenergebnisse und Suchbereichen
$n_{tbo}$ bzw. $n_{tbo4}$ , $n_{tbo8}$ , $n_{tbo16}$	: Anzahl der Überprüfungen bei Ableitungsbäumen ohne Speicherung der Zwischenergebnisse bzw. diese Größe mit 4, 8, 16 zusammenzufassenden terminalen Instanzen (Gl. 3-24)
$n_{tbo}$	: Anzahl der Überprüfungen bei Ableitungsbäumen (Gl. 3-18) ohne Speicherung der Zwischenergebnisse aber mit Suchbereichen
$n_{tk}$	: Anzahl der Überprüfungen bei Ableitungsketten (Gl. 3-6)

$O_b^H$	: Menge bekannter Hintergrundobjekte (Gl. 2-27)
$O_u^H$	: Menge unbekannter Hintergrundobjekte (Gl. 2-29)
$O^K$	: Menge der Kontextobjekte
$O^V$	: Menge der Verbundobjekte (Gl. 2-27)
$O^Z$	: Menge der Zielobjekte (Gl. 2-27)
$P$	: Menge der Produktionen eines Zuordnungssystems mit Konzepten (Gl. B-42)
$P$	: Menge der Produktionen einer Zeichenkettengrammatik
$\dots, \mathcal{P}, \mathcal{Q}, \mathcal{R}, \mathcal{S}, \mathcal{T}, \dots$	: Konfiguration für Zuordnungen bzw. Zustandsübergänge
PDL	: Picture Description Language (Seite 85)
$P_M$	: Menge aller Teilmengen mit Indices $\in I_M$ von Ableitung $\mathcal{I}_A$
rbb-Baum	: regulär binärer balancierter Baum
$\mathcal{R}$	: Menge der zu einem Startsymbol zusammenzufassenden terminalen Instanzen (Abschnitt 3.3)
$\sigma$	: Startsymbol
$\Sigma$	: Menge der Startsymbole
$S_a$	: analysiertes Szenario
$S_a^{te}$ bzw. $S_a^{tz}$ bzw. $S_a^{tz'}$	: teilweise ersetzend bzw. zuordnend analysiertes Szenario bzw. mit Überprüfung der Ableitungsgraphen teilweise zuordnend analysiertes Szenario (Gl. 2-19, 2-20, 2-23)
$S_a^{ve}$ bzw. $S_a^{vz}$ bzw. $S_a^{vz'}$	: vollständig ersetzend bzw. zuordnend analysiertes Szenario bzw. mit Überprüfung der Ableitungsgraphen vollständig zuordnend analysiertes Szenario (Gl. 2-18, 2-22)
$S_g$ bzw. $S_g^O$	: generiertes Szenario (Gl. 2-15) bzw. mit Obermengen generiertes Szenario (Gl. 2-16)
STS	: Semi-Thue-System (Gl. 2-6)
$T$	: Typ-Merkmal (Gl. B-50)
$\mathcal{T}$	: zu analysierende Menge terminaler Instanzen (Abschnitt 3.3)
TS	: Thue-System (Gl. 2-6)
$t_{i,j}$	: Anzahl der einfach generierten Zwischenergebnisse $\mu_{i,j}$
$t_{i,j}^m$	: Anzahl der mehrfach generierten Zwischenergebnisse $\nu_{i,j}$
$u$	: Menge der mit einer Konfiguration zu bildenden Überprüfungsaufträge (Gl. B-83)
$u^r$	: Menge der mit einer Konfiguration zu bildenden reduzierten Überprüfungsaufträge (Gl. B-127)
$\hat{u}$	: Menge der zur vereinten Überprüfung zu bildenden Überprüfungsaufträge (Gl. B-121)
$\dots, \mathcal{U}, \mathcal{V}, \mathcal{W}, \mathcal{X}, \mathcal{Y}, \mathcal{Z}$	: Konfiguration von Assoziationen
$V$	: Zeichenmenge (Gl. 2-4)
$V_N$	: Menge nichtterminaler Zeichen (Gl. 2-13)
$V_T$	: Menge terminaler Zeichen (Gl. 2-13)
$v_{ibr}$ bzw. $v_{ibr4}$ , $v_{ibr8}$ , $v_{ibr16}$	: Verhältnis von $n_{ibo}$ zu $n_{ibm}$ bzw. diese Größe mit 4, 8, 16 zusammenzufassenden terminalen Instanzen (Gl. 3-36)

$V, W$	: Tupel von Assoziationenmengen (Gl. B-21)
$Z, Z$ -System	: Zuordnungssystem
$Z_A, ZA$ -System	: Zuordnungssystem mit Ableitungen
$Z_K, ZK$ -System	: Zuordnungssystem mit Konzepten
$Z$ -Automat	: Zuordnungsautomat
$ZA$ -Automat	: Zuordnungsautomat mit Ableitungen



## Verzeichnis der Funktionen und Relationen

Ableitungsgraph (Gl. B-59) .....	155
abschließend übergehen (Gl. B-90) .....	164
abschließend zuordnen (Gl. B-5) .....	135
anlagernde Menge der Assoziationen (Gl. B-77) .....	37
Assoziationen (Menge der) (Gl. B-1) .....	25
asynchron zuordnen (Gl. B-24) .....	143
analysierbarer Zustand (Gl. B-98) .....	27
Analysefunktion (Gl. B-15) .....	19
Ausdehnung einer Instanz (Gl. B-76) .....	37
Beschreibungsgraph (Gl. B-63) .....	156
deterministische Menge der Assoziationen (Gl. B-73) .....	158
einfach assoziierende Menge der Assoziationen (Gl. B-72) .....	26
ersetzen von Konfigurationen (Gl. 2-7) .....	18
ersetzende Erkennung (Gl. 2-18) .....	34
ersetzende Erzeugung (Gl. 2-15) .....	33
global kontextfreie Menge der Assoziationen (Gl. B-71) .....	26
graph-differenzierende Menge der Assoziationen (Gl. B-75) .....	158
Instanziierungsfunktion (Gl. B-44) .....	24
kaskadiert zuordnen (Gl. B-23) .....	22
konfluente Transformation (Gl. 2-2) .....	15
konvergente Transformation .....	16
kontextsensitive Menge der Assoziationen (Gl. B-71) .....	26
lokal kontextfreie Menge der Assoziationen (Gl. B-69) .....	26
mehrfach assoziierende Menge der Assoziationen (Gl. B-72) .....	26
mehrfach zuordnen (Gl. B-27) .....	22
merkmal-differenzierende Menge der Assoziationen (Gl. B-74) .....	158
parallel zuordnen (Gl. 2-10) .....	20
Produktionen (Gl. B-46) .....	24
Produktionsnetz (Gl. B-58) .....	155
Projektionen .....	195
teilweise Erkennung .....	34
terminierende Transformation .....	16
übergehen in einen Zustand (Gl. B-89) .....	164
Überprüfungsaufträge (Gl. B-83) .....	26
verteilt zuordnen (Gl. B-22) .....	21
vexierende Erkennung .....	47
vexierende Menge der Assoziationen (Gl. B-67) .....	157
vollständige Erkennung .....	34
vollständig zuordnen (Gl. B-39) .....	23
zuordnen von Konfigurationen (Gl. 2-8, Gl. B-4) .....	18
zuordnende Erkennung (Gl. 2-19) .....	35

## Verzeichnis weiterer Begriffe

Grammatik .....	32
Instanz .....	38
nichtterminale Instanz .....	31
Konzept .....	38
Startsymbol .....	30
Sprache .....	32
Szene .....	33
Szenario .....	33
Szenario-Beschreibung .....	33
terminale Instanz .....	31

Siehe auch Stichwortverzeichnis ab Seite 197.

# 1 Einleitung

Bei der automatischen Auswertung von Bilddaten sind eine Reihe von Problemen zu berücksichtigen, die je nach Aufgabenstellung und Randbedingungen unterschiedliches Gewicht haben.



**Abb. 1-1:** Vexierbild: Gesicht oder Vase  
links: Originalbild, rechts mit markiertem Ausschnitt

Ein grundsätzliches Problem bei der Erkennung von Objekten ist die Behandlung mehrdeutiger und alternativer Interpretationen, die sich anhand von Vexierbildern erläutern lassen. Diese Bilder zeigen besonders deutlich den bereits von den Gestaltpsychologen /Koe69/ formulierten Zusammenhang zwischen Teil und Ganzem, nachdem etwas, das Teil einer größeren Einheit ist, eine besondere Interpretation erhält. Umgekehrt erhält auch das Ganze seine Interpretation aus den es aufbauenden Teilen, so dass sich zwischen Teil und Ganzem eine gegenseitige Abhängigkeit und Rückkopplung ergibt. Beispielsweise ist ohne die Interpretation des Ganzen der markierte untere Teil in dem von Rubin /Rub21/ angegebenen Vexierbild (Abbildung 1-1) nur als Kreisbogen zu bezeichnen. Erst nachdem die Gesamtheit als zwei Gesichter bzw. als Vase interpretiert wurde, kann der Kreisbogen zum Kinn des Gesichts bzw. zum Stiel der Vase werden.

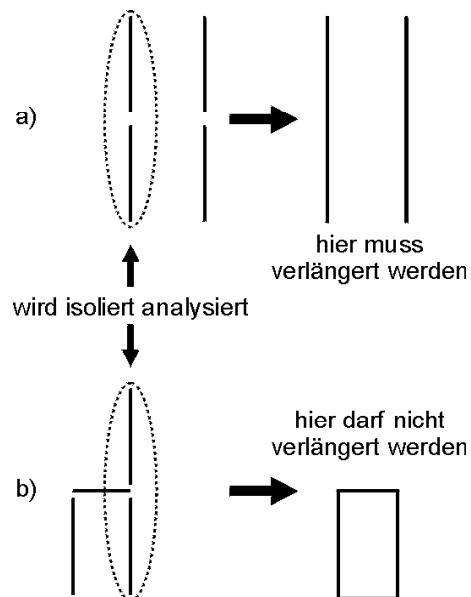
Zur Interpretation des Vexierbildes mussten für den Kreisbogen die zwei konkurrierenden Alternativen *Kinn* und *Stiel* berücksichtigt werden. Bei komplexen Bildern ist für jedes Teilobjekt in der Regel eine größere Anzahl von Interpretationen möglich. Der in Abbildung 1-2 markierte Teil zeigt eine Statue auf einem Sockel vor dem Karlsruher Schloss. Um den markierten weißen Fleck als Statue zu identifizieren, könnte so vorgegangen werden, dass der Hintergrund als Schloss erkannt wird, das Objekt unterhalb des weißen Flecks als Sockel und der weiße Fleck als Statue, die auf einem Sockel vor einem Schloss steht. Wird jedes Teilobjekt für sich betrachtet, so sind für das jeweilige Teilobjekt in der Regel viele unterschiedliche Interpretationen möglich, die grundsätzlich alle bei der automatischen Bildauswertung zu berücksichtigen sind.

Speziell bei gestörten Bildern und vage beschriebenen Teilobjekten ergibt sich so eine große Vielfalt von Interpretationsalternativen. Liegen zum Beispiel in einem gestörten Bild auf einer Geraden zwei voneinander abgesetzte Linienstücke



**Abb. 1-2:** Karlsruher Schloss mit markierter Statue

vor, so hängt die Zulässigkeit der Überbrückung von der Interpretation des Ganzen ab. In Abbildung 1-3 a) muss, wenn der Streifen erkannt werden soll, das Linienpaar zusammengefasst werden; in Abbildung 1-3 b) muss dagegen zum Aufbau des Rechtecks die nicht verlängerte untere Linie verwendet werden. Solange diese Interpretation noch nicht vorliegt, müssen beide Alternativen betrachtet werden: zwei getrennte kürzere Linienstücke als die eine Alternative und eine verlängerte Linie als die zweite Alternative. Diese Kombinatorik sollte von einer automatischen Bildanalyse bewältigt werden.



**Abb. 1-3:** Erkennung mit mehrdeutigen Segmenten

## 1.1 Bildanalyse

Vor der eigentlichen Analyse wird in vielen Fällen ein Vorverarbeitungsverfahren /Kaz80/ vorgeschaltet, das die auszuwertenden Grauwertbilder auf die anschließende Analyse vorbereitet. Beispielsweise werden Bilder gefiltert oder geometrisch und radiometrisch entzerrt. Das Ergebnis der Vorverarbeitung ist wiederum ein Grauwertbild.

Die anschließende Erkennung des Bildes kann bei *einfachen* Mustern in einem einzigen Zuordnungsschritt und bei *komplexen* Mustern /Nie89/ mit einer mehrstufigen Analyse erfolgen. Das Ergebnis der Analyse ist eine Liste mit erkannten Objekten.

### Einstufige Erkennung einfacher Muster

Ein Beispiel für die einstufige Erkennung ist die Multispektralklassifikation /Kaz80/, bei der allen Bildpunkten eines Farb-Bildes jeweils eine Klasse zugeordnet wird. Die Farben beschreiben den jeweiligen Punkt und sind seine Merkmale, die für seine Klassenzuordnung ausgewertet werden.

Bei der Multispektralklassifikation werden von Sensoren vermessene Bildpunkte klassifiziert. In der Regel sind für eine Bildanalyse aber andere Objekte von Interesse, z. B. Kreissegmente, Ecken und Geradenstücke oder Flächen, die durch Merkmale zu beschreiben sind. Zur Analyse dieser Objekte werden geeignete Bildpunkte zu Segmenten zusammengefasst, denen jeweils ein Merkmalvektor zugeordnet wird. Ein Beispiel für diesen Segmentationsschritt ist das Mehrschwellenverfahren (level slicing) /ATh89/. Anschließend werden den extrahierten Segmenten entsprechend ihrer Merkmale Objektklassen zugewiesen. Beispielsweise könnten die extrahierten Flächen unmittelbar als *Kreuzung* bzw. als *Fläche ist keine Kreuzung* klassifiziert werden, so dass die aus den Grauwertbildern extrahierten Flächen mit nur einem Zuordnungsschritt erkannt werden.

Dieser Ansatz wird in der Regel nur bei relativ einfachen Aufgabenstellungen unter kontrollierten Randbedingungen, wie beispielsweise die Erkennung gedruckter Schriftzeichen, angewandt. Bildsegmentation und Merkmalgenerierung werden bei diesen Systemen auf spezielle Aufnahmebedingungen adaptiert und hängen von kritischen Schwellereinstellungen ab. Analyseergebnisse, die mit diesem Ansatz bei komplexen Aufgabenstellungen entstehen würden, wären daher weder nachvollziehbar noch reproduzierbar.

### Mehrstufige Erkennung komplexer Muster

In vielen Fällen muss ein Objekt aus mehreren Segmenten aufgebaut werden. Im oben erwähnten Kreuzungsbeispiel (Abbildung A-3) hieße dies, dass eine Kreuzung z. B. aus den Straßen zusammengesetzt wird, die in sie einmünden. Die unmittelbar aus den Bilddaten extrahierten Segmente werden diesen Straßen zugeordnet. Diese Vorgehensweise, bei der zu analysierende Objekte durch ihre Teile

und die Relationen zwischen den Teilen definiert sind, wird in der vorliegenden Arbeit zu Grunde gelegt.

Bei der mehrstufigen Analyse ist festzuhalten, welche Zuordnungen zulässig sind. Es bietet sich an, die Zuordnungsschritte entsprechend einer Teil-von-Relation festzulegen und gleichzeitig die Teil-von-Relation mit Produktionen zu präzisieren. So gelangt man zu den in dieser Arbeit verwendeten Produktionsnetzen (Abbildung 1-4), die angeben, welche Zwischenergebnisse von Produktionen  $P_i$  analysiert und generiert werden.

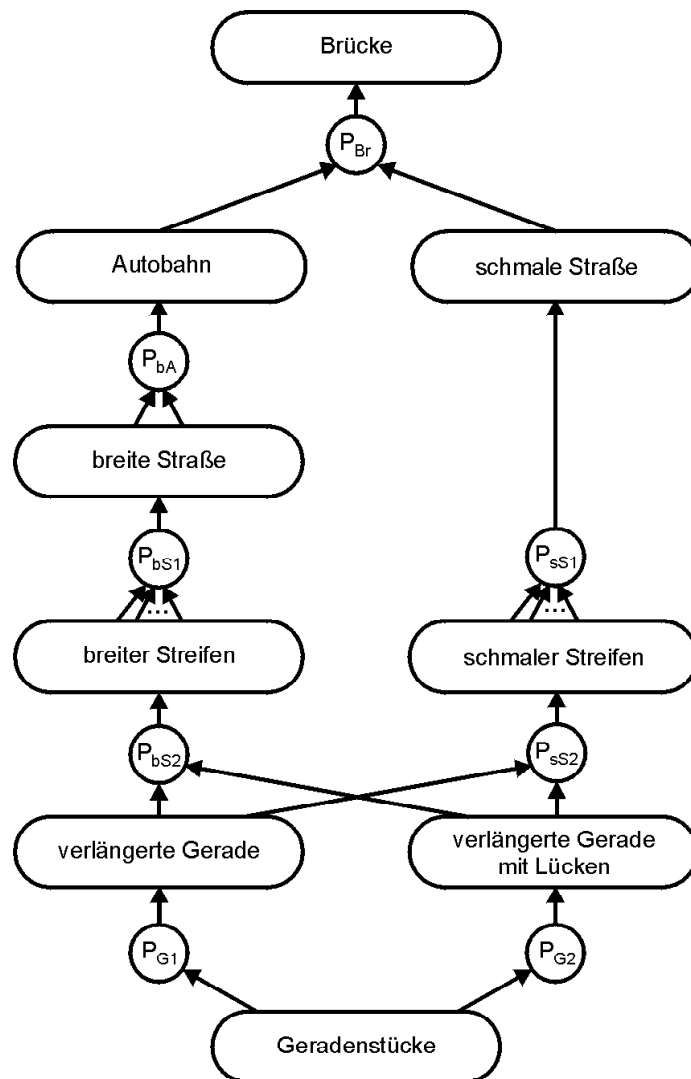


Abb. 1-4: Beispiel für ein Produktionsnetz

Die Produktionen dieser Netze geben jeweils für eine Menge ausgewählter Objekte, *Konzept-Repräsentanten* genannt, an, wie diese Menge auf einem höheren Abstraktionsniveau durch andere Objekte beschrieben werden kann und definieren so Assoziationen, die zu analysierenden Mengen von Instanzen neue Instanzen zu-

ordnen. Die in dieser Arbeit eingeführten Zuordnungssysteme beschreiben Folgen von Musteranalyseschritten mit Assoziationen. Zuordnungsautomaten definieren zusätzlich eine Ablaufkontrolle. Erzeugungs- und Erkennungsverfahren für Konfigurationen stellen den Bezug zur Musteranalyse und ihren speziellen Randbedingungen her.

Zuordnungssysteme, -automaten sowie Erzeugungs- und Erkennungsverfahren zur Analyse komplexer Muster, die als Mengen von Instanzen repräsentiert sind, sind neue Beiträge der vorliegenden Arbeit. Diese Systeme und Verfahren wurden mit dem Ziel entwickelt, Analyseprogrammen, die als Tafel- bzw. Blackboardsysteme (vgl. Abbildung 1-5) organisiert sind, eine formal abgesicherte Basis zu geben, so dass sie z. B. für zeitkritische Anwendungen auf Parallelrechnern implementiert werden können.

## 1.2 Überblick

Das erste Kapitel führt zuerst die strukturelle Bildanalyse mit Produktionsnetzen ein. Anschließend wird das der Arbeit zu Grunde liegende Problem vorgestellt. Dazu wird ein bei FGAN-FOM eingesetztes Tafelsystem für die Bildanalyse erläutert, für dessen Portierung auf Parallelrechner eine formale Beschreibung benötigt wird, die im zweiten Kapitel erfolgt.

Die formale Beschreibung im zentralen zweiten Kapitel definiert, aufbauend auf Transformations- bzw. Reduktionssystemen, als neue Beiträge Zuordnungssysteme und -automaten sowie darauf aufbauende Erzeugungs- und Erkennungsverfahren. Mit diesen Definitionen werden wichtige Eigenschaften der Analyse bewiesen, beispielsweise die Konfluenz bzw. bei terminierenden Folgen von Analyseschritten die Konvergenz der Analyse. Damit wird das erste Mal ein struktureller Ansatz zur Analyse komplexer zweidimensionaler Muster als Automat formuliert. Einleitend werden u. a. auch Thue-Systeme und Chomsky-Grammatiken für eindimensionale Zeichenketten aufgeführt.

Im dritten Kapitel schließt sich eine Diskussion mit Vergleichen und Aufwandsabschätzungen, die auch den Bezug zu allgemeinen Suchverfahren aufzeigen, an. Außerdem werden Konzepte zur Minimierung des Analyseaufwandes diskutiert. Die Abgrenzung zu anderen Ansätzen, insbesondere zu einer ebenfalls am FOM erstellten Arbeit über Koordinatengrammatiken, erfolgt im vierten Kapitel. Die Ausarbeitung schließt mit Vorschlägen für weiterführende Arbeiten.

Die Anhänge vertiefen spezielle sowie ergänzende Aspekte der Arbeit. Im ersten Teil des Anhangs wird anhand des Anwendungsbeispiels *bildgestützte Navigation* ein Eindruck von der betrachteten Bildanalyseaufgabe vermittelt. Hierzu werden auch Analyseergebnisse erläutert. Dazu gehören Ergebnisse von stark gestörten, z. B. mehrfachbelichteten Bildern. Im zweiten Teil wird eine detaillierte formale Darstellung der vorher nur verkürzt eingeführten Systeme, Automaten und Verfahren gegeben.

### 1.3 Ausgangspunkte: Ansätze und Systeme für die strukturelle Bildanalyse

Zur Organisation und Implementierung struktureller Musteranalysen werden in der Literatur u. a. Ansätze aus dem Bereich der sog. künstlichen Intelligenz beschrieben. Zu diesen Ansätzen gehören PANDEMONIUM und tafelbasierte Systeme, die anschließend erläutert werden und Ausgangspunkte der hier entwickelten Systeme und Verfahren sind. Syntaktische Bildanalyse-Verfahren, die ein weiterer Ausgangspunkt sind, werden in Abschnitt 4.1 dargestellt.

Im 4. Kapitel werden diese Ansätze sowie regelbasierte Systeme, semantische Netze und weitere Systeme zur strukturellen Musteranalyse mit dem in dieser Arbeit entwickelten System verglichen.

#### PANDEMONIUM-Ansatz

Einer der ersten Ansätze, mit denen Rechnersysteme Wahrnehmungsabläufe simulierten, war das von Selfridge vorgeschlagene PANDEMONIUM<sup>1</sup> /Sel58, And88, Bal82/, das als Ansatz für lernende Systeme entwickelt wurde. Nach Selfridge sollte man sich den Ablauf im PANDEMONIUM so vorstellen, dass für unterschiedliche Aufgaben jeweils eine unbegrenzte Anzahl von *Dämonen* zur Verfügung steht. Diese warten auf Kommandos, die ihnen von anderen Dämonen zugerufen werden. Hat ein Dämon genügend viele der von ihm erwarteten Kommandos mit ausreichender Intensität gehört, so ruft er sein Kommando weiteren Dämonen zu, die es interpretieren und mit den Zurufen anderer Dämonen verknüpfen. Dies beschreibt einen wesentlichen Aspekt der Analyse mit Assoziationsabläufen, wie sie mit Produktionsnetzen realisiert wird (vgl. auch Seite 89).

#### Tafelbasierte Systeme

Die Produktionen eines Erkennungsverfahrens werden im entwickelten Ansatz als prozedural formulierte Überprüfungsprogramme realisiert. Diese sind voneinander unabhängige Module, deren Zusammenspiel in einem größeren System zu organisieren ist. Einer der möglichen Organisationsrahmen für diese Aufgabenstellung ist das sogenannte *Tafel-* oder *Blackboard*-Konzept (vgl. auch Seite 90), das der im Abschnitt 1.4 erläuterten BPI-Architektur zu Grunde liegt.

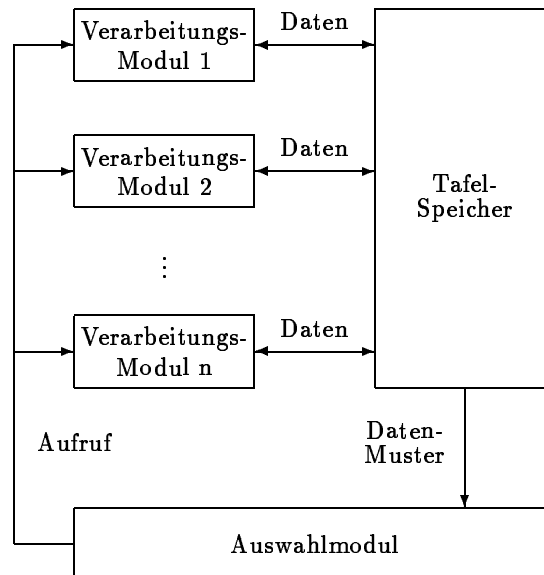
Tafel- und PANDEMONIUM-Konzept wurden anfänglich weitgehend identisch gesehen /New62/. Heute verbindet sich mit dem Begriff Tafelspeicher vor allem der in einem Rechnerprogramm von unabhängigen Modulen angesprochene globale Datenspeicher /Hrb85/. Diese Funktion definiert die Tafel als die gemeinsame Datenbasis eines Rechnersystems, auf die die autonomen Module dieses Rechnersystems ihre Ergebnisse schreiben und von der sie ihre Eingangsdaten lesen /Bar81, Win84, Bal82/. Newell /New62/ schrieb, dass man sich dies vorzustellen habe

---

<sup>1</sup>PANDEMONIUM: Reich der Dämonen



wie bei einer Sitzung, auf der für unterschiedliche Aufgaben eingesetzte *Experten* ihre vorläufigen Ergebnisse auf eine Tafel schreiben, so dass die Ergebnisse



**Abb. 1-5:** Struktur eines Tafelsystems

von anderen wahrgenommen und weiter bearbeitet werden können (Abbildung 1-5) /Sim66/. Außerdem gibt es einen *Moderator*, der bestimmt, wer als nächstes zur Tafel treten darf und als Auswahlmodul implementiert wird.

### Weitere Veröffentlichungen

Zusammenfassungen über wissensbasierte Ansätze werden in *Computational Intelligence* /Poo98/, *Mathematical Methods in Artificial Intelligence* /Ben96/, *Artificial Intelligence* /Win84/ sowie im *Handbook of Artificial Intelligence* /Bar81/ gegeben.

*Pattern Analysis and Understanding* /Nie89/ gibt eine umfassende Darstellung der Musteranalyse. In *Machine Learning and Image Interpretation* /Cae97/, *Modellgesteuerte Bildanalyse* /Bun85/, *Computer VISION* /Bal82/ und *Computer Vision Systems* /Han78/ werden wissensbasierte Systeme zur Bildauswertung dargestellt. Grundlagen und Ansätze der syntaktischen Bildanalyse werden in *Syntactic Pattern Recognition* /Gon82/, *Syntactic and Structural Pattern Recognition - Theory and Applications* - /Bun90/, *Visual Language Theory* /Mar98/ sowie in /Fu77/ und /Ros79/ gegeben. Binford /Bin82/ diskutiert einige der hier angesprochenen Systeme. In den Dissertationen von Shaheen /Sha79/, Nazif /Naz83/, Ohta /Oht85/, Sagerer /Sag85/, Michaelsen /Mich98a/, Quint /Qui97/ und Grau /Gra00/ werden weitere Systeme erläutert. *The Perception of the Visual World* /Gib50/ gibt u. a. eine Zusammenfassung der bis 1950 durchgeführten Untersuchungen zu Wahrnehmungsproblemen.

## 1.4 Motivation: Erweiterung der BPI-Architektur zur strukturellen Musteranalyse für Parallelrechner

In den 80-er Jahren wurde am FIM vom Autor die BPI-Architektur<sup>2</sup> zur strukturellen Mustererkennung entwickelt /Lüt86/. Die Architektur realisiert ein Tafel- bzw. Blackboard-System als wissensbasierten Ansatz. Bis heute fehlen für derartige Ansätze zur Analyse komplexer Muster formale Beschreibungen (z.B. als Automat), aus denen Eigenschaften wie die Konvergenz der Analyse abgeleitet werden können (vgl. Kapitel 4).

Zielsetzung war, der BPI-Architektur eine formal abgesicherte Basis zu geben, mit der die Konvergenz der Analyse bewiesen wird und die es z. B. ermöglicht, für zeitkritische Analysen benötigte Parallelrechner einzusetzen (vgl. Abschnitt 2.2.4) sowie explorative Analysen, bei denen die Datenerfassung parallel zur Analyse und aufbauend auf vorläufigen Analyseergebnissen erfolgt, durchzuführen.

Die formale Beschreibung erfolgt in dieser Arbeit mit Zuordnungssystemen bzw. -automaten sowie darauf aufbauenden Verfahren zur Erzeugung und Erkennung von Mustern, die neue Beiträge dieser Arbeit sind und im 2. Kapitel bzw. im Anhang B definiert werden. Die BPI-Architektur ist eine Realisierungsmöglichkeit von Zuordnungsautomaten und dient auch zur nicht formalen Einführung dieser Automaten.

Ergänzend zu den sich anschließenden Ausführungen dieses Abschnitts werden im Anhang A Ergebnisse, die für die bildgestützte Navigation mit einem BPI-basierten Analyseprogramm erzielt wurden, vorgestellt /Lüt86/.

Weitere Ergebnisse werden in /Füg92/, /Sti93/, /Füg94/, /Sti94/, /Bäh95/, /Sti95a/, /Sti95b/, /Sti96a/, /Sti96b/, /Sti97a/, /Sti97b/, /Sti97c/, /Mich98a/, /Mich98b/, /Schä98/, /Schw98/ und /Sti98/ erläutert.

### 1.4.1 Ablaufprinzip der BPI-Architektur

Der Ablaufzyklus des hier betrachteten Tafelsystems für die Bildanalyse wird in Abbildung 1-6 dargestellt. Vorverarbeitungsmodul generieren aus den Grauwertdaten des zu interpretierenden Bildes bewertete Primitivobjekte, z. B. *Geradenstücke* und *Kreissegmente*. Diese werden zur symbolischen Bearbeitung des Bildes als bewertete Elemente in eine prioritätsgeordnete Warteschlange eingetragen. Außerdem werden sie zusammen mit ihrer Beschreibung im Tafelspeicher abgelegt.

Nach der Generierung der Primitivobjekte werden diese zu komplexen Objekten, z. B. *Streifen*, zusammengesetzt. Aus diesen Teilobjekten werden weitere Teilobjekte, z. B. *Straßen*, zusammengesetzt, bis schließlich wiederzufindende Objekte, z. B. *Brücken*, aufgebaut sind.

Die Synthese eines komplexen Teilobjektes aus einem einfachen Teilobjekt läuft zweistufig ab. Zuerst werden ausgewählten Teilobjekten, die in der Warteschlange vorliegen, Hypothesen zugeordnet. Hierfür werden sog. Überprüfungsaufträge generiert. Beispielsweise werden für die im Abschnitt A.3 erläuterten Analysen einem

---

<sup>2</sup>BPI steht für *Blackboard Based Production System for Image Interpretation*.

*Geradenstück* die Hypothesen zugeordnet, dass das *Geradenstück* Teil eines *schmalen Streifens* und Teil eines *breiten Streifens* ist. Dann werden im zweiten Schritt ausgewählte Hypothesen überprüft, und falls sie bestätigt werden, werden die vermuteten komplexen Teilobjekte generiert, bewertet und abgespeichert.

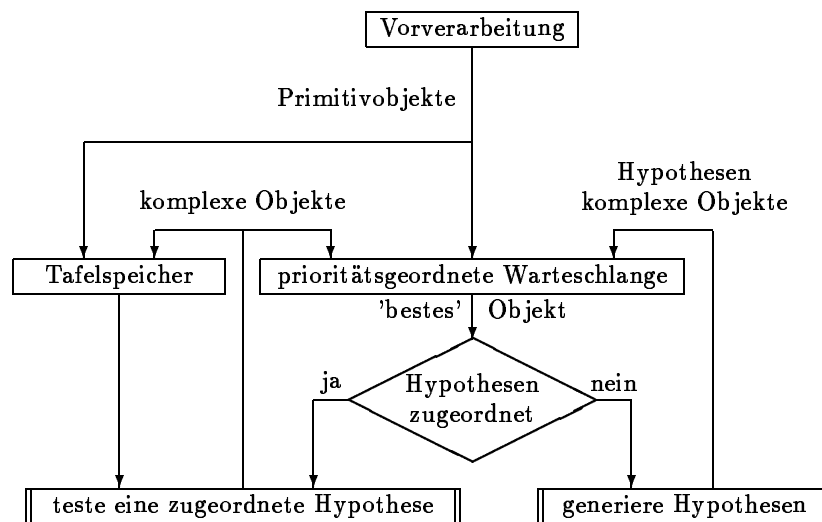


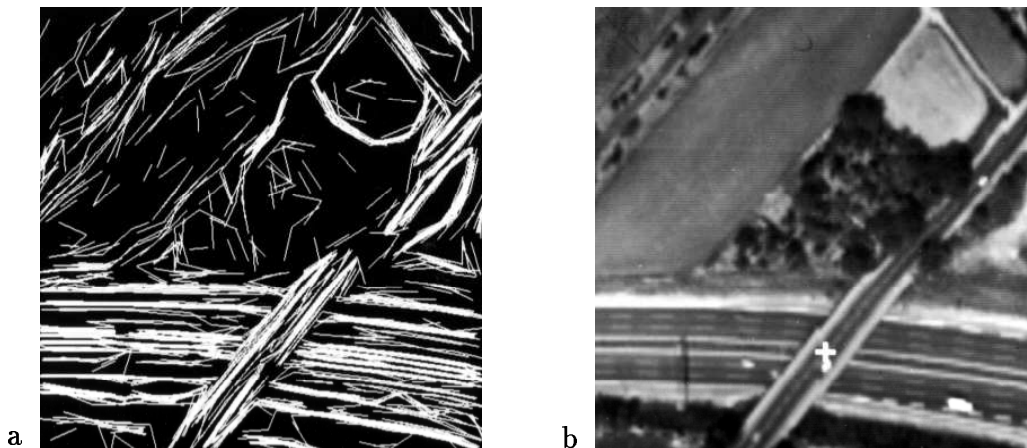
Abb. 1-6: Ablaufzyklus eines BPI-Analyseprogramms

Aus Sicht der Tafelsysteme sind die für jedes Objekt implementierten Analysemodule, die zugeordnete Hypothesen testen, die Experten des Tafelsystems. Die prioritätsgeordnete Warteschlange realisiert das Auswahlmodul bzw. den Moderator.

In der Warteschlange sind bewertete Objekte eingetragen, die zwei Zustände einnehmen können. Nach ihrer Generierung sind sie im Zustand *generiere Hypothesen*. Nachdem Hypothesen zugeordnet wurden, sind die Objekte im Zustand *überprüfe Hypothesen*. Unabhängig vom Zustand der Objekte wird aus der Warteschlange immer das derzeit bestbewertete Objekt ausgelesen und dann entsprechend seines Zustandes bearbeitet.

Ist das ausgelesene Objekt im Zustand *generiere Hypothesen*, so werden ihm Hypothesen zugeordnet, indem dem Objekt für jede Hypothese ein Überprüfungsprogramm zugeordnet wird, das die jeweilige Hypothese später überprüfen soll. Jede Hypothese erhält eine eigene Bewertung, die sich aus der Bewertung des zu expandierenden Objekts ableitet. Sind die zugeordneten Hypothesen unterschiedlich bewertet, so erhält das Objekt mehrere unterschiedliche Bewertungen. Ist das aus der Warteschlange ausgelesene Objekt im Zustand *überprüfe Hypothese*, so wird eines der dem Teilobjekt zugeordneten Überprüfungsprogramme zur Überprüfung einer Hypothese ausgeführt. Dabei greift dieses Programm zum Tafel-Speicher zu und testet zuerst, ob die zum Aufbau des zur Hypothese gehörenden komplexen Objektes benötigten Teilobjekte vorliegen. Das aus der Warteschlange ausgelesene Objekt im Zustand *überprüfe Hypothese* ist eines dieser Teilobjekte. Beispielsweise wird das Streifen-Überprüfungsprogramm nachprüfen, ob

rechts oder links vom ausgewählten, weiter zu analysierenden Geradenstück passende andere Geradenstücke bereits generiert worden sind. Dazu werden im Beispiel Typ, Abstand und Orientierungsunterschied überprüft. Falls passende Teilobjekte vorliegen, wird das vermutete Teilobjekt (z. B. ein Streifen) aufgebaut. Im anderen Fall wird die Hypothesenüberprüfung beendet. Das neue komplexe Teilobjekt wird dann bewertet und zur weiteren Bearbeitung in die Warteschlange und den Tafel-Speicher eingetragen. Die Bewertung kann bei Streifen anhand ihrer Breite und der Orientierungsunterschiede der sie begrenzenden Geradenstücke erfolgen.

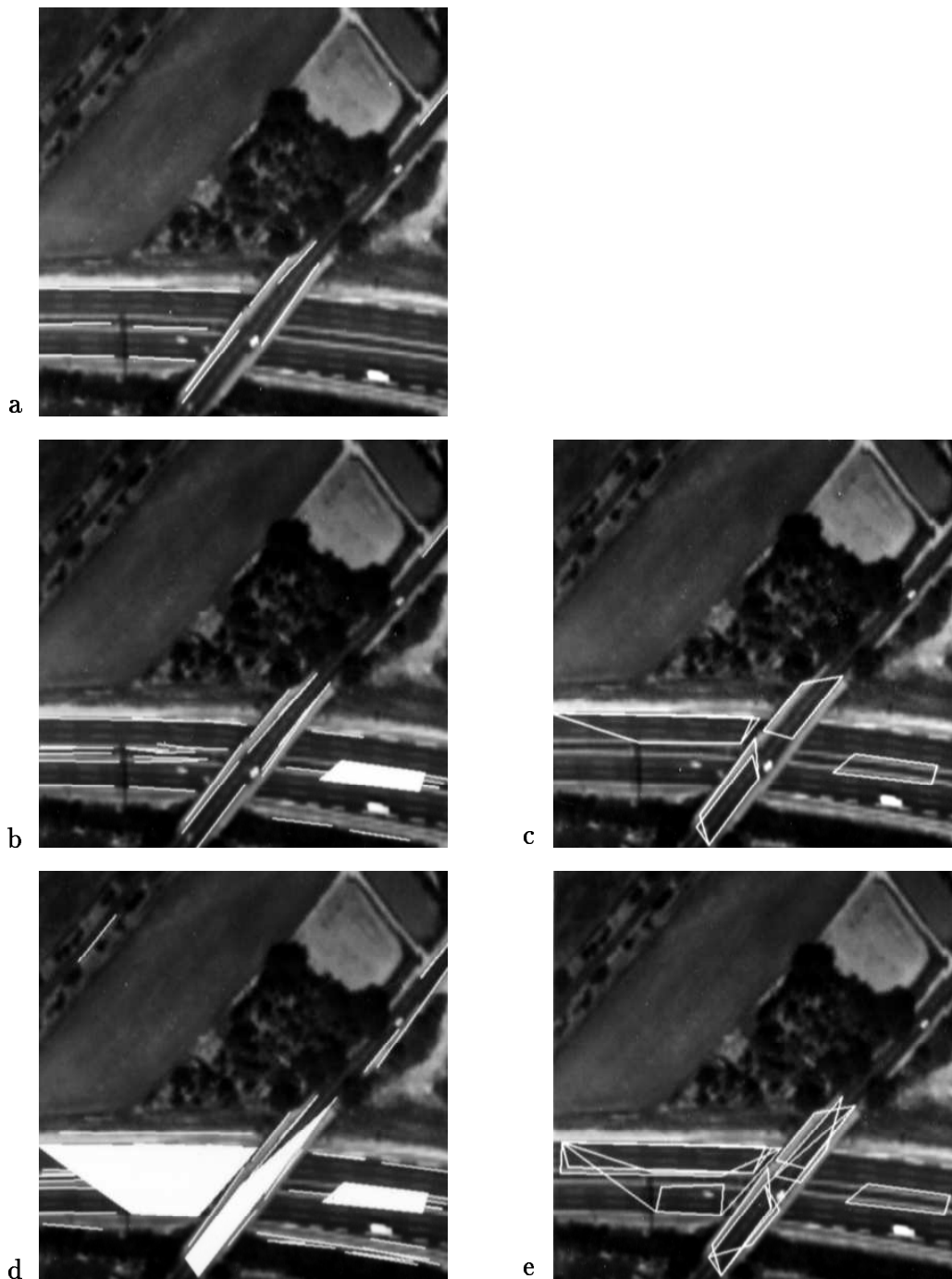


**Abb. 1-7:** Ein Anwendungsbeispiel zur Demonstration des Ablaufzyklus  
a) Primitivobjekte, b) markiertes Referenzobjekt

#### 1.4.2 Demonstration eines Analyseablaufs mit der BPI-Architektur

Mit dem BPI-Ansatz werden gut bewertete Teilobjekte zu immer komplexeren Teilobjekten zusammengesetzt, bis schließlich die wiederzufindenden Objekte aufgebaut sind. Da die Objekte entsprechend ihrer Bewertung ausgewählt werden, liegen nacheinander bearbeitete Objekte in der Regel in unterschiedlichen Bildbereichen. Es wird also nicht versucht, in einem lokal begrenzten Bildausschnitt solange weiterzuarbeiten, bis die dort vorliegende Information vollständig ausgewertet ist.

Die Abbildungen 1-7 und 1-8 demonstrieren diesen Ablauf anhand eines Beispiels und zeigen in zeitlicher Reihenfolge den sich weiter entwickelnden Analysezustand, indem die zu einem bestimmten Zeitpunkt im Tafel-Speicher abgespeicherten Teilobjekte dargestellt werden. Aus Darstellungsgründen sind die unterschiedlichen Objekttypen *Geradenstück* sowie *schmaler* und *breiter Streifen* in getrennten Abbildungen abgebildet.



**Abb. 1-8:** Zwischenergebnisse zu Abbildung 1-7  
links: verlängerte Geradenstücke und Straßen  
rechts: Streifen  
oben: nach 10% aller Zyklen  
Mitte: nach 50% aller Zyklen  
unten: nach 90% aller Zyklen

Die von der Vorverarbeitung generierten Geradenstücke sind in Abbildung 1-7a dargestellt, das schließlich wiedergefundene Brücken-Referenzobjekt in Abbildung 1-7b. Aus den Primitivobjekten wurden *verlängerte Geraden* (Abbildung 1-8a, b und d) generiert, aus denen wiederum *schmale* und *breite Streifen* (Abbildung 1-8c und e) und *schmale* und *breite Straßen* (Abbildung 1-8b und d) aufgebaut wurden. Die Abbildung 1-8a wurde nach 10% aller Überprüfungszyklen aufgenommen, die Abbildungen 1-8b und c) nach 50% aller Überprüfungszyklen und die Abbildungen 1-8d und e) nach 90% aller Überprüfungszyklen.

Die in diesem Abschnitt dargestellten Zwischenergebnisse sollen den Analyseablauf demonstrieren. Die eigentlichen Analyseergebnisse werden im Anhang Abschnitt A.3 erläutert.

## 2 Systeme und Verfahren für die Musteranalyse mit Produktionsnetzen

Dieses Kapitel stellt einen neuen strukturellen Musteranalyseansatz vor. Der Ansatz wird mit Zuordnungssystemen und -automaten modelliert, baut auf Reduktions- bzw. Transformationssystemen auf und ist Grundlage für Erzeugungs- und Erkennungsverfahren zur Analyse von Mustern, die als Mengen von Symbolen definiert sind. Der Ansatz wurde entwickelt, um der im Abschnitt 1.4 dargestellten BPI-Architektur eine formal abgesicherte Basis zu geben und für zeitkritische Analysen notwendige Parallelrechner einsetzen zu können. Außerdem sollten Grundlagen für explorative Analysen, bei denen die Datenerfassung parallel zur Analyse und aufbauend auf vorläufigen Analyseergebnissen erfolgt, bereitgestellt werden.

Nach einem einleitenden Überblick werden Transformationssysteme, Zuordnungssysteme und -automaten angegeben. Zusätzlich werden aus der Literatur bekannte Thue-Systeme und Chomsky-Grammatiken zur Manipulation von Zeichenketten aufgeführt. Danach werden Erzeugungs- und Erkennungsverfahren für Mengen von Symbolen definiert und ihr Einsatz zur strukturellen Musteranalyse diskutiert. Außerdem werden im Anhang B Aussagen zur Analyse mit Zuordnungssystemen und -automaten bewiesen.

In diesem Kapitel werden zur besseren Übersichtlichkeit einige formale Darstellungen verkürzt. Im Anhang sind die Definitionen, Sätze und Beweise detailliert aufgeführt. Neue Beiträge werden in den Abschnitten 2.2.3, 2.2.4, 2.3.2, 2.4, in Kapitel 3 und im Anhang B dargestellt. Die anderen Abschnitte erläutern aus der Literatur bekannte Grundlagen (Transformationssysteme, Systeme und Verfahren zur Zeichenkettenverarbeitung) oder geben Überblicke und Beispiele.

### 2.1 Einleitender Überblick

*Zuordnungssysteme* und *Zuordnungsautomaten* definieren die strukturelle Musteranalyse mit Produktionsnetzen und sind zentrale neue Beiträge der vorliegenden Arbeit. Aus ihnen werden Erzeugungs- und Erkennungsverfahren abgeleitet.

Zuordnungssysteme und -automaten lassen sich als abstrakte Transformationssysteme interpretieren, die in der Informatik als Reduktionssysteme in den unterschiedlichsten Formen und Zusammenhängen verwendet werden, z. B. für syntaktische Bildanalyse-Verfahren /Gon78/ sowie Übersetzer von Programmiersprachen /Jan88/.

Aufbauend auf Zuordnungssystemen werden Erkennungsverfahren definiert. Die zur Erkennung verwendeten Modelle sowie die bearbeiteten Aufgabenstellungen können mit Erzeugungsverfahren, die die bei der Erkennung betrachteten Muster erzeugen, veranschaulicht werden.

Eine Analyse beginnt in der Regel damit, dass Vorverarbeitungsprogramme aus Sensordaten (z. B. Bildern) Primitivobjekte extrahieren. Diese ersten symbolisch beschriebenen Einheiten werden wie alle weiteren Zwischen- und Endergebnisse

*Instanzen* genannt. Instanzen-Mengen heißen *Konfigurationen*. Die Regeln, mit denen Instanzen durch andere Instanzen beschrieben und so analysiert werden, heißen *Assoziationen*.

*Zuordnungssysteme mit Konzepten* definieren die bei der Analyse der Instanzen verwendeten Assoziationen durch *Produktionen*, die für *Konzept-Repräsentanten* definiert sind, und durch *Instanzierungsfunktionen*. Konzept-Repräsentanten sind dabei spezielle, geeignet ausgewählte Instanzen und Produktionen spezielle Assoziationen.

*Zuordnungssysteme mit Ableitungen* definieren Instanzen als Elemente des Kartesischen Produktes endlicher Merkmalmengen mit einer Menge von Vorgängerinstanzen. Diese Instanzen heißen *Ableitungen*. Für Zuordnungssysteme mit Ableitungen werden Produktionsnetze, Ableitungsgraphen und Beschreibungsgraphen definiert. Produktionsnetze beschreiben die Modelle, Ableitungsgraphen die Verarbeitungsabläufe und Beschreibungsgraphen die Verarbeitungsergebnisse. Auswahl-, Ablauf- und Bewertungsfragen werden bei Zuordnungssystemen nicht betrachtet.

Für eine Ablaufkontrolle werden *Zuordnungsautomaten* mit Zuständen und Zustandsübergängen definiert. Der Einsatz von Parallelrechnern und Analyseanwendungen, bei denen die Datenerfassung parallel mit der Analyse erfolgt und von ihr gesteuert wird, wird mit speziellen Relationen beschrieben.

Beweise zur Analyse mit Parallelrechnern und zur Analyse in der Schleife mit der Datenerfassung, bei der die Datenerfassung auf vorläufigen Analyseergebnissen aufbaut, werden mit Zuordnungsautomaten formuliert. Zuordnungssysteme dienen dazu, Eigenschaften von Zuordnungsautomaten, die unabhängig von der Ablaufkontrolle der Zuordnungsautomaten sind, zu formulieren und zu beweisen.

Die im Abschnitt 1.4 erläuterte BPI-Architektur ist eine spezielle Implementierung von Zuordnungsautomaten. Als Rechnerprogramme implementierte Zuordnungsautomaten lassen sich als *Tafel-* bzw. *Blackboardsysteme* (vgl. Abschnitt 1.3) beschreiben.

## 2.2 Transformationssysteme

Zuordnungssysteme können als eine Abstraktion spezieller Tafelsysteme aufgefasst werden. Sie sind keine syntaktischen, d. h. auf der Zeichenkettenverarbeitung aufbauende Ansätze, lassen sich aber wie die syntaktischen Ansätze als *abstrakte Transformationssysteme* /Ave95, Jan88, Boo93/ beschreiben.

Abstrakte Transformationssysteme werden auch *abstrakte Reduktionssysteme* genannt. Der Begriff Reduktionssystem rührt daher, dass diese Systeme zuerst mit dem Ziel definiert wurden, gegebene Zeichenketten so umzuformen, dass sie anschließend vereinfacht, d. h. reduziert, vorliegen. Da in dieser Arbeit Transformationssysteme auch zur Definition expandierender Systeme verwendet werden, wird der Begriff Reduktionssystem vermieden.



### 2.2.1 Abstrakte Transformationssysteme

Abstrakte Transformationssysteme geben, aufbauend auf einer Menge  $B$  (z. B. von Zwischenergebnissen einer Bildanalyse) und einer Relation  $\Rightarrow$  in der Menge  $B$ , an, wie eine Folge von Schritten (z. B. in einer Verarbeitungsschleife) abläuft. Die Relation  $\Rightarrow$  definiert dabei einzelne Transformationsschritte<sup>1</sup> (z. B. Bildanalyse-schritte).

Ein *abstraktes Transformationssystem* ist ein Paar  $(B, \Rightarrow)$  mit /Ave95/:

$$\begin{aligned}
 & \Rightarrow \subseteq B \times B \\
 & \xrightarrow{0} := \{(b, b) \mid b \in B\} \\
 & \xrightarrow{i+1} := \left\{ (a, c) \in B \times B \mid \begin{array}{l} \exists b \in B : \\ (a, b) \in \xrightarrow{i} \\ (b, c) \in \Rightarrow \end{array} \right\}, i \in \mathbb{N}_0 \\
 & \xrightarrow{+} := \bigcup_{i \in \mathbb{N}} \xrightarrow{i} \\
 & \xrightarrow{*} := \xrightarrow{+} \cup \xrightarrow{0} \\
 & \xrightarrow{\epsilon} := \Rightarrow \cup \xrightarrow{0}
 \end{aligned} \tag{2-1}$$

Die Normalformen bezgl. der Relation  $\Rightarrow$  sind diejenigen Elemente der Menge  $B$ , zu denen es keine (weiteren) Transformationsschritte in  $\Rightarrow$  gibt (/Ave95/ Seite 10).

Wenn ausgehend von einem Element  $r \in B$  über zwei Zwischenergebnisse  $u \in B$  und  $v \in B$  ein (einziges) Ergebnis  $s \in B$  erreicht werden kann, heißt die Relation  $\Rightarrow$  *konfluent* (Abbildung 2-1). Formal gilt: die Relation  $\Rightarrow$  heißt *konfluent*, falls

$$\begin{aligned}
 & \forall r, u, v \in B : \\
 & \left. \begin{array}{l} r \xrightarrow{*} u \\ r \xrightarrow{*} v \end{array} \right\} \rightarrow \left\{ \begin{array}{l} \exists s \in B : \\ u \xrightarrow{*} s \\ v \xrightarrow{*} s \end{array} \right. \tag{2-2}
 \end{aligned}$$

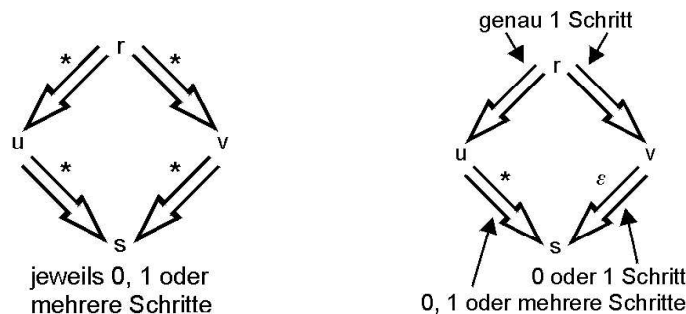
gilt (/Ave95/ Definition 1.2.1).

Wenn ausgehend von einem Element  $r \in B$  über zwei mit jeweils einem direkten Transformationsschritt zu erreichende Zwischenergebnisse  $u \in B$  und  $v \in B$  ein (einziges) Ergebnis  $s \in B$  erreicht werden kann und dabei von  $u$  oder von  $v$  nicht mehr als ein direkter Transformationsschritt benötigt wird, so heißt die Relation  $\Rightarrow$  *streng konfluent*. Formal gilt: die Relation  $\Rightarrow$  heißt *streng konfluent*, falls

$$\begin{aligned}
 & \forall r, u, v \in B : \\
 & \left. \begin{array}{l} r \xRightarrow{} u \\ r \xRightarrow{} v \end{array} \right\} \rightarrow \left\{ \begin{array}{l} \exists s \in B : \\ u \xrightarrow{*} s \\ v \xrightarrow{\epsilon} s \end{array} \right. \tag{2-3}
 \end{aligned}$$

gilt (/Ave95/ Definition 1.2.10). *Strenge Konfluenz* ist, wenn sie zutrifft, in der Regel leichter nachzuprüfen als Konfluenz und impliziert Konfluenz (/Ave95/ Satz 1.2.11).

<sup>1</sup>In /Jan88/ Definition 1.1.1 (Seite 7) wird  $\Rightarrow$  als die Transformationsrelation eines Reduktionssystems bezeichnet. Dementsprechend wird hier von einem Transformationssystem gesprochen.



**Abb. 2-1:** Konfluente Transformationssysteme  
links: Konfluenz; rechts: strenge Konfluenz

Die Relation  $\Rightarrow$  heißt *terminierend*, wenn es keine unendliche Folge  $b_1 \Rightarrow b_2 \Rightarrow \dots \Rightarrow b_n \Rightarrow \dots$  gibt (/Ave95/ Definition 1.1.3).

Die Relation  $\Rightarrow$  heißt *konvergent*, wenn sie konfluent und terminierend ist (/Ave95/ Definition 1.2.7). In diesem Fall gibt es eindeutige Normalformen (/Ave95/ Lemma 1.2.6).

Eine der zentralen Fragestellungen, die mit Reduktions- bzw. Transformationssystemen bearbeitet wird, ist, ob ein Verfahren eindeutige Lösungen erzeugt, ob es also konvergent ist. Für die in dieser Arbeit definierten Zuordnungssysteme, bei denen die Elemente von  $B$  Mengen sind, wird gezeigt, dass sie streng konfluent (Satz B.11), d. h. konfluent und bei endlichen zugeordneten Mengen  $s \in B$  terminierend (Korollar B.5), d. h. konvergent sind. Ohne auf die Konfluenzeigenschaft zurückzugreifen, wird zusätzlich für Zuordnungssysteme und Zuordnungsautomaten gezeigt, dass die Normalformen bei endlichen zugeordneten Mengen eindeutig sind (Satz B.9, Korollar B.62) und für diesen Fall eine Analysefunktion berechnet wird (Korollar B.63).

## 2.2.2 Beispiele für Transformationssysteme

Transformationssysteme sind ein sehr allgemeiner Ansatz, der als Reduktionssystem in der Informatik in den unterschiedlichsten Formen und Zusammenhängen verwendet wird. Beispielsweise lassen sich Zeichenketten-Ersetzungssysteme, die anschließend angegeben werden, und darauf aufbauend Übersetzer für Programmiersprachen /Aho72, Sal78/ mit Transformationssystemen beschreiben.

### Zeichenketten-Ersetzungssysteme

Syntaktische Bildanalyseverfahren (vgl. Abschnitt 4.1) basieren auf Verfahren zur Manipulation von Zeichenketten, die anschließend erläutert werden.

Zum Ersetzen von Zeichenketten werden Transformationssysteme  $(W^*(V), \xrightarrow{s})$  definiert /Sal78/, bei denen  $W^*(V)$  die Menge aller Zeichenketten ist, die mit dem Vokabular  $V$  gebildet werden kann. Dabei wird die Konkatenation (d. h. das

Hintereinanderschreiben) als assoziative Verknüpfung verwendet, wobei die Konkatenation mit dem leeren Zeichen  $\lambda$  eine Zeichenkette nicht verändert. In  $W^+(V)$  fehlt die leere Zeichenkette  $\lambda$ .

$W^*(V)$  und  $\xRightarrow[S]{}$  sowie  $W^+(V)$  sind wie folgt definiert /Sal78/:

$$\begin{aligned} W^0(V) &:= \{\lambda\} \\ W^j(V) &:= \{v_1v_2 \cdots v_j \mid v_i \in V, i = 1, 2, \dots, j\}, j \in \mathbb{N} \\ W^+(V) &:= \bigcup_{j \in \mathbb{N}} W^j(V) \\ W^*(V) &:= W^+(V) \cup W^0(V) \\ \forall s \in W^*(V) &: \lambda s = s = s \lambda \\ \xRightarrow[S]{} &\subseteq W^*(V) \times W^*(V) \end{aligned} \quad (2-4)$$

**Thue-Systeme** Zeichenketten werden mit Thue- (TS) /Thu14/ und Semi-Thue-Systemen (STS) verändert, die Mengen von Produktionen<sup>2</sup>

$$P \subseteq W^*(V) \times W^*(V) \quad (2-5)$$

sind /Jan88, Ave95/. Mit der Transformationsrelation  $\xrightarrow[TS]{(V,P)}$  bzw.  $\xrightarrow[STS]{(V,P)}$

$$\begin{aligned} \xrightarrow[TS]{(V,P)} &:= \left\{ (pxq, pyq) \mid \begin{array}{l} p, q \in W^*(V) \\ (x, y) \in P \vee (y, x) \in P \end{array} \right\} \\ \xrightarrow[STS]{(V,P)} &:= \left\{ (pxq, pyq) \mid \begin{array}{l} p, q \in W^*(V) \\ (x, y) \in P \end{array} \right\} \end{aligned} \quad (2-6)$$

ist das Paar  $\left( W^*(V), \xrightarrow[TS]{(V,P)} \right)$  bzw.  $\left( W^*(V), \xrightarrow[STS]{(V,P)} \right)$  ein Transformationssystem für Thue- bzw. Semi-Thue-Systeme.

### 2.2.3 Transformationssysteme für Konfigurationen

In der Literatur angegebene syntaktische Bildanalyse-Verfahren (vgl. Kapitel 4) manipulieren in der Regel spezielle Einheiten (z. B. Zeichenketten). Dabei wird jeweils nur eine spezifische Einbettung<sup>3</sup> (z. B. die Konkatenation von Zeichenketten) betrachtet.

Eine Alternative ist, (ungeordnete) Mengen zu transformieren und mit den Transformationen beliebige Einbettungen zu definieren. Diese Mengen heißen *Konfigurationen*. Die Transformationsvorschriften werden dann umfangreicher und spezifizieren sowohl die beabsichtigten Veränderungen als auch die einzuhaltende Einbettung.

<sup>2</sup>/Sal78/ und /Ste88/ bezeichnen das Paar  $(V, P)$ , das /Sal78/ Ersetzungssystem und /Ste88/ Regel- oder Produktionssystem nennt, als STS.

<sup>3</sup>Die Einbettung definiert den Mechanismus, der Teilmuster zu übergeordneten Mustern zusammenfasst und hängt von der gewählten Repräsentation der Instanzen ab (z. B. Zeichenkette, Graph, NAPE /Gon78/).

Die in diesem Abschnitt definierten Relationen schaffen eine neue Basis für die strukturelle Musteranalyse mit Produktionsnetzen. Die Relationen *parallel*, *verteilt* und *asynchron zuordnen* bieten z. B. eine Grundlage für die Analyse mit Parallelrechnern, die für zeitkritische Analysen benötigt werden.

### Systeme zur Ersetzung von Konfigurationen

Für die im Abschnitt 2.3.2 definierten Erzeugungs- und ersetzenden Erkennungsverfahren lässt sich ein Transformationssystem  $(\wp(\mathcal{I}), \xrightarrow[e]{\mathbf{A}})$  definieren, mit dem Teilmengen ersetzt werden. Bei einem Ersetzungsschritt mit einer Assoziation  $(\mathcal{X}, \mathcal{Y}) \in \mathbf{A}$  müssen alle Instanzen von  $\mathcal{X}$  vorliegen. Diese werden dann aus der Ausgangskonfiguration  $\mathcal{R}$  entfernt und durch die Instanzen von  $\mathcal{Y}$  ersetzt (Abbildung 2-2 oben).

Mit der Menge der Assoziationen  $\mathbf{A} \subseteq \wp(\mathcal{I}) \times \wp(\mathcal{I})$  und der Relation *ersetzt direkt eine Konfiguration*  $\xrightarrow[e]{\mathbf{A}}$

$$\xrightarrow[e]{\mathbf{A}} := \left\{ (\mathcal{R}, \mathcal{S}) \in \wp(\mathcal{I}) \times \wp(\mathcal{I}) \mid \begin{array}{l} \exists (\mathcal{X}, \mathcal{Y}) \in \mathbf{A} : \\ \mathcal{X} \subseteq \mathcal{R} \\ \mathcal{S} = (\mathcal{R} \setminus \mathcal{X}) \cup \mathcal{Y} \end{array} \right\} \quad (2-7)$$

wird für Transformationssysteme  $(\wp(\mathcal{I}), \xrightarrow[e]{\mathbf{A}})$  die Relation *ersetzt eine Konfiguration*  $\xrightarrow[e]{*\mathbf{A}}$  des Konfigurationen-Ersetzungssystems  $(\mathcal{I}, \mathbf{A})$  definiert (Gl. 2-1). Den Größen  $B$  und  $\Rightarrow$  aus Gleichung 2-1 entsprechen dabei  $\wp(\mathcal{I})$  und  $\xrightarrow[e]{\mathbf{A}}$ .

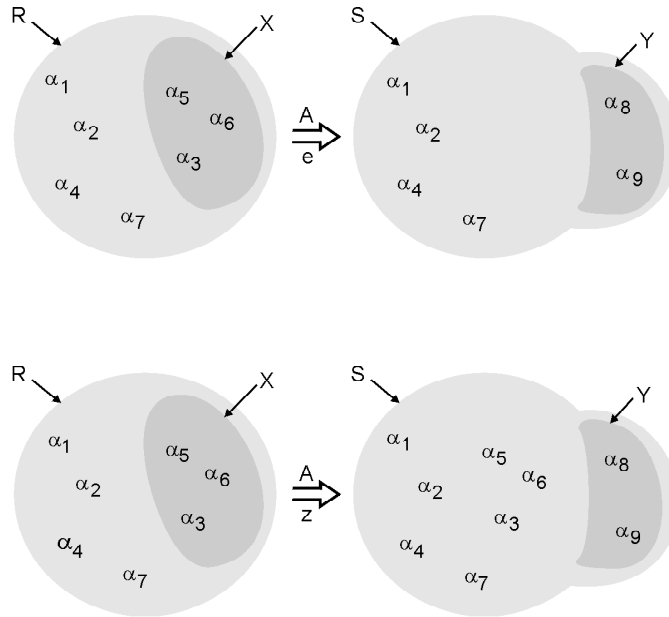
### Systeme zur Zuordnung von Konfigurationen

Für zuordnende Erkennungsverfahren mit Konfigurationen werden Zuordnungssysteme und Zuordnungsautomaten eingeführt. Bei einem Zuordnungsschritt mit einer Assoziation  $(\mathcal{X}, \mathcal{Y}) \in \mathbf{A}$  müssen alle Instanzen von  $\mathcal{X}$  vorliegen und die Instanzen von  $\mathcal{Y}$  zur Ausgangskonfiguration  $\mathcal{R}$  hinzugefügt werden (Abbildung 2-2 unten). Außerdem darf  $\mathcal{Y}$  noch nicht vorliegen.

Mit der Menge der Assoziationen  $\mathbf{A} \subseteq \wp(\mathcal{I}) \times \wp(\mathcal{I})$  und der Relation *ordnet direkt zu*  $\xrightarrow[z]{\mathbf{A}}$

$$\xrightarrow[z]{\mathbf{A}} := \left\{ (\mathcal{R}, \mathcal{S}) \in \wp(\mathcal{I}) \times \wp(\mathcal{I}) \mid \begin{array}{l} \exists (\mathcal{X}, \mathcal{Y}) \in \mathbf{A} : \\ \mathcal{X} \subseteq \mathcal{R} \\ \mathcal{Y} \not\subseteq \mathcal{R} \\ \mathcal{S} = \mathcal{R} \cup \mathcal{Y} \end{array} \right\} \quad (2-8)$$

wird für Transformationssysteme  $(\wp(\mathcal{I}), \xrightarrow[z]{\mathbf{A}})$  die Relation *ordnet zu*  $\xrightarrow[z]{*\mathbf{A}}$  des Zuordnungssystems  $(\mathcal{I}, \mathbf{A})$  definiert (Gl. 2-1, B-3, B-4). Hier wird mit  $\mathcal{S} = \mathcal{R} \cup \mathcal{Y}$



**Abb. 2-2:** Ersetzen und zuordnen von Konfigurationen  
 oben: Ersetzung; unten: Zuordnung  
 mit der Assoziation  $(\mathcal{X} = \{\alpha_3, \alpha_5, \alpha_6\}, \mathcal{Y} = \{\alpha_8, \alpha_9\})$

die vorliegende Menge  $\mathcal{R}$  ergänzt, während in Gleichung 2-7 mit  $\mathcal{S} = (\mathcal{R} \setminus \mathcal{X}) \cup \mathcal{Y}$  ersetzt wird (Abbildung 2-2). Abschnitt B.1 definiert Zuordnungssysteme. Die in Gleichung 2-8 verwendete Einschränkung  $\mathcal{Y} \not\subseteq \mathcal{R}$  stellt sicher, dass Folgen von Zuordnungen terminieren. Wegen der Definition mehrfach assoziierender Assoziationen (Gl. B-72) fällt diese Bedingung bei Ersetzungen (Gl. 2-7) weg.

**Abschließende Zuordnungen** Wenn für die Analyse einer Konfiguration  $\mathcal{R}$  alle mit einer Menge von Assoziationen  $\mathbf{A}$  möglichen Zuordnungsschritte durchgeführt wurden, liegt das Analyseergebnis vor. Die Relation *ordnet abschließend* zu  $\xrightarrow[z]{\dagger \mathbf{A}}$

$$\xrightarrow[z]{\dagger \mathbf{A}} := \left\{ (\mathcal{R}, \mathcal{S}) \in \xrightarrow[z]{* \mathbf{A}} \mid \forall (\mathcal{U}, \mathcal{V}) \in \xrightarrow[z]{\mathbf{A}} : \mathcal{S} \neq \mathcal{U} \right\} \quad (2-9)$$

beschreibt dies (Gl. B-5) und definiert für terminierende Zuordnungen die Analysefunktion  $AF$ , für die  $\mathcal{R} \xrightarrow[z]{\dagger \mathbf{A}} AF(\mathcal{R})$  gilt (Satz B.9, Korollar B.10).  $\xrightarrow[z]{\dagger \mathbf{A}}$  definiert auch die Normalformen von  $\xrightarrow[z]{\mathbf{A}}$ :

$$NF \xrightarrow[z]{\mathbf{A}} := \left\{ \mathcal{S} \mid \exists \mathcal{R} \subseteq \mathcal{I} : \mathcal{R} \xrightarrow[z]{\dagger \mathbf{A}} \mathcal{S} \right\}$$

**Parallele Zuordnungen** Die Zuordnungsschritte mit einer Menge von Assoziationen  $\mathbf{A}$  können nacheinander oder parallel erfolgen. Ausgehend von der Zielsetzung, mit  $n$  Prozessoren in  $n$  Strängen Analysen parallel auszuführen, wird die Relation *parallel zuordnen* formuliert.

Die Relation *direkt parallel zuordnen*  $\xrightarrow{\mathbf{A}}_{\parallel z}$

$$\xrightarrow{\mathbf{A}}_{z(i+1)} := \left\{ (\mathcal{R}, \mathcal{S}) \in \wp(\mathcal{I}) \times \wp(\mathcal{I}) \left| \begin{array}{l} \exists(\mathcal{R}, \mathcal{T}) \in \xrightarrow{\mathbf{A}}_{z(i)} : \\ \exists(\mathcal{R}, \mathcal{U}) \in \xrightarrow{* \mathbf{A}}_z : \\ \mathcal{S} = \mathcal{T} \cup \mathcal{U} \end{array} \right. \right\}, i \in \mathbb{N} \quad (2-10)$$

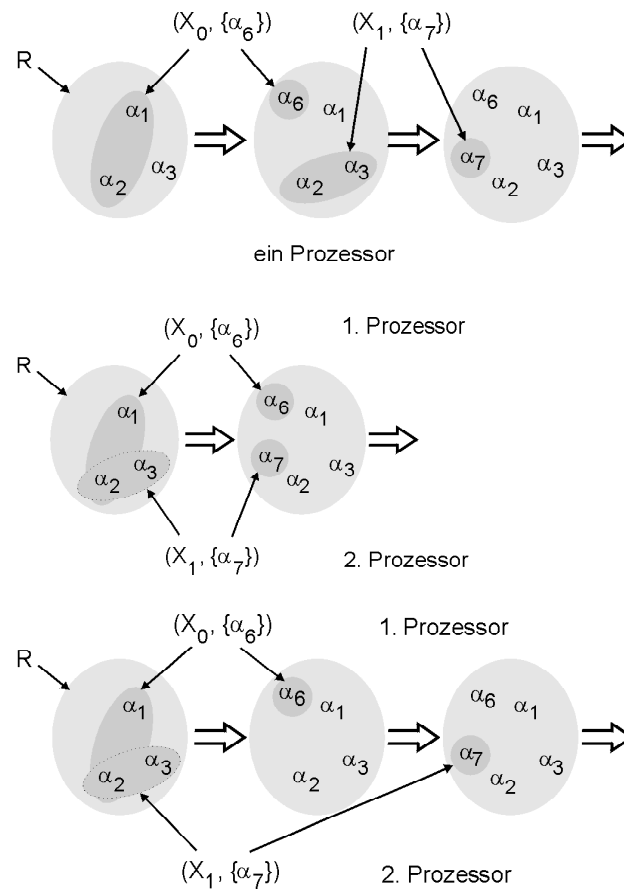
$$\xrightarrow{\mathbf{A}}_{\parallel z} := \bigcup_{i \in \mathbb{N}} \xrightarrow{\mathbf{A}}_{z(i)}$$

definiert die *parallele Zuordnung*  $\xrightarrow{* \mathbf{A}}_{\parallel z}$  (Gl. B-19, B-20), die mit der Relation *zuordnen* übereinstimmt  $\left( \xrightarrow{* \mathbf{A}}_{\parallel z} = \xrightarrow{* \mathbf{A}}_z \right)$  (Satz B.19).

**Asynchrone Zuordnungen** Die einzelnen Zuordnungsschritte in parallel ablaufenden Analysesträngen können asynchron ablaufen. Diese asynchronen Zuordnungen der Relation  $\xrightarrow{* \mathbf{A}}_{a \parallel z}$  (Gl. B-24) erlauben, in jedem Strang mit seinem zugeordneten Prozessor nach Überprüfung einer Assoziation sofort die nächste Assoziation zu überprüfen und neue Instanzen zu generieren. Dabei wird geprüft, ob die Konfiguration  $\mathcal{X}$  einer Assoziation  $(\mathcal{X}, \mathcal{Y}) \in \mathbf{A}$  in einer aus der zu analysierenden Konfiguration  $\mathcal{R}$  abgeleiteten Konfiguration, z. B.  $\mathcal{Q}_1$  mit  $\mathcal{R} \subseteq \mathcal{Q}_1$ , vorliegt ( $\mathcal{X} \subseteq \mathcal{Q}_1$ ) und danach, falls sie vorliegt, eine gegebenenfalls erweiterte Konfiguration, z. B.  $\mathcal{Q}_2$  mit  $\mathcal{Q}_1 \subseteq \mathcal{Q}_2$ , zu der das jeweilige Zwischenergebnis darstellenden Konfiguration  $\mathcal{S}$  ergänzt ( $\mathcal{S} = \mathcal{Q}_2 \cup \mathcal{Y}$ ).  $\mathcal{Q}_2$  soll dabei aus  $\mathcal{Q}_1$  entstanden sein durch Zuordnungen in anderen parallel ablaufenden Analysesträngen<sup>4</sup>, die  $\mathcal{Q}_1$  um rechte Konfigurationen anderer Assoziationen zu  $\mathcal{Q}_2$  ergänzen. Abbildung 2-3 vergleicht diesen Ablauf mit der parallelen Zuordnung. Die Relation *asynchron zuordnen* stimmt mit der Relation *zuordnen* überein (Satz B.24)  $\left( \xrightarrow{* \mathbf{A}}_{a \parallel z} = \xrightarrow{* \mathbf{A}}_z \right)$  und ist Grundlage für die parallele Analyse mit mehreren Prozessoren, die einen gemeinsamen Speicher verwenden (Abbildung 2-6 und Satz B.69).

**Verteilte Zuordnungen** Eine Verallgemeinerung der Relation *parallel zuordnen*  $\xrightarrow{* \mathbf{A}}_{\parallel z}$ , die in jeder der parallelen Analysestränge von der gleichen Konfiguration ausgeht ( $\mathcal{R}$  in Gl. 2-10), ist die Relation *verteilt zuordnen*  $\xrightarrow{\mathbf{A}}_{v \parallel z}$   $\left( \xrightarrow{\mathbf{A}}_{v \parallel z} = \right)$

<sup>4</sup>Der Sonderfall  $\mathcal{Q}_2 = \mathcal{Q}_1$  ist auch zulässig.



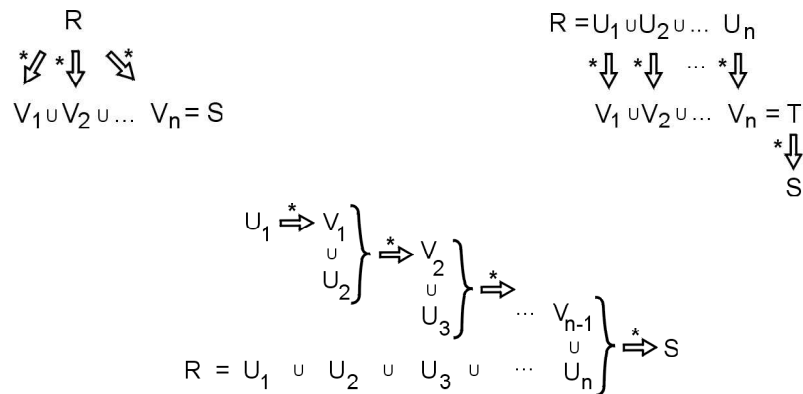
**Abb. 2-3:** Parallel und asynchron zuordnen  
 mit den Assoziationen  $(\mathcal{X}_0 = \{\alpha_1, \alpha_2\}, \{\alpha_6\})$  und  $(\mathcal{X}_1 = \{\alpha_2, \alpha_3\}, \{\alpha_7\})$   
 oben: (2 serielle) Zuordnungen  
 Mitte: parallel zuordnen,  
 unten: asynchron zuordnen

$\left\{ (\mathcal{R}, W, \mathcal{S}) \mid (\mathcal{R}, \mathcal{S}) \in \mathcal{R} \mid_{\frac{v||Z}{W}} \mathcal{S} \right\}$  Gleichungen B-22). Verteilt zuordnen wird für Zuordnungen mit mehreren parallelen Analysesträngen, die jeweils unterschiedliche Konfigurationen mit unterschiedlichen Assoziationen analysieren, eingesetzt (Abbildung 2-4). Die Vereinigung der parallel analysierten Konfigurationen ist die insgesamt zu analysierende Konfiguration. Für jeden der  $n$  Analysestränge wird eine Menge von Assoziationen  $\mathbf{B}_i \subseteq \mathbf{A}; i = 1, 2, \dots, n$  in einem Tupel  $W = (\mathbf{B}_1, \mathbf{B}_2, \dots, \mathbf{B}_n)$  vermerkt. Die Relation  $\xrightarrow[\frac{v||z}{A}]{}$  enthält Tripel  $(\mathcal{R}, W, \mathcal{S})$ , mit denen angegeben wird, dass der Konfiguration  $\mathcal{R}$  mit dem Tupel der Assoziationenmengen  $W$  verteilt die Konfiguration  $\mathcal{S}$  zugeordnet wird. Für jedes Tupel  $W$  gilt, dass die Menge der zugehörigen Zuordnungen mit der Relation *zuordnen* übereinstimmt

(Satz B.22):

$$\forall W \in \prod_{i=1}^k \wp(\mathbf{A}) : \left\{ (\mathcal{R}, \mathcal{S}) \mid (\mathcal{R}, W, \mathcal{S}) \in \frac{\mathbf{A}}{v||z} \right\} = \frac{* \mathbf{A}}{z}$$

Die Relation ist Grundlage für die parallele Analyse mit mehreren Rechnern, die mit Zuordnungsautomaten modelliert wird (Satz B.71).



**Abb. 2-4:** Parallel, verteilt und kaskadiert zuordnen  
links oben: parallel zuordnen; rechts oben: verteilt zuordnen,  
unten: kaskadiert zuordnen

**Kaskadierte Zuordnungen** Eine insgesamt zu analysierende Konfiguration  $\mathcal{R}$  kann auch in  $n$  Teilmengen  $\mathcal{U}_1, \mathcal{U}_2, \dots, \mathcal{U}_n$  aufgeteilt und dann schrittweise analysiert werden. Diese Situation ergibt sich beispielsweise bei der explorativen Analyse (Abbildung 2-5). Hier steht am Anfang der Analyse nur  $\mathcal{U}_1$  zur Verfügung. Dann werden nach und nach weitere Daten  $\mathcal{U}_2, \dots$  erfasst und unmittelbar nach der Übergabe analysiert. Abbildung 2-4 unten verdeutlicht diesen Ablauf im Vergleich zu verteilten Zuordnungen. Die Relation *kaskadiert zuordnen* (Gl. B-23) stimmt mit der Relation *zuordnen* überein (Satz B.26).

**Vergleich der Zuordnungen** Wenn *parallel* oder *verteilt zugeordnet* wurde, können die in mehreren parallelen Analysesträngen erfolgten Überprüfungen und Ergänzungen von Konfigurationen auch seriell mit den gleichen Assoziationen nacheinander durchgeführt werden. Dabei kann höchstens eine neu hinzuzufügende Konfiguration, die parallel noch nicht vorlag, seriell bereits vorliegen, so dass es sein kann, dass parallel Vereinigungsoperationen ausgeführt werden, die seriell unnötig wären. Dies wird von der Relation *mehrfach zuordnen* (Gl. B-26, B-27) berücksichtigt, mit der bewiesen wird, dass die Relationen *parallel* und *verteilt zuordnen* mit der Relation *zuordnen* übereinstimmen (Beweise von Satz B.19, B.22).

Um auch für jede Folge *direkter Zuordnungsschritte*, die für eine Zuordnung



$(\mathcal{R}, \mathcal{S}) \in \xrightarrow[z]{*\mathbf{A}}$  durchgeführt wird, einen entsprechenden Eintrag in den Relationen *parallel zuordnen* zu erhalten, sind bei *parallelen Zuordnungen* analog zur Definition von *zuordnen* Folgen *direkter paralleler Zuordnungsschritte* vorgesehen, mit denen jede Folge *direkter Zuordnungsschritte* gleichzeitig eine *direkte parallele Zuordnung* ist (Abbildung 2-4 links oben). Ohne diesen "Trick" könnte es vorkommen, dass seriell erfolgreich ausgeführte Überprüfungen, die nur deshalb erfolgreich sind, weil im seriellen Ablauf unmittelbar vor der Überprüfung Instanzen neu zugeordnet wurden, von der Relation *parallel zuordnen* nicht erfasst werden. Mit diesem "Trick" ist jeder serielle Ablauf der Relation *zuordnen* in der Relation *parallel zuordnen* enthalten.

Bei *parallelen* sowie *asynchronen Zuordnungen* stehen alle Instanzen für die Überprüfungen aller parallel durchgeführten Folgen von Zuordnungen zur Verfügung. Bei *verteilten Zuordnungen* werden die Instanzen dagegen aufgeteilt und können nur eingeschränkt überprüft werden (Abbildung 2-4 rechts oben). Deswegen ist es möglich, dass neue Instanzen, die seriell, parallel oder asynchron zugeordnet worden wären, bei der *verteilten Zuordnung* fehlen. Trotzdem können auch die bei der *verteilten Zuordnung* unabhängig (parallel) ausgeführten Folgen von Zuordnungsschritten seriell nacheinander ausgeführt werden. Da es aber möglich ist, dass parallel Instanzen zugeordnet werden, die verteilt nicht zugeordnet werden können, müssen alle Instanzen nach der *verteilten Zuordnung* zusammengefasst und danach weiteren Zuordnungen unterworfen werden. In Gleichung B-22 wird deshalb  $\left| \frac{v||Z}{M} \right|$  als Produkt von  $\left| \frac{v|Z}{M} \right|$  und  $\left| \frac{*Z}{M} \right|$  definiert.

### Systeme zur vollständigen Zuordnung

Für Zustandsübergänge von Zuordnungsautomaten (Gl. B-86 bis B-88) werden weitere Relationen benötigt, mit denen für Mengen von Assoziationen mit relativ wenigen Assoziationen Folgen von Zuordnungen beschrieben werden, bei denen jede Assoziation genau ein Mal überprüft wird.

Bei parallelen Zuordnungen heißt dies z. B., dass einer Konfiguration  $\mathcal{R}$  mit der Menge der Assoziationen  $\mathbf{A}$  *direkt vollständig parallel* eine Konfiguration  $\mathcal{S}$  *zugeordnet* wird, wenn für alle Assoziationen  $(\mathcal{X}, \mathcal{Y}) \in \mathbf{A}$  geprüft wird, ob  $\mathcal{X}$  in der Konfiguration  $\mathcal{R}$  vorliegt, und, falls  $\mathcal{X}$  vorliegt, alle Instanzen von  $\mathcal{Y}$  zusätzlich zu  $\mathcal{R}$  in  $\mathcal{S}$  eingetragen werden (Gl. B-40).

Bei *direkten vollständigen Zuordnungen* wird eine Folge von Zuordnungsschritten betrachtet, die so viele Schritte umfasst, wie es Assoziationen gibt (Gl. B-39). Zuerst werden alle Assoziationen einer Menge von Assoziationen mit  $k$  Assoziationen in beliebiger Reihenfolge angeordnet. Dann wird für jede Assoziation  $(\mathcal{X}_i, \mathcal{Y}_i) \in \mathbf{A}, i = 0, 1, \dots, k - 1$  entweder eine direkte Zuordnung mit  $Q_{i+1} = Q_i \cup \mathcal{Y}_i$  oder der leere Zuordnungsschritt mit  $Q_{i+1} = Q_i$  durchgeführt. Dabei wird die erste Konfiguration  $Q_0$  als linke Konfiguration  $\mathcal{R}$  und die letzte Konfiguration  $Q_k$  als rechte Konfiguration  $\mathcal{S}$  der Assoziation  $(\mathcal{R}, \mathcal{S})$  in die Relation *direkt vollständig zuordnen* aufgenommen. Eine direkte Zuordnung wird ausgeführt, wenn  $\mathcal{X}_i$  in  $Q_i$  vor-

liegt ( $\mathcal{X}_i \subseteq Q_i$ ), der leere Zuordnungsschritt wird ausgeführt, wenn  $\mathcal{X}$  nicht vorliegt ( $\mathcal{X}_i \not\subseteq Q_i$ ).

*Direkte vollständige asynchrone Zuordnungen* bauen auf *direkten vollständigen Zuordnungen* auf. Bei *direkten vollständigen asynchronen Zuordnungen* wird die Überprüfung, ob  $\mathcal{X}_i$  vorliegt, aber nicht auf  $Q_i$  beschränkt. Vielmehr kann jede Konfiguration  $Q_j$  mit  $j \leq i$  überprüft werden. Die Zuordnung wird also durchgeführt, wenn mit einem beliebig ausgewählten  $Q_j, j \leq i$  gilt  $\mathcal{X}_i \subseteq Q_j$ . Wenn diese Bedingung nicht zutrifft, wird der leere Zuordnungsschritt  $Q_{i+1} = Q_i$  für diese Assoziation durchgeführt (Gl. B-41). Dies erfolgt auch, wenn  $\mathcal{X}_i \subseteq Q_i$  gelten würde.

Alle vollständigen Zuordnungen führen zu den gleichen Ergebnissen wie die entsprechenden einfachen Zuordnungen (Lemma B.28, B.29, B.30).

### Systeme zur Zuordnung von Konfigurationen mit Konzepten

Die Assoziationen von  $\mathbf{A}$  definieren für jeden vorstellbaren Analyseschritt eine Assoziation  $(\mathcal{X}, \mathcal{Y}) \in \mathbf{A} \subseteq \wp(\mathcal{I}) \times \wp(\mathcal{I})$ . Alternativ kann die Analysevorschrift durch wenige Produktionen  $(\mathcal{A}, \mathcal{B}) \in \mathbf{P} \subseteq \wp(\mathcal{I}) \times \wp(\mathcal{I})$  und eine Instanzierungsfunktion  $i$  definiert werden.

Jede Konfiguration  $\mathcal{X}$ , für die es eine Produktion  $(\mathcal{A}, \mathcal{B}) \in \mathbf{P}$  gibt, mit der die Instanzierungsfunktion  $i$  eine nicht leere Menge  $\mathcal{Y} = i(\mathcal{B}|\mathcal{A} \rightarrow \mathcal{X})$  (mit  $\mathcal{A}, \mathcal{B}, \mathcal{X}, \mathcal{Y} \subseteq \mathcal{I}$ ) berechnet, wird zusammen mit  $\mathcal{Y}$  in die Menge der Assoziationen  $\mathbf{A}_K(\mathcal{I}, \mathbf{P}, i)$  aufgenommen (Gl. B-46):

$$\mathbf{A}_K(\mathcal{I}, \mathbf{P}, i) := \left\{ (\mathcal{X}, \mathcal{Y}) \left| \begin{array}{l} \mathcal{X} \subseteq \mathcal{I} \\ \exists (\mathcal{A}, \mathcal{B}) \in \mathbf{P} : \\ \mathcal{Y} = i(\mathcal{B}|\mathcal{A} \mapsto \mathcal{X}) \neq \emptyset \end{array} \right. \right\}$$

Die Schreibweise  $\mathcal{A} \mapsto \mathcal{X}$  soll ausdrücken, dass die Instanzierungsfunktion die Konfiguration  $\mathcal{A}$  an die zu analysierende Konfiguration  $\mathcal{X}$  anpasst. Dazu muss die Instanzierungsfunktion eine Transformation bestimmen, die  $\mathcal{A}$  in  $\mathcal{X}$  überführt [ $\mathcal{X} = i(\mathcal{A}|\mathcal{A} \mapsto \mathcal{X})$ ]. Die Funktion  $i(\mathcal{B}|\mathcal{A} \mapsto \mathcal{X})$  definiert die Assoziationen  $(\mathcal{X}, \mathcal{Y}) \in \mathbf{A}_K(\mathcal{I}, \mathbf{P}, i)$  des Zuordnungssystems  $(\mathcal{I}, \mathbf{A}_K(\mathcal{I}, \mathbf{P}, i))$ , indem die Konfiguration  $\mathcal{B}$  der gleichen Transformation unterworfen wird wie  $\mathcal{A}$ , so dass aus  $\mathcal{B}$  die Konfiguration  $\mathcal{Y}$  entsteht [ $\mathcal{Y} = i(\mathcal{B}|\mathcal{A} \mapsto \mathcal{X})$ ]. Einer Produktion  $(\mathcal{A}, \mathcal{B})$  entsprechen auf diese Weise viele Assoziationen  $(\mathcal{X}, \mathcal{Y})$ .

Zusätzlich zur Menge der definierten Instanzen  $\mathcal{I}$  wird bei Zuordnungssystemen mit Konzepten  $(\mathcal{I}, \mathbf{P}, i)$  die Menge der Konzept-Repräsentanten  $\mathcal{I}_K$  und die Menge der instanziierten Konzepte  $\mathcal{I}_i$  mit  $\mathcal{I}_K = \bigcup_{(\mathcal{A}, \mathcal{B}) \in \mathbf{P}} (\mathcal{A} \cup \mathcal{B})$  (Gl. B-43) und

$\mathcal{I}_i = \bigcup_{(\mathcal{X}, \mathcal{Y}) \in \mathbf{A}_K(\mathcal{I}, \mathbf{P}, i)} (\mathcal{X} \cup \mathcal{Y})$  (Gl. B-47) definiert. Instanzen  $\gamma$ , die in  $\mathcal{I}$ , nicht aber in  $\mathcal{I}_i$  enthalten sind, sind in keiner Konfiguration einer Assoziation enthalten (Gl. B-47) und beeinflussen daher die Analyse nicht. Mit den einzelnen Konzept-Repräsentanten von  $\mathcal{I}_K$  werden Konzept-Mengen definiert, in denen die aus dem jeweiligen Konzept zu instanziiierenden Instanzen enthalten sind (Gl. B-45).  $(\mathcal{I}, \mathbf{A}_K(\mathcal{I}, \mathbf{P}, i))$  ist das zu  $(\mathcal{I}, \mathbf{P}, i)$  gehörige Zuordnungssystem mit Instanzen.

Zuordnungssysteme mit Konzepten  $(\mathcal{I}, \mathbf{P}, i)$  dienen zur Definition von Zuordnungsautomaten, die in den nächsten Abschnitten eingeführt werden.

Abschnitt B.1.4 definiert Zuordnungssysteme mit Konzepten.

### Systeme zur Zuordnung von Konfigurationen mit Ableitungen

Instanzen werden in den folgenden Beispielen durch Merkmalvektoren, die Elemente der Menge  $\mathcal{I}_M := M_1 \times M_2 \cdots \times M_m$  sind, beschrieben (Gl. B-51). Die Mengen  $M_i$  sind dabei die sog. Merkmale der Instanz, und  $T = M_1$  ist das Typ-Merkmal, mit dem u. a. die Anzahl der Überprüfungsaufträge (Gl. B-83, B-127) reduziert wird, das aber auch für die Definition von Produktionsnetzen (Gl. B-58, Abbildung 1-4), Ableitungsgraphen (Gl. B-59, B-60) und Beschreibungsgraphen (Gl. B-61 bis B-64) notwendig ist. Die Merkmalmengen bzw. Attribute  $M_i$  werden in dem Tupel  $\mathbf{I} = (M_1, M_2, \dots, M_m)$  zusammengefasst und definieren das Zuordnungssystem mit Ableitung  $(\mathbf{I}, \mathbf{P}, i)$  (Gl. B-50).

Das Typ-Merkmal  $T$  entspricht in vieler Hinsicht der Menge der Symbole  $V$ , mit der Zeichenketten definiert werden (Gl. 2-4). Mit  $M_i, i = 2, 3, \dots$  als "Koordinaten" sowie  $T$  als Symbolmenge ergeben sich mit Merkmalvektoren gewisse Parallelen zu *Koordinatengrammatiken* /Mil72, Sha69, Mich96, Mich98a/, die im Abschnitt 4.1 mit Zuordnungssystemen und -automaten verglichen werden.

Bei Ableitungen wird zusätzlich für jeden Merkmalvektor angegeben, aus welchen Merkmalvektoren er entstand. Dazu wird bei jeder Ableitung ein Index  $i_x \in \mathbb{N}$  und eine Menge von Vorgängern eingetragen.  $P_M$  sei die Familie  $\wp(\mathbb{N})$  aller möglicher Indexmengen. Ableitungen sind dann Elemente der Menge  $\mathcal{I}_A := \mathcal{I}_M \times \mathbb{N} \times P_M$  (Gl. B-54) und verketteten Merkmalvektoren, so dass die Ableitungsgraphen (Gl. B-59, B-60) dieser Instanzen unmittelbar vorliegen und z. B. auch von den Instanzierungsfunktionen zur Analyse sowie von Visualisierungswerkzeugen genutzt werden können. Ausführbare Programme beschreiben Instanzen deshalb in der Regel als Ableitungen. Ableitungen bzw. Assoziationen, die Ableitungen generieren, können eingeschränkt werden. Graph-differenzierende Assoziationen (Gl. B-75) ignorieren den Index, merkmal-differenzierende Assoziationen (Gl. B-74) zusätzlich die Verkettung. Vorteil graph-differenzierender Assoziationen ist, dass Zuordnungsautomaten mit endlichen Merkmalmengen  $M_i$  und graph-differenzierenden Assoziationen terminieren (Satz B.38, Satz B.53).

Zuordnungssysteme mit Ableitungen  $(\mathbf{I}, \mathbf{P}, i)$  werden in Abschnitt B.1.5 definiert und lassen sich als Zuordnungssysteme mit Konzepten oder mit Instanzen darstellen:  $(\mathcal{I}_A, \mathbf{P}, i)$  ist das zu  $(\mathbf{I}, \mathbf{P}, i)$  gehörige Zuordnungssystem mit Konzepten,  $(\mathcal{I}_A, \mathbf{A}_K(\mathcal{I}_A, \mathbf{P}, i))$  ist das zugehörige Zuordnungssystem mit Instanzen.

### Eigenschaften von Assoziationen

Zeichenketten, Bäume und Graphen stellen die Einbettung von Teilmustern in übergeordnete Muster durch die jeweilige Repräsentation der Muster sicher. Bei Zuordnungssystemen, die Muster durch (ungeordnete) Mengen repräsentieren,

muss die Einbettung durch die Assoziationen sichergestellt werden. Zusätzliche Anforderungen entstehen dadurch, dass der Entwickler in der Regel bei der Formulierung der Assoziationen die ersetzende Analyse vor Augen hat, die Analyse aber zuordnend durchgeführt wird. Dabei werden die Zwischenergebnisse nicht gelöscht, so dass zuordnend entsprechend Gleichung 2-8 andere Ergebnisse entstehen können als bei einer ersetzenden Analyse nach Gleichung 2-7 (vgl. Abschnitt 3.1).

Um Anforderungen zur Realisierung der Einbettung und zur zuordnenden Analyse formal zu erfassen, werden Mengen von Assoziationen  $\mathbf{A} \subseteq \wp(\mathcal{I}) \times \wp(\mathcal{I})$  als mehrfach assoziierend, baumproduzierend, deterministisch, vexierend<sup>5</sup>, global sowie lokal kontextfrei, hierarchisch, heterarchisch, kontextsensitiv, anlagernd, merkmaldifferenzierend und graph-differenzierend eingeordnet. Dabei gibt es die Eigenschaften teilweise noch in einer ersetzenden und einer zuordnenden Variante:

1. Die Menge der Assoziationen  $\mathbf{A}$  heißt *mehrfach assoziierend*, wenn in einer Folge von Ersetzungsschritten eine bereits vorliegende Instanz als Ersatz für eine Menge von Instanzen hinzugefügt wird (Gl. B-72). Dieses wiederholte Eintragen in eine Menge bewirkt keine Änderung und wird daher ignoriert (vgl. auch die Diskussion von Verfahren mit einfach assoziierenden Assoziationen auf Seite 39).
2. Wenn die Konfigurationen  $\mathcal{X}$  und  $\mathcal{Y}$  aller Assoziationen  $(\mathcal{X}, \mathcal{Y}) \in \mathbf{A}$  disjunkt sind, heißt die Menge der Assoziationen  $\mathbf{A}$  *lokal kontextfrei* (Gl. B-69).
3. Wenn die zu entfernenden Instanzen der Konfigurationen  $\mathcal{X}$ , der Assoziationen  $(\mathcal{X}, \mathcal{Y}) \in \mathbf{A}$  mit oder nach einem Ersetzungsschritt mit  $(\mathcal{X}, \mathcal{Y})$  nicht wieder eingefügt werden, heißt die Menge der Assoziationen *global kontextfrei*; andernfalls heißt sie *kontextsensitiv* (Gl. B-71).
4. Mit beschreibenden Merkmalen können Instanzen Ausdehnungen zugeordnet werden (Gl. B-76).
5. Assoziationenmengen  $\mathbf{A}$  heißen *anlagernd*, wenn die Ausdehnungen (Gl. B-76) der Instanzen von  $\mathcal{X} \setminus \mathcal{Y}$  disjunkt sind (mit  $(\mathcal{X}, \mathcal{Y}) \in \mathbf{A}$ , Gl. B-77).
6. Wenn bei Zuordnungen für jeden Merkmalvektor nur eine Ableitung generiert werden kann, heißen Assoziationenmengen *merkmal-differenzierend* (Gl. B-74).
7. Wenn bei Zuordnungen für jede Ableitung (unter Vernachlässigung ihres Indexwertes) nur ein Ableitungsgraph generiert werden kann, heißen Assoziationenmengen *graph-differenzierend* (Gl. B-75).

Im Anhang sind die entsprechenden Definitionen mit Gleichung B-67 bis Gleichung B-75 aufgelistet. Die Untersuchungen zur Einordnung der Assoziationenmengen  $\mathbf{A}$  sind noch nicht abgeschlossen und werden im Ausblick (Abschnitt 5.2) aufgegriffen.

#### 2.2.4 Automaten zur Zuordnung von Konfigurationen

Zuordnungsautomaten dienen zur formalen Beschreibung der im Abschnitt 1.4 erläuterten Tafelsysteme mit BPI-Architektur und sind z. B. eine formale Grund-

---

<sup>5</sup>vexierend: irreführend; Abbildung 1-1 zeigt z. B. ein Vexierbild

lage zur Implementierung dieser Programme auf Parallelrechnern. Zuordnungsautomaten  $((\mathcal{I}, \mathbf{P}, i), a)$  bauen auf Zuordnungssystemen mit Konzepten  $(\mathcal{I}, \mathbf{P}, i)$  auf und realisieren zusätzlich eine Ablaufkontrolle mittels Zuständen und Zustandsübergängen sowie einer Auswahlfunktion  $a$ . Mit Zuordnungssystemen werden dabei Eigenschaften von Zuordnungsautomaten, die unabhängig von der Ablaufkontrolle der Zuordnungsautomaten sind, formuliert und bewiesen.

### Zustände und Zustandsübergänge

Die Relation *geht direkt über*  $\xRightarrow[A]{\Rightarrow} \subseteq [\wp(\mathcal{I}) \times \wp(u(\mathcal{I}))]^2$  (Gl. B-86) für Transformationssysteme  $(\wp(\mathcal{I}), \xRightarrow[A]{\Rightarrow})$  definiert die Relation *geht über*  $\xRightarrow[A]{*}$  (Gl. 2-1, B-89). Dabei ist  $u(\mathcal{I}) := \mathcal{I} \times \mathbf{P}$  die Menge der Überprüfungsaufträge (Gl. B-83), die zu überprüfende Hypothesen vormerken. Der Zustand eines Zuordnungsautomaten wird durch eine Menge zu analysierender Instanzen und eine Menge von Überprüfungsaufträgen definiert (Def. B.44). Zusätzlich wird für Zuordnungsautomaten die Relation *geht abschließend über*  $\xRightarrow[\dagger A]{\Rightarrow}$  mit  $\xRightarrow[\dagger A]{*} \subseteq \xRightarrow[A]{*}$  definiert (Gl. B-90).

Bei jedem Zustandsübergang  $(\mathcal{R}, U_r) \xRightarrow[A]{*} (\mathcal{S}, U_s)$  des Automaten  $A((\mathcal{I}, \mathbf{P}, i), a)$  wird der Konfiguration  $\mathcal{R}$  durch das System  $(\mathcal{I}, \mathbf{P}, i)$  die Konfiguration  $\mathcal{S}$  zugeordnet ( $\mathcal{R} \xRightarrow[z]{*} \mathcal{S}$ , Satz B.49). Mit den Überprüfungsaufträgen in  $U_r$  bzw.  $U_s$  für die Zustände  $(\mathcal{R}, U_r)$  bzw.  $(\mathcal{S}, U_s)$  wird vorgemerkt, welche Instanzen  $\gamma \in \mathcal{R}$  bzw.  $\gamma \in \mathcal{S}$  mit welchen Produktionen überprüft werden müssen. Sobald keine Überprüfungsaufträge mehr vorliegen, hält der Automat an. Mit der Auswahlfunktion  $a$  werden die beim jeweiligen Zustandsübergang zu bearbeitenden Überprüfungsaufträge ausgewählt.

Wenn alle aus den Instanzen eines Zustandes ableitbaren neuen Instanzen vorliegen oder durch Hypothesen von Überprüfungsaufträgen abgedeckt sind, heißt der Zustand  $(\mathcal{R}, U_r)$  *analysierbar* ( $\mathcal{R} \boxtimes U_r$ , Gl. B-98). Durch die Art und Weise, wie Zustandsübergänge Hypothesen überprüfen und neue Überprüfungsaufträge generieren, ist sichergestellt, dass ein analysierbarer Zustand wieder in einen analysierbaren Zustand übergeht (Satz B.59). Der Zustand  $(\mathcal{R}, u(\mathcal{R}))$ , für den alle Überprüfungsaufträge  $u(\mathcal{R})$  von  $\mathcal{R}$  zu berechnen sind, ist immer analysierbar (Korollar B.66). Mit ihm wird in der Regel jede Analyse begonnen.

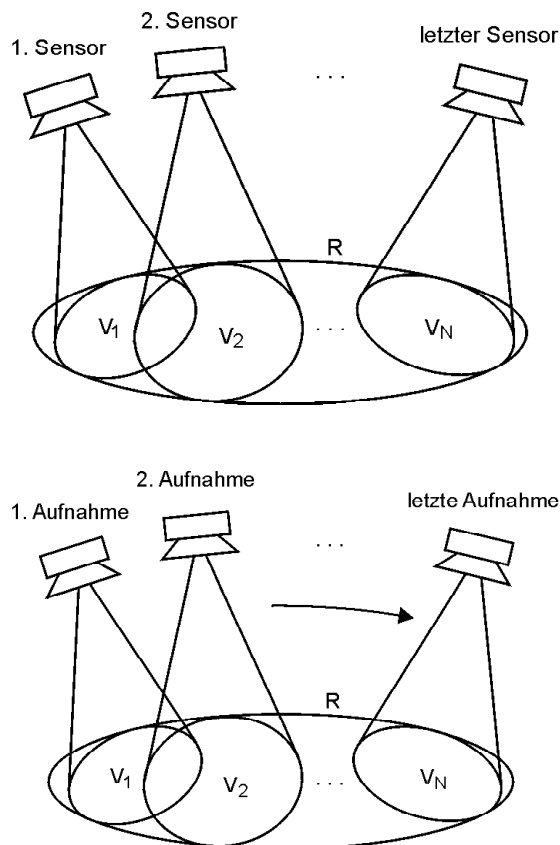
Abschnitt B.2 definiert Zuordnungsautomaten.

### Analyse mit Zuordnungsautomaten

Die mit Zuordnungssystemen für terminierende Zuordnungen definierte Analysefunktion  $AF$  wird mit Zuordnungsautomaten berechnet (Korollar B.63):  $(\mathcal{R}, u(\mathcal{R})) \xRightarrow[\dagger A]{\Rightarrow} (AF(\mathcal{R}), \emptyset)$ .

Die Analyse kann flexibel organisiert werden:

1. Die Analyse kann mit mehreren Prozessoren, die einen gemeinsamen Speicher nutzen, parallel durchgeführt werden (Satz B.69).
2. Die zu analysierenden Instanzen können in beliebige Teilmengen, die einzeln analysiert werden, zerlegt werden (Satz B.71). Damit ist es z.B. möglich, die Analyse mit mehreren Rechnern eines Rechnernetzes auszuführen. Außerdem können mehrere Sensoren parallel Bilddaten von einer zu analysierenden Szene erfassen und parallel auswerten (Abbildung 2-5 oben).
3. Die zu analysierenden Instanzen können in beliebiger Reihenfolge auch während der Analyse übergeben werden (Satz B.68). Damit ist es z.B. möglich, eine mit einem autonomen System durchgeführte explorative Analyse, die rückgekoppelt neue Daten, aufbauend auf bisherigen Analyseergebnissen, erfasst und dann zusammen mit den alten Daten analysiert, durchzuführen (Abbildung 2-5 unten).



**Abb. 2-5:** Explorative Analyse mit "Nachschieben" der Daten während der Analyse  
 oben: verteilte nicht explorative Auswertung (ohne "Nachschieben")  
 unten: explorative Auswertung mit bewegtem Sensor (und "Nachschieben")

Die Sätze B.68, B.69 und B.71 sind Grundlagen für die Erweiterung der im Abschnitt 1.4 beschriebenen BPI-Architektur für den Einsatz mit Parallelrechnern und die Analyse in der Schleife mit einer analysegesteuerten Datenerfassung.

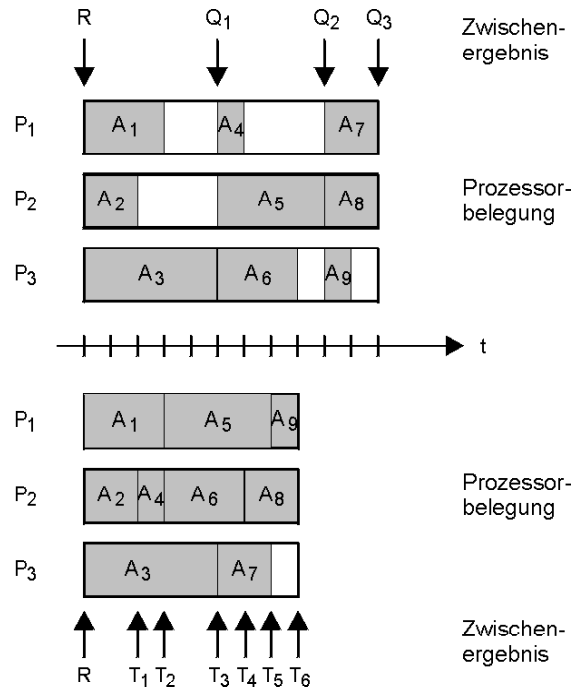
Die Analyse mit mehreren Rechnern eines Rechnernetzes wird vom Prinzip her entsprechend Satz B.71 organisiert. Die technische Realisierung kann versuchen, auf die Bestimmung der Menge der Überprüfungsaufträge zur *vereinten Überprüfung*  $\hat{u}$  (Gl. B-121), die nach Gleichung B-124 für die Analyse im Rechnernetz benötigt wird, zu verzichten. Arbeitshypothese ist, dass  $\hat{u}$  nicht benötigt wird, wenn die zu überprüfenden Instanzen in lokal überprüfbare Instanzen ( $\mathcal{I}_i$  in Gl. B-122), die im jeweiligen verteilten Rechner, und global überprüfbare Instanzen, die im nachgeschalteten zentralen Rechner durchgeführt werden müssen, aufgeteilt werden.

Die Abbildung 2-6 zeigt entsprechend Satz B.69 den Einsatz mehrerer Prozessoren mit einem gemeinsamen Arbeitsspeicher zur Bearbeitung einer Menge von Assoziationen bei einem Zustandsübergang.

Mit Abbildung 2-6 (oben) wird entsprechend Gleichung B-87 der Fall erläutert, dass alle Assoziationen der Überprüfungsaufträge parallel überprüft werden. Dabei steht für jede Assoziation  $A_i = (\mathcal{X}_i, \mathcal{Y}_i)$  ein Prozessor  $P_i$  zur Verfügung, der eine Konfiguration  $\mathcal{P}_j$  auf  $\mathcal{X}_i \subseteq \mathcal{P}_j$  überprüft und aus ihr die Konfiguration  $\mathcal{P}_{j+1}$  generiert mit  $\mathcal{P}_{j+1} = \mathcal{P}_j$  (falls  $\mathcal{X}_i \not\subseteq \mathcal{P}_j$ ) oder  $\mathcal{P}_{j+1} = \mathcal{P}_j \cup \mathcal{Y}_i$  (falls  $\mathcal{X}_i \subseteq \mathcal{P}_j$ ). Die Auswahlfunktion  $a$  (Gl. B-87) sollte also in diesem Fall bei Zustandsübergängen so viele Überprüfungsaufträge auswählen, dass jeder Prozessor eine Assoziation zu bearbeiten hat. Die maximale Rechenzeit zur Überprüfung einer Assoziation ist auch die Zeit zur Analyse aller parallel bearbeiteten Assoziationen, so dass mit Mengen von Assoziationen, die zur Überprüfung der einzelnen Assoziationen sehr unterschiedliche Rechenzeiten benötigen, erhebliche Leerzeiten auftreten. Die Abbildung 2-6 (oben) zeigt für drei Prozessoren drei Zyklen: im ersten Zyklus werden die Assoziationen  $\mathbf{A}_1$ ,  $\mathbf{A}_2$  und  $\mathbf{A}_3$  bearbeitet; im zweiten Zyklus bearbeiten die drei Prozessoren die Assoziationen  $\mathbf{A}_4$ ,  $\mathbf{A}_5$  und  $\mathbf{A}_6$  und im dritten Zyklus die Assoziationen  $\mathbf{A}_7$ ,  $\mathbf{A}_8$  und  $\mathbf{A}_9$ . Die Auswahlfunktion  $a$  wählt also in diesem vereinfachten Beispiel bei jedem Zustandsübergang Überprüfungsaufträge mit insgesamt drei Assoziationen aus.  $\mathcal{R}$  ist im ersten Zyklus gleich  $\mathcal{P}_0$ , im zweiten Zyklus ist  $\mathcal{R}$  gleich  $\mathcal{Q}_1$ , und im dritten Zyklus ist  $\mathcal{R}$  gleich  $\mathcal{Q}_2$ . Als Zwischenergebnisse entstehen  $\mathcal{Q}_1 = \mathcal{P}_1 \cup \mathcal{P}_2 \cup \mathcal{P}_3$ ,  $\mathcal{Q}_2 = \mathcal{P}_4 \cup \mathcal{P}_5 \cup \mathcal{P}_6$  und  $\mathcal{Q}_3 = \mathcal{P}_7 \cup \mathcal{P}_8 \cup \mathcal{P}_9$ .

Bei Abbildung 2-6 (unten) wird im Gegensatz dazu davon ausgegangen, dass die Anzahl der zur Verfügung stehenden Prozessoren wesentlich kleiner als die Anzahl der für einen Zustandsübergang von der Auswahlfunktion  $a$  (Gl. B-88) ausgewählten und zu überprüfenden Assoziationen ist. Sobald ein Prozessor eine Assoziation überprüft hat, beginnt der Prozessor sofort mit der Überprüfung der nächsten noch von keinem anderen Prozessor bearbeiteten Assoziation. Diese Überprüfungen erfolgen, ohne dass der Beginn und das Ende der Bearbeitung der einzelnen Assoziationen aufeinander abgestimmt werden und heißen deshalb asynchron (Gl. B-88). Überprüft wird die bei Start der Überprüfung vorliegende Konfiguration  $\mathcal{T}_k$ , modifiziert wird die bei Abschluss der Überprüfung vorliegende Konfiguration  $\mathcal{T}_l$ , die sich in der Regel deshalb von  $\mathcal{T}_k$  unterscheidet, weil andere Prozessoren sie in der Zwischenzeit ergänzten. So entstehen praktisch keine Leerzeiten. Die Abbildung zeigt für drei Prozessoren einen Zyklus mit neun Assoziationen.  $\mathcal{T}_6$  und  $\mathcal{Q}_3$  entsprechen sich und werden in der Regel die gleichen Instanzen enthalten; sie

können aber auch voneinander abweichen.



**Abb. 2-6:** Parallelverarbeitung mit parallelen und asynchronen Zuordnungen  
 oben: parallele Analyse mit mehreren Leerzeiten  
 unten: asynchrone Analyse mit reduzierter Leerzeit

Im Anhang B werden Zuordnungssysteme und Zuordnungsautomaten lückenlos detailliert dargestellt. Dort werden auch  $\xRightarrow{A}$  und  $\xRightarrow{\dagger A}$  definiert. Um dabei Transformationssysteme, Zuordnungssysteme sowie Zuordnungsautomaten besser zu unterscheiden, werden die hier verwendeten sehr ähnlichen Relations-Symbole  $\xRightarrow{z}$ ,  $\xRightarrow{*A}$ ,  $\xRightarrow{\dagger A}$ ,  $\xRightarrow{||z}$ ,  $\xRightarrow{v||z}$ ,  $\xRightarrow{a||z}$  bzw.  $\xRightarrow{A}$ ,  $\xRightarrow{*A}$  und  $\xRightarrow{\dagger A}$  der Transformationssysteme im Anhang anders bezeichnet; dort gilt für Zuordnungssysteme:  $\xRightarrow{z} = \left| \frac{z}{\quad} \right|$ ,  $\xRightarrow{*A} = \left| \frac{*Z}{\quad} \right|$ ,  $\xRightarrow{\dagger A} = \left| \frac{\dagger Z}{\quad} \right|$ ,  $\xRightarrow{||z} = \left| \frac{||Z}{\quad} \right|$ ,  $\xRightarrow{v||z} = \left| \frac{v||Z}{W} \right|$ ,  $\xRightarrow{a||z} = \left| \frac{a||Z}{\quad} \right|$ ,  $\xRightarrow{A} = \left| \frac{A}{\quad} \right|$ ,  $\xRightarrow{*A} = \left| \frac{*A}{\quad} \right|$  und  $\xRightarrow{\dagger A} = \left| \frac{\dagger A}{\quad} \right|$ .

### 2.3 Erzeugungs- und Erkennungsverfahren

Zuordnungssysteme und Zuordnungsautomaten wurden entwickelt, um Muster, die durch Konfigurationen (d.h. Mengen) von Instanzen definiert sind, zu erkennen. Anschließend werden zur Präzisierung und Diskussion dieses Musteranaly-



seansatzes Verfahren für die Erzeugung und Erkennung von Konfigurationen definiert, die auf Konfigurationen-Ersetzungs- bzw. -Zuordnungssystemen aufbauen. Damit wird der Bezug der Zuordnungssysteme und -automaten zur Musteranalyse hergestellt.

Erzeugungsverfahren generieren terminale Instanzen  $\gamma \in \mathcal{I}_T$ , die von Erkennungsverfahren analysiert werden. Bei der Analyse von Bildern sind die terminalen Instanzen in der Regel als Merkmalvektoren beschriebene primitive Flächen- oder Liniensegmente und heißen deswegen auch Primitivobjekte.

Startsymbole  $\sigma \in \Sigma$  sind diejenigen Instanzen, die vollständige Muster repräsentieren. Jede Menge  $\Sigma$  repräsentiert eine *Aussage* mit ihren unterschiedlichen *Ausprägungen*  $\sigma$ . Die Aussage kann beispielsweise sein, dass ein *Objekt Kreuzung vorliegt*; die spezielle Ausprägung ergäbe sich z. B. durch die Objektform, die Umgebung der Kreuzung, den Sensor, die Sensorposition und -einstellung sowie das Extraktionsprogramm zur Generierung terminaler Instanzen aus den Sensordaten.

Erzeugungsverfahren beginnen mit Startsymbolen, und Erkennungsverfahren übergeben sie als Endergebnis der Analyse. Abgesehen von den Startsymbolen sind die nichtterminalen Instanzen  $\gamma \in \mathcal{I}_N$  Zwischenergebnisse, die während der Erzeugung bzw. Erkennung verwendet werden. Eine Instanz  $\gamma \in \mathcal{I}$  kann nicht gleichzeitig terminal und nichtterminal sein:

$$\begin{aligned}\mathcal{I} &= \mathcal{I}_T \cup \mathcal{I}_N \\ \mathcal{I}_T \cap \mathcal{I}_N &= \emptyset\end{aligned}\tag{2 - 11}$$

Startsymbole werden den nichtterminalen Instanzen zugerechnet:

$$\Sigma \subseteq \mathcal{I}_N\tag{2 - 12}$$

Abbildung 2-7 erläutert diese Zusammenhänge.

Bevor die Verfahren für mehrdimensionale Muster definiert werden, werden einleitend aus der Literatur bekannte Erzeugungs- und Erkennungsverfahren für eindimensionale Zeichenketten angegeben.

Die in diesem Abschnitt für Konfigurationen eingeführten Begriffe *Startsymbol*, *terminale* und *nichtterminale Instanzen* leiten sich aus den entsprechenden für Zeichenkettengrammatiken definierten Begriffen ab /Ste88/.

### 2.3.1 Erzeugungs- und Erkennungsverfahren für Zeichenketten

Erzeugungs- und Erkennungsverfahren für Zeichenketten bauen auf Semi-Thue-Systemen  $P \subseteq W^*(V) \times W^*(V)$  auf /Thu14/ (Gl. 2-5) und manipulieren Zeichenketten  $W^*(V)$  /Sal78/ (Gl. 2-4) über dem Vokabular  $V$ . Mit Semi-Thue-Systemen werden (Gl. 2-6) Transformationssysteme  $(W^*(V), \xrightarrow[STS]{(V,P)})$  und Grammatiken /Sal78/ definiert.

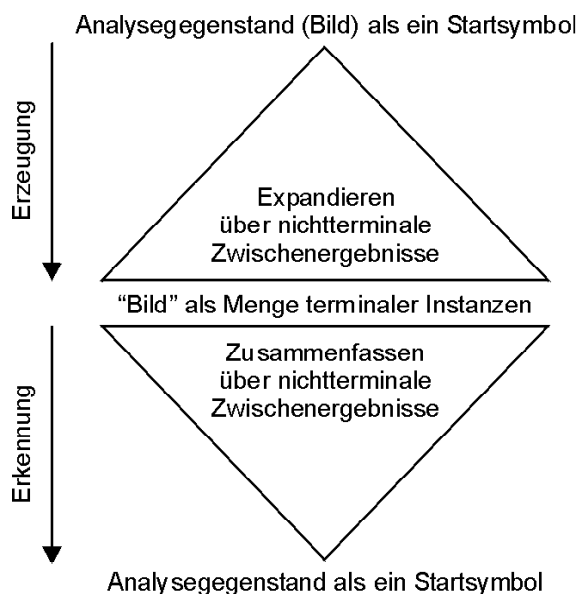


Abb. 2-7: Erzeugungs- und Erkennungsverfahren

### Chomsky-Grammatiken

Chomsky-Grammatiken  $(V_T, V_N, \Sigma, P)$  sind Verfahren zur Generierung und Analyse von Zeichenketten /Cho65/ mit Produktionen eines Semi-Thue-Systems (Gl. 2-5), die gegebenenfalls entsprechend ihres Typs eingeschränkt sind. Dabei wird für die Generierung die Menge der Produktionen  $P$  und für die Analyse die invertierte Relation  $P^{-1}$  zu Grunde gelegt. Es wird auch gefordert, dass in den Zeichenketten der mit der Chomsky-Grammatik  $(V_T, V_N, \Sigma, P)$  generierten bzw. analysierten Sprache  $L_C^g$  bzw.  $L_C^a$  nur terminale Zeichen der Menge  $V_T$  auftreten:

$$\begin{aligned}
 L_C^g &: \wp(V)^3 \times \wp[W^*(V) \times W^*(V)] \rightarrow W^*(V_T) \\
 & (V_T, V_N, \Sigma, P) \mapsto L_C^g(V_T, V_N, \Sigma, P) \\
 L_C^g(V_T, V_N, \Sigma, P) &:= \left\{ a \left| \begin{array}{l} a \in W^*(V_T) \\ \exists \sigma \in \Sigma : \\ \sigma \xrightarrow[STS]{+(V,P)} a \end{array} \right. \right\} \quad (2-13)
 \end{aligned}$$

$$\begin{aligned}
 L_C^a &: \wp(V)^3 \times \wp[W^*(V) \times W^*(V)] \rightarrow W^*(V_T) \\
 & (V_T, V_N, \Sigma, P^{-1}) \mapsto L_C^a(V_T, V_N, \Sigma, P^{-1}) \\
 L_C^a(V_T, V_N, \Sigma, P^{-1}) &:= \left\{ a \left| \begin{array}{l} a \in W^*(V_T) \\ \exists \sigma \in \Sigma : \\ a \xrightarrow[STS]{+(V,P^{-1})} \sigma \end{array} \right. \right\} \quad (2-14)
 \end{aligned}$$

Diese Sprachen entsprechen Szenarien, die von den in Abschnitt 2.3.2 definierten Erzeugungs- und Erkennungsverfahren für Konfigurationen generiert bzw. erkannt werden.

### 2.3.2 Erzeugungs- und Erkennungsverfahren für Konfigurationen

Die bisher betrachtete aus der Literatur bekannte Erzeugung und Erkennung eindimensionaler Zeichenketten wird jetzt für durch Mengen definierte Muster, die Konfigurationen heißen, verallgemeinert.

Zeichenketten-Grammatiken  $(V_T, V_N, \Sigma, P)$  entsprechen dabei Szenario-Beschreibungen  $(\mathcal{I}_T, \mathcal{I}_N, \Sigma, \mathbf{A})$ , die zu den neuen Beiträgen der Arbeit gehören.

$L_C$  ist die Menge aller mit einer Grammatik beschriebenen bzw. generierten und analysierten Sätze. Das Szenario  $S$  ist analog die Menge aller mit einer Szenario-Beschreibung beschriebenen bzw. generierten und analysierten Szenen, die Mengen terminaler Instanzen der in der jeweiligen Szene enthaltenen Objekte sind.

Wenn zweidimensionale Bilder analysiert werden, entsprechen die Szenen im Prinzip den Bildern.

#### Erzeugungsverfahren für Konfigurationen

Erzeugungsverfahren beschreiben von Erkennungsverfahren zu erkennende Konfigurationen. Für die Erzeugung wird die Menge der *generierenden* Assoziation  $\mathbf{A}^g$  mit dem Ersetzungssystem  $(\mathcal{I}_T \cup \mathcal{I}_N, \mathbf{A}^g)$  eingesetzt.

**Direkt ersetzend generierte Konfigurationen** Im einfachsten Fall werden Startsymbole  $\sigma \in \Sigma$  durch eine Folge von Ersetzungen in eine Menge terminaler Instanzen  $\mathcal{R}$  überführt:  $\{\sigma\} \xrightarrow[e]{+\mathbf{A}^g} \mathcal{R}$  (Gl. 2-7, Gl. 2-1, Abbildung 2-8).

$S_g(\mathcal{I}_T, \mathcal{I}_N, \Sigma, \mathbf{A}^g)$  ist die Menge aller erzeugten Konfigurationen und heißt *ersetzend generiertes Szenario*. Da in  $\mathcal{R}$  ausschließlich terminale Instanzen sind und  $\sigma$  als Startsymbol eine nichtterminale Instanz ist, muss mindestens eine direkte Ersetzung erfolgen, so dass die Relation  $\xrightarrow[e]{+\mathbf{A}^g}$  und nicht  $\xrightarrow[e]{*\mathbf{A}^g}$  (Gl. 2-1) betrachtet wird:

$$\begin{aligned}
 S_g : \wp(\mathcal{I})^3 \times \wp[\wp(\mathcal{I})^2] &\rightarrow \wp[\wp(\mathcal{I}_T)], \text{ mit } \mathcal{I} = \mathcal{I}_T \cup \mathcal{I}_N \\
 (\mathcal{I}_T, \mathcal{I}_N, \Sigma, \mathbf{A}^g) &\mapsto S_g(\mathcal{I}_T, \mathcal{I}_N, \Sigma, \mathbf{A}^g) \\
 S_g(\mathcal{I}_T, \mathcal{I}_N, \Sigma, \mathbf{A}^g) &:= \left\{ \mathcal{R} \left| \begin{array}{l} \mathcal{R} \subseteq \mathcal{I}_T \\ \exists \sigma \in \Sigma : \\ \{\sigma\} \xrightarrow[e]{+\mathbf{A}^g} \mathcal{R} \end{array} \right. \right\} \quad (2-15)
 \end{aligned}$$

Die Menge  $S_g$  entspricht bei Zeichenketten-Ersetzungssystemen der erzeugten Sprache  $L_C^g$  mit eindimensionalen Zeichenketten (Gl. 2-13).

**Obermengen generierter Konfigurationen** Sobald eine Konfiguration terminaler Instanzen  $\mathcal{T}$  nur teilweise analysiert wird, sind konzeptionell alle Mengen von Konfigurationen zu berücksichtigen, in deren Konfigurationen  $\mathcal{T}$  generierte Konfigurationen  $\mathcal{R}$  enthalten sind.  $S_g^O(\mathcal{I}_T, \mathcal{I}_N, \Sigma, \mathbf{A}^g)$  enthält diese Obermengen und

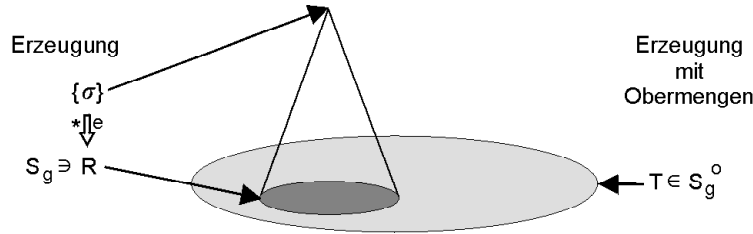


Abb. 2-8: Erzeugung von Konfigurationen

heißt *ersetzend mit Obermengen generiertes Szenario* (Abbildung 2-8):

$$\begin{aligned}
 S_g^O &: \wp(\mathcal{I})^3 \times \wp[\wp(\mathcal{I})^2] \rightarrow \wp[\wp(\mathcal{I}_T)], \text{ mit } \mathcal{I} = \mathcal{I}_T \cup \mathcal{I}_N \\
 &(\mathcal{I}_T, \mathcal{I}_N, \Sigma, \mathbf{A}^g) \mapsto S_g^O(\mathcal{I}_T, \mathcal{I}_N, \Sigma, \mathbf{A}^g) \\
 S_g^O(\mathcal{I}_T, \mathcal{I}_N, \Sigma, \mathbf{A}^g) &:= \left\{ \mathcal{T} \mid \exists \mathcal{R} \in S_g(\mathcal{I}_T, \mathcal{I}_N, \Sigma, \mathbf{A}^g) : \right. \\
 &\quad \left. \mathcal{R} \subseteq \mathcal{T} \right\}
 \end{aligned} \tag{2-16}$$

### Erkennungsverfahren für Konfigurationen

Die Erkennung von Konfigurationen kann *ersetzend* oder *zuordnend* sowie *vollständig* oder *teilweise* erfolgen. Für die Erkennung wird die Menge der *analysierenden* Assoziationen  $\mathbf{A}^a$  mit dem Paar  $(\mathcal{I}_T \cup \mathcal{I}_N, \mathbf{A}^a)$ , das ersetzend ein Ersetzungssystem und zuordnend ein Zuordnungssystem ist, verwendet. Einige der anschließend definierten Verfahren können mit den im Abschnitt 3.3 angegebenen Algorithmen realisiert werden. Die zuordnenden Erkennungsverfahren können zusätzlich mit Zuordnungsautomaten realisiert werden.

**Ersetzende Erkennungsverfahren für Konfigurationen** Im einfachsten Fall werden alle Instanzen der zu analysierenden Menge der terminalen Instanzen  $\mathcal{R}$  durch eine Folge von Ersetzungen in ein Startsymbol  $\sigma \in \Sigma$  überführt:  $\mathcal{R} \xrightarrow[e]{+\mathbf{A}^g} \{\sigma\}$  (Gl. 2-7, Gl. 2-1, Abbildung 2-9).  $S_a^{ve}(\mathcal{I}_T, \mathcal{I}_N, \Sigma, \mathbf{A}^a)$  ist die Menge aller erkannten Konfigurationen und heißt *vollständig ersetzend erkanntes Szenario*:

$$\begin{aligned}
 S_a^{ve} &: \wp(\mathcal{I})^3 \times \wp[\wp(\mathcal{I})^2] \rightarrow \wp[\wp(\mathcal{I}_T)], \text{ mit } \mathcal{I} = \mathcal{I}_T \cup \mathcal{I}_N \\
 &(\mathcal{I}_T, \mathcal{I}_N, \Sigma, \mathbf{A}^a) \mapsto S_a^{ve}(\mathcal{I}_T, \mathcal{I}_N, \Sigma, \mathbf{A}^a) \\
 S_a^{ve}(\mathcal{I}_T, \mathcal{I}_N, \Sigma, \mathbf{A}^a) &:= \left\{ \mathcal{R} \mid \begin{array}{l} \exists \sigma \in \Sigma : \\ \mathcal{R} \xrightarrow[e]{+\mathbf{A}^a} \{\sigma\} \end{array} \right\}
 \end{aligned} \tag{2-17}$$

Die Menge  $S_a^{ve}$  entspricht bei Zeichenketten-Ersetzungssystemen der analysierten Sprache  $L_C^a$  mit eindimensionalen Zeichenketten (Gl. 2-14).

Wenn nicht alle terminalen Instanzen der zu analysierenden Menge  $\mathcal{T}$  zu einem Startsymbol  $\sigma \in \Sigma$  ersetzt werden müssen, genügt es, wenn die Analyse

ersetzend eine Konfiguration  $Q$  erzeugt, in der ein Startsymbol  $\sigma$  enthalten ist.  $S_a^{te} = (\mathcal{I}_T, \mathcal{I}_N, \Sigma, \mathbf{A}^a)$  ist die Menge aller erkannten Konfigurationen und heißt *teilweise ersetzend erkanntes Szenario*:

$$S_a^{te} : \wp(\mathcal{I})^3 \times \wp[\wp(\mathcal{I})^2] \rightarrow \wp[\wp(\mathcal{I}_T)], \text{ mit } \mathcal{I} = \mathcal{I}_T \cup \mathcal{I}_N$$

$$(\mathcal{I}_T, \mathcal{I}_N, \Sigma, \mathbf{A}^a) \mapsto S_a^{te}(\mathcal{I}_T, \mathcal{I}_N, \Sigma, \mathbf{A}^a)$$

$$S_a^{te}(\mathcal{I}_T, \mathcal{I}_N, \Sigma, \mathbf{A}^a) := \left\{ \mathcal{T} \left| \begin{array}{l} \exists Q \subseteq \mathcal{I} : \\ \mathcal{T} \xrightarrow[e]{+\mathbf{A}^a} Q \\ \exists \sigma \in \Sigma : \\ \sigma \in Q \end{array} \right. \right\} \quad (2-18)$$

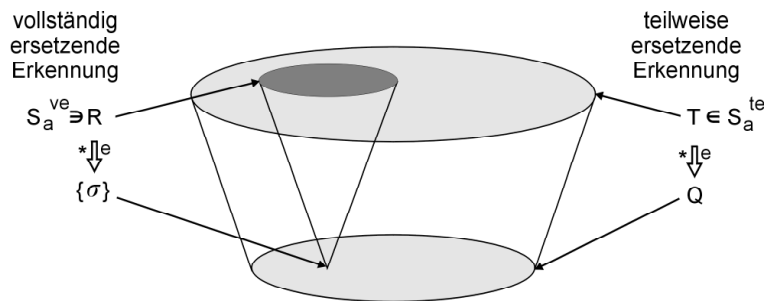


Abb. 2-9: Ersetzende Erkennung von Konfigurationen

Abbildung 2-9 zeigt das Prinzip der vollständig und teilweise ersetzenden Erkennung.

**Zuordnende Erkennungsverfahren für Konfigurationen** Eine Alternative zu ersetzenden Verfahren sind zuordnende Erkennungsverfahren, die zuordnend, d.h. mit Speicherung der Zwischenergebnisse, analysieren. In /Mich98a/ heißen vergleichbare Verfahren *kumulierend*.

Zur *teilweise zuordnenden Erkennung* werden Konfigurationen  $\mathcal{T}$  mit der Menge der analysierenden Assoziationen  $\mathbf{A}^a$  und der Relation  $\xrightarrow[z]{+\mathbf{A}^a}$  (Gl. 2-8, Gl. 2-1) Konfigurationen  $Q$  zugeordnet, in denen ein Startsymbol  $\sigma \in \Sigma$  enthalten ist.  $S_a^{tz}(\mathcal{I}_T, \mathcal{I}_N, \Sigma, \mathbf{A}^a)$  ist die Menge aller erkannten Konfigurationen und heißt *teilweise zuordnend erkanntes Szenario*:

$$S_a^{tz} : \wp(\mathcal{I})^3 \times \wp[\wp(\mathcal{I})^2] \rightarrow \wp[\wp(\mathcal{I}_T)], \text{ mit } \mathcal{I} = \mathcal{I}_T \cup \mathcal{I}_N$$

$$(\mathcal{I}_T, \mathcal{I}_N, \Sigma, \mathbf{A}^a) \mapsto S_a^{tz}(\mathcal{I}_T, \mathcal{I}_N, \Sigma, \mathbf{A}^a)$$

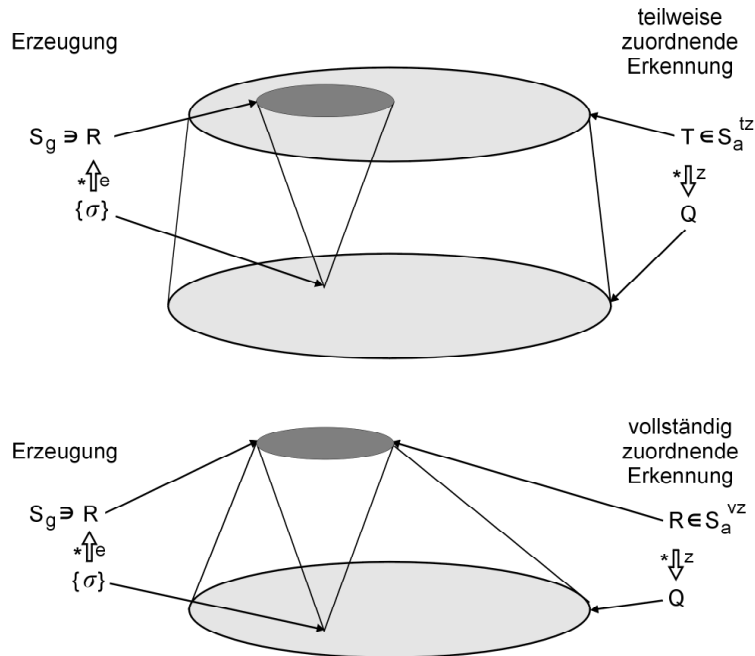
$$S_a^{tz}(\mathcal{I}_T, \mathcal{I}_N, \Sigma, \mathbf{A}^a) := \left\{ \mathcal{T} \left| \begin{array}{l} \exists Q \subseteq \mathcal{I} : \\ \mathcal{T} \xrightarrow[z]{+\mathbf{A}^a} Q \\ \exists \sigma \in \Sigma : \\ \sigma \in Q \end{array} \right. \right\} \quad (2-19)$$

Mit der Relation

$$T^{tz}(\mathcal{I}_T, \mathcal{I}_N, \Sigma, \mathbf{A}^a) := \left\{ (\mathcal{T}, \sigma) \left| \begin{array}{l} \mathcal{T} \subseteq \mathcal{I}_T \\ \exists Q \subseteq \mathcal{I}_T \cup \mathcal{I}_N : \\ \mathcal{T} \xrightarrow[\quad z]{+\mathbf{A}^a} Q \\ \sigma \in \Sigma \cap Q \end{array} \right. \right\} \quad (2-20)$$

lässt sich Gleichung 2-19 wie folgt schreiben:

$$S_a^{tz}(\mathcal{I}_T, \mathcal{I}_N, \Sigma, \mathbf{A}^a) = \left\{ \mathcal{T} \left| \begin{array}{l} \exists \sigma \in \Sigma : \\ (\mathcal{T}, \sigma) \in T^{tz}(\mathcal{I}_T, \mathcal{I}_N, \Sigma, \mathbf{A}^a) \end{array} \right. \right\}$$



**Abb. 2-10:** Zuordnende Erkennung von Konfigurationen  
oben: teilweise zuordnend mit Hintergrund-Instanzen  $\mathcal{T} \setminus \mathcal{R}$   
unten: vollständig zuordnend

Sobald Instanzen als Ableitungen  $\mathcal{I}_A$  (vgl. Seite 25, Gl. B-54) beschrieben werden, ist die *vollständig zuordnende Erkennung* definiert. Dazu werden, wie bei der teilweise zuordnenden Erkennung, Konfigurationen  $\mathcal{R}$  mit der Menge der analysierenden Assoziationen  $\mathbf{A}^a$  und der Relation  $\xrightarrow[\quad z]{+\mathbf{A}^a}$  (Gl. 2-8, Gl. 2-1) Konfigurationen zugeordnet, in denen ein Startsymbol  $\sigma \in \Sigma$  enthalten ist. Zusätzlich ist die zu analysierende Konfiguration  $\mathcal{R}$  die Menge der terminalen Instanzen der Knoten  $AG_N(\sigma)$  der Ableitungsgraphen von  $\sigma$  (Gl. B-59).  $S_a^{vz}(\mathcal{I}_T, \mathcal{I}_N, \Sigma, \mathbf{A}^a)$  ist die Menge aller erkannten Konfigurationen und heißt *vollständig zuordnend erkanntes Szenario*.

rio.

$$\begin{aligned}
S_a^{vz} &: \wp(\mathcal{I}_A)^3 \times \wp[\wp(\mathcal{I}_A)^2] \rightarrow \wp[\wp(\mathcal{I}_T)], \text{ mit } \mathcal{I}_A = \mathcal{I}_T \cup \mathcal{I}_N \\
&(\mathcal{I}_T, \mathcal{I}_N, \Sigma, \mathbf{A}^a) \mapsto S_a^{vz}(\mathcal{I}_T, \mathcal{I}_N, \Sigma, \mathbf{A}^a) \\
S_a^{vz}(\mathcal{I}_T, \mathcal{I}_N, \Sigma, \mathbf{A}^a) &:= \left\{ \mathcal{R} \left| \begin{array}{l} \exists \sigma \in \Sigma : \\ (\mathcal{R}, \sigma) \in T^{tz}(\mathcal{I}_T, \mathcal{I}_N, \Sigma, \mathbf{A}^a) \\ \mathcal{R} = AG_N(\sigma) \cap \mathcal{I}_T \end{array} \right. \right\} \quad (2-21)
\end{aligned}$$

**Zuordnende Erkennungsverfahren für Konfigurationen mit Ergebnisverifikation** Zuordnend können Ergebnisse entstehen, die ersetzend nicht entstehen können und u.U. auch nicht der Intension des Entwicklers entsprechen. Zur Bestimmung der im Sinne der Analyseaufgabe richtigen Endergebnisse können sowohl teilweise zuordnend ( $S_a^{tz'}$ ) als auch vollständig zuordnend ( $S_a^{vz'}$ ) die Ableitungsgraphen der zugeordneten Startsymbole überprüft werden:

$$\begin{aligned}
S_a^{tz'} &: \wp(\mathcal{I}_A)^3 \times \wp[\wp(\mathcal{I}_A)^2] \rightarrow \wp[\wp(\mathcal{I}_T)], \text{ mit } \mathcal{I}_A = \mathcal{I}_T \cup \mathcal{I}_N \\
&(\mathcal{I}_T, \mathcal{I}_N, \Sigma, \mathbf{A}^a) \mapsto S_a^{tz'}(\mathcal{I}_T, \mathcal{I}_N, \Sigma, \mathbf{A}^a) \\
S_a^{tz'}(\mathcal{I}_T, \mathcal{I}_N, \Sigma, \mathbf{A}^a) &:= \left\{ \mathcal{R} \left| \begin{array}{l} \exists \sigma \in \Sigma : \\ (\mathcal{R}, \sigma) \in T^{tz}(\mathcal{I}_T, \mathcal{I}_N, \Sigma, \mathbf{A}^a) \\ AG(\sigma) \text{ entspricht der Analyseaufgabe} \end{array} \right. \right\} \quad (2-22)
\end{aligned}$$

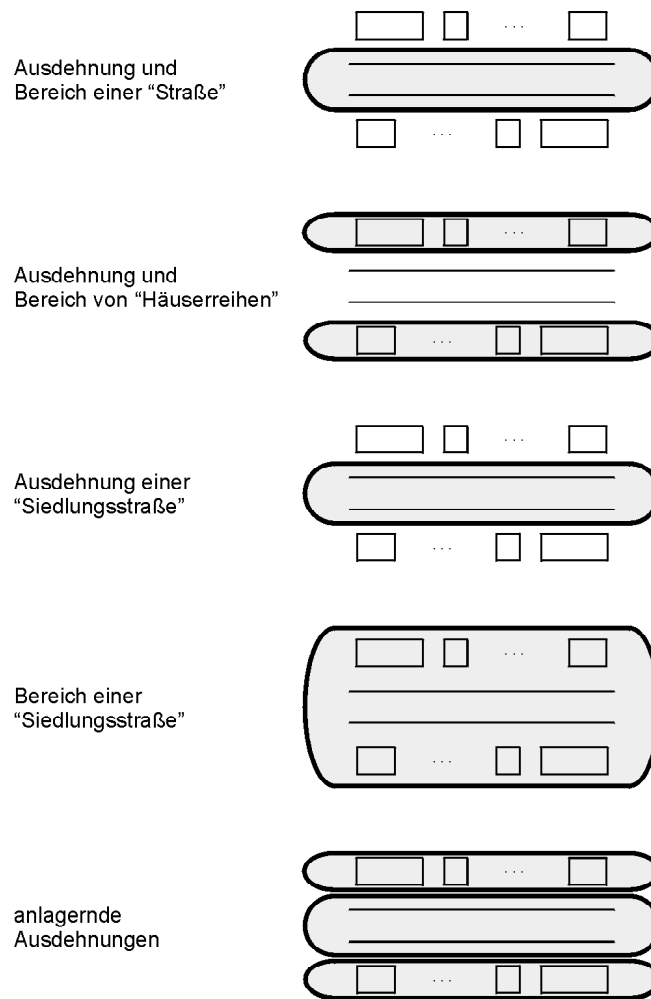
$$\begin{aligned}
S_a^{vz'} &: \wp(\mathcal{I}_A)^3 \times \wp[\wp(\mathcal{I}_A)^2] \rightarrow \wp[\wp(\mathcal{I}_T)], \text{ mit } \mathcal{I}_A = \mathcal{I}_T \cup \mathcal{I}_N \\
&(\mathcal{I}_T, \mathcal{I}_N, \Sigma, \mathbf{A}^a) \mapsto S_a^{vz'}(\mathcal{I}_T, \mathcal{I}_N, \Sigma, \mathbf{A}^a) \\
S_a^{vz'}(\mathcal{I}_T, \mathcal{I}_N, \Sigma, \mathbf{A}^a) &:= \left\{ \mathcal{R} \left| \begin{array}{l} \exists \sigma \in \Sigma : \\ (\mathcal{R}, \sigma) \in T^{tz}(\mathcal{I}_T, \mathcal{I}_N, \Sigma, \mathbf{A}^a) \\ \mathcal{R} = AG_N(\sigma) \cap \mathcal{I}_T \\ AG(\sigma) \text{ entspricht der Analyseaufgabe} \end{array} \right. \right\} \quad (2-23)
\end{aligned}$$

**Zuordnende Erkennungsverfahren für Konfigurationen mit beschreibenden Merkmalen und Ausdehnungen** Wenn für die Instanzen  $\gamma \in \mathcal{I}$  Ausdehnungen  $A(\gamma)$  (Gl. B-76) definiert sind, wird davon ausgegangen, dass in Gleichung 2-22 und 2-23 die Analyse des Ableitungsgraphen von  $\sigma$  zur Bestimmung gültiger Konfigurationen durch die Überprüfung der Ausdehnung der Instanzen von  $\mathcal{R}$  ersetzt werden kann.

Arbeitshypothese ist, dass mit anlagernden Assoziationen (Abbildung 2-11 und Gl. B-77) bei allen direkten Zuordnungen der Zuordnung  $\mathcal{R} \xrightarrow[z]{+\mathbf{A}^a} \mathcal{Q}$  die Ausdehnungen der Zwischenergebnisse "richtig" zusammengesetzt werden und statt  $S_a^{tz'}$  wieder  $S_a^{tz}$  (bzw. statt  $S_a^{vz'}$  wieder  $S_a^{vz}$ ) für die Analyse verwendet werden kann:

$$\mathbf{A}^a \text{ ist anlagernd} \rightarrow \left\{ \begin{array}{l} S_a^{tz'}(\mathcal{I}_T, \mathcal{I}_N, \Sigma, \mathbf{A}^a) = S_a^{tz}(\mathcal{I}_T, \mathcal{I}_N, \Sigma, \mathbf{A}^a) \\ S_a^{vz'}(\mathcal{I}_T, \mathcal{I}_N, \Sigma, \mathbf{A}^a) = S_a^{vz}(\mathcal{I}_T, \mathcal{I}_N, \Sigma, \mathbf{A}^a) \end{array} \right. \quad (2-24)$$

Für die Bildanalyse wird daher in der Regel  $S_a^{tz}$  (Gl. 2-19) mit anlagernden Assoziationen eingesetzt. Die Untersuchungen hierzu sind noch nicht abgeschlossen und werden im Ausblick (Abschnitt 5.2) aufgegriffen.



**Abb. 2-11:** Ausdehnung und Bereich von Instanzen sowie anlagernde Assoziationen

### Verfahren zur Erzeugung und Erkennung von Konfigurationen mit Konzepten und Ableitungen

Die eine Analyse beschreibende Menge der Assoziationen  $\mathbf{A}$  wird zweckmäßigerweise mit Produktionen einer Menge  $\mathbf{P}$ , die wesentlich weniger Elemente als  $\mathbf{A}$  hat, und einer Instanzierungsfunktion  $i$  definiert (Gl. B-46). Die Produktionen beschreiben dabei, wie elementare Konzepte durch abstrakte Konzepte beschrieben werden.

Abschnitt B.1.4 definiert Zuordnungssysteme mit Konzepten, auf denen die Erkennung mit Konzepten aufbaut und die auf Seite 24 eingeführt werden. Szenario-Beschreibungen mit Konzepten haben die Form  $(\mathcal{I}_T, \mathcal{I}_N, \Sigma, \mathbf{P}, i)$ . Die zugehörige Szenario-Beschreibung mit Instanzen ist  $(\mathcal{I}_T, \mathcal{I}_N, \Sigma, \mathbf{A}_K(\mathcal{I}_T \cup \mathcal{I}_N, \mathbf{P}, i))$  mit (Gl. B-



46)

$$\mathbf{A}_K(\mathcal{I}, \mathbf{P}, i) := \left\{ (\mathcal{X}, \mathcal{Y}) \left| \begin{array}{l} \mathcal{X} \subseteq \mathcal{I} \\ \exists (\mathcal{A}, \mathcal{B}) \in \mathbf{P} : \\ i(\mathcal{B}|\mathcal{A} \mapsto \mathcal{X}) \neq \emptyset \\ \mathcal{Y} = i(\mathcal{B}|\mathcal{A} \mapsto \mathcal{X}) \end{array} \right. \right\}.$$

Die in Abschnitt B.1.5 definierten und auf Seite 25 eingeführten Zuordnungssysteme mit Ableitungen gestatten, Ableitungen für Erkennungsaufgaben einzusetzen und so Produktionsnetze (Gl. B-58) sowie Ableitungs- und Beschreibungsgraphen (Gl. B-59 ff.) zu nutzen. Szenario-Beschreibungen mit Ableitungen haben die Form  $(\mathbf{I}, \mathcal{I}_T, \Sigma, \mathbf{P}, i)$ . Die zugehörige Szenario-Beschreibung mit Konzepten ist  $(\mathcal{I}_T, \mathcal{I}_A \setminus \mathcal{I}_T, \Sigma, \mathbf{P}, i)$  mit  $\mathcal{I}_A := \mathcal{I}_M \times \mathbb{IN} \times P_M$  (Gl. B-54) und  $\mathcal{I}_M := M_1 \times M_2 \times \dots$  (Gl. B-51) sowie  $\mathbf{I} = (M_1, M_2, \dots)$  und  $P_M := \wp(\mathbb{IN})$  (Gl. B-53).

## 2.4 Erkennung erzeugter Konfigurationen

Erkennungsverfahren erkennen von ersetzenden Erzeugungsverfahren generierte Konfigurationen. Die Erkennung kann ersetzend oder zuordnend erfolgen. Zu analysierende Konfigurationen werden konzeptionell mit Verfahren nach Gleichung 2-15 erzeugt oder sind Obermengen generierter Konfigurationen (Gl. 2-16). In diesem Abschnitt werden die mit dieser Arbeit eingeführten neuen Verfahren für Analyseaufgaben diskutiert. Beispiele zur Bildanalyse mit Produktionsnetzen stehen im Abschnitt A.3 des Anhangs.

### 2.4.1 Ersetzende Erkennung erzeugter Konfigurationen

Die ersetzende Erkennung präzisiert das Konzept der Erkennung von Mustern, die mit ungeordneten Mengen beschrieben werden. Der Zusammenhang zwischen Erzeugung und Erkennung ist konzeptionell durch die Menge der generierenden Assoziation  $\mathbf{A}^g$  gegeben: Zur Erzeugung wird  $\mathbf{A}^g$  direkt verwendet (Gl. 2-15). Zur Erkennung wird die invertierte Relation  $\mathbf{A}^a = (\mathbf{A}^g)^{-1}$  eingesetzt.

#### Vollständig ersetzende Analyse

Vollständig ersetzende Analysen ersetzen zu analysierende Mengen terminaler Instanzen zu Startsymbolen. Dabei müssen nach der Analyse alle terminalen Instanzen ersetzt sein.

**Erkennung, wenn  $\mathbf{A}^g$  einfach assoziierend generiert** Mit einfach assoziierend generierenden Assoziationen (vgl. Seite 25, Gl. B-72) kann jede generierte Konfiguration ersetzend erkannt werden (Lemma B.75):

$$\begin{aligned} & \mathbf{A}^g \text{ ist einfach assoziierend} \rightarrow \\ & \rightarrow S_g(\mathcal{I}_T, \mathcal{I}_N, \Sigma, \mathbf{A}^g) \subseteq S_a^{ve}(\mathcal{I}_T, \mathcal{I}_N, \Sigma, (\mathbf{A}^g)^{-1}) \end{aligned}$$

In diesem Fall ist es zulässig, dass die Menge der Assoziationen  $(\mathbf{A}^g)^{-1}$  mehrfach assoziierend analysiert und analysierte terminale Konfigurationen  $\mathcal{R} \in S_a^{ve}(\mathcal{I}_T, \mathcal{I}_N, \Sigma, (\mathbf{A}^g)^{-1})$  nicht generiert werden können:  $\mathcal{R} \notin S_g(\mathcal{I}_T, \mathcal{I}_N, \Sigma, \mathbf{A}^g)$ .

**Erkennung, wenn  $\mathbf{A}^a$  einfach assoziierend analysiert** Mit einfach assoziierend analysierenden Assoziationen  $\mathbf{A}^a = (\mathbf{A}^g)^{-1}$  kann jede analysierte Konfiguration ersetzend generiert werden (Lemma B.76):

$$\begin{aligned} & (\mathbf{A}^g)^{-1} \text{ ist einfach assoziierend} \rightarrow \\ & \rightarrow S_a^{ve}(\mathcal{I}_T, \mathcal{I}_N, \Sigma, (\mathbf{A}^g)^{-1}) \subseteq S_g(\mathcal{I}_T, \mathcal{I}_N, \Sigma, \mathbf{A}^g) \end{aligned}$$

In diesem Fall ist es zulässig, dass die Menge der Assoziationen  $\mathbf{A}^g$  mehrfach assoziierend generiert und generierte terminale Konfigurationen  $\mathcal{R} \in S_g(\mathcal{I}_T, \mathcal{I}_N, \Sigma, \mathbf{A}^g)$  nicht analysiert werden können:  $\mathcal{R} \notin S_a^{ve}(\mathcal{I}_T, \mathcal{I}_N, \Sigma, (\mathbf{A}^g)^{-1})$ .

**Erkennung, wenn  $\mathbf{A}^g$  und  $\mathbf{A}^a = (\mathbf{A}^g)^{-1}$  einfach assoziierend sind** Genau dann, wenn  $\mathbf{A}^g$  und  $\mathbf{A}^a = (\mathbf{A}^g)^{-1}$  einfach assoziierend sind, werden ersetzend die Muster erkannt, die mit den korrespondierenden Erzeugungsverfahren generiert werden (Satz B.77):

$$\begin{aligned} & \mathbf{A}^g \text{ und } (\mathbf{A}^g)^{-1} \text{ sind einfach assoziierend} \leftrightarrow \\ & \leftrightarrow S_g(\mathcal{I}_T, \mathcal{I}_N, \Sigma, \mathbf{A}^g) = S_a^{ve}(\mathcal{I}_T, \mathcal{I}_N, \Sigma, (\mathbf{A}^g)^{-1}) \end{aligned}$$

### Teilweise ersetzende Analyse

Teilweise ersetzende Analysen fassen Teilmengen zu analysierender terminaler Instanzen zu Startsymbolen zusammen. Es wird also nicht mehr gefordert, dass nach der Analyse alle terminalen Instanzen ersetzt sind.

Wenn  $\mathbf{A}^g$  und  $(\mathbf{A}^g)^{-1}$  einfach assoziierend sind, ist die Menge der teilweise ersetzend analysierten Konfigurationen  $S_a^{te}(\mathcal{I}_T, \mathcal{I}_N, \Sigma, (\mathbf{A}^g)^{-1})$  gleich der Obermenge der generierten Konfiguration  $S_g^O(\mathcal{I}_T, \mathcal{I}_N, \Sigma, \mathbf{A}^g)$  (Satz B.78):

$$\begin{aligned} & \mathbf{A}^g \text{ und } (\mathbf{A}^g)^{-1} \text{ sind einfach assoziierend} \rightarrow \\ & \rightarrow S_g^O(\mathcal{I}_T, \mathcal{I}_N, \Sigma, \mathbf{A}^g) = S_a^{te}(\mathcal{I}_T, \mathcal{I}_N, \Sigma, (\mathbf{A}^g)^{-1}) \end{aligned}$$

Die teilweise ersetzend erkannten Konfigurationen  $\mathcal{R}$  von  $S_g^O(\mathcal{I}_T, \mathcal{I}_N, \Sigma, \mathbf{A}^g)$  sind als Obermengen definiert und daher extrem weit gefasst (Gl. 2-16). Gleichung 2-16 sagt nur aus, dass jede Teilmenge von  $\mathcal{I}_T$ , in der eine Konfiguration  $\mathcal{R} \in S_g(\mathcal{I}_T, \mathcal{I}_N, \Sigma, \mathbf{A}^g)$  enthalten ist, erkannt wird, so dass auch Mengen terminaler Instanzen  $\mathcal{T}$ , die keinen Bezug zu Analyseaufgaben haben, in der Menge der erkannten Konfigurationen enthalten sein können. Beispielsweise ist die Zusammenfassung aller vorstellbaren Szenen [d. h. die Vereinigung der Konfigurationen aller möglichen Szenarien  $S_g(\mathcal{I}_T, \mathcal{I}_N, \Sigma_2, \mathbf{A}_2^g)$ ] in jeder Menge  $S_g^O(\mathcal{I}_T, \mathcal{I}_N, \Sigma_1, \mathbf{A}_1^g)$  enthalten (Satz B.79):

$$\begin{aligned} & \forall \mathbf{A}_1^g \subseteq \wp(\mathcal{I}_A) \times \wp(\mathcal{I}_A) : \forall \Sigma_1 \subseteq \mathcal{I}_N : \\ & \bigcup_{\Sigma_2 \subseteq \mathcal{I}_N} \left[ \bigcup_{\mathbf{A}_2^g \subseteq \wp(\mathcal{I}_A) \times \wp(\mathcal{I}_A)} \left( \bigcup_{\mathcal{R} \in S_g(\mathcal{I}_T, \mathcal{I}_N, \Sigma_2, \mathbf{A}_2^g)} \mathcal{R} \right) \right] \in S_g^O(\mathcal{I}_T, \mathcal{I}_N, \Sigma_1, \mathbf{A}_1^g) \end{aligned}$$

### 2.4.2 Analyse mit Konfigurationen-Erkennungsverfahren

Die strukturelle Analyse mit Produktionsnetzen trennt zu erkennende Objekte vom Hintergrund<sup>6</sup> und beschreibt die Objekte.

Abbildung 2-12 erläutert dies anhand der Analyse des Brückenbildes Abbildung 1-7. Die Analyse beginnt mit den von einem Vorverarbeitungsprogramm generierten Geradenstücken (Abbildung 2-12a). Aus diesen werden über die Zwischenergebnisse "Straßen" bzw. "Streifen" (Abbildung 2-12b) zuordnend "Brücken" (Kreuze in Abbildung 2-12c) zusammengesetzt. Aus den 5 in Abbildung 2-12c zugeordneten Brücken<sup>7</sup> wird die beste Brücke ausgewählt und in Abbildung 2-12d als Analyseergebnis übernommen. Zu dieser Brücke gehören die in den Abbildungen 2-12e und 2-12f dargestellten Teilobjekte.

Die aus  $\mathcal{T}$  (Geradenstücke in Abbildung 2-12a) teilweise ersetzend erzeugte Menge  $\mathcal{Q}$  (Abbildung 2-9 rechts mit  $\mathcal{T} \xrightarrow[e]{+\mathbf{A}^a} \mathcal{Q}, \exists \sigma \in \Sigma : \sigma \in \mathcal{Q}$ ) enthält das Brückenkreuz in Abbildung 2-12d als Startsymbol  $\sigma$  und die Geradenstücke in Abbildung 2-12a, die nicht in Abbildung 2-12f enthalten sind. Das vollständig ersetzend aus  $\mathcal{R}$  (Geradenstücke in Abbildung 2-12f) erzeugte Startsymbol  $\sigma$  (Abbildung 2-9 links mit  $\mathcal{R} \xrightarrow[e]{+\mathbf{A}^a} \{\sigma\}$ ) ist das Brückenkreuz in Abbildung 2-12d. Die aus  $\mathcal{T}$  (Abbildung 2-12a) teilweise zuordnend erzeugte Menge  $\mathcal{Q}$  (Abbildung 2-10 oben mit  $\mathcal{T} \xrightarrow[z]{+\mathbf{A}^a} \mathcal{Q}, \exists \sigma \in \Sigma : \sigma \in \mathcal{Q}$ ) enthält alle terminalen und nichtterminalen Instanzen u. a. die Objekte der Abbildungen 2-12a bis 2-12c.

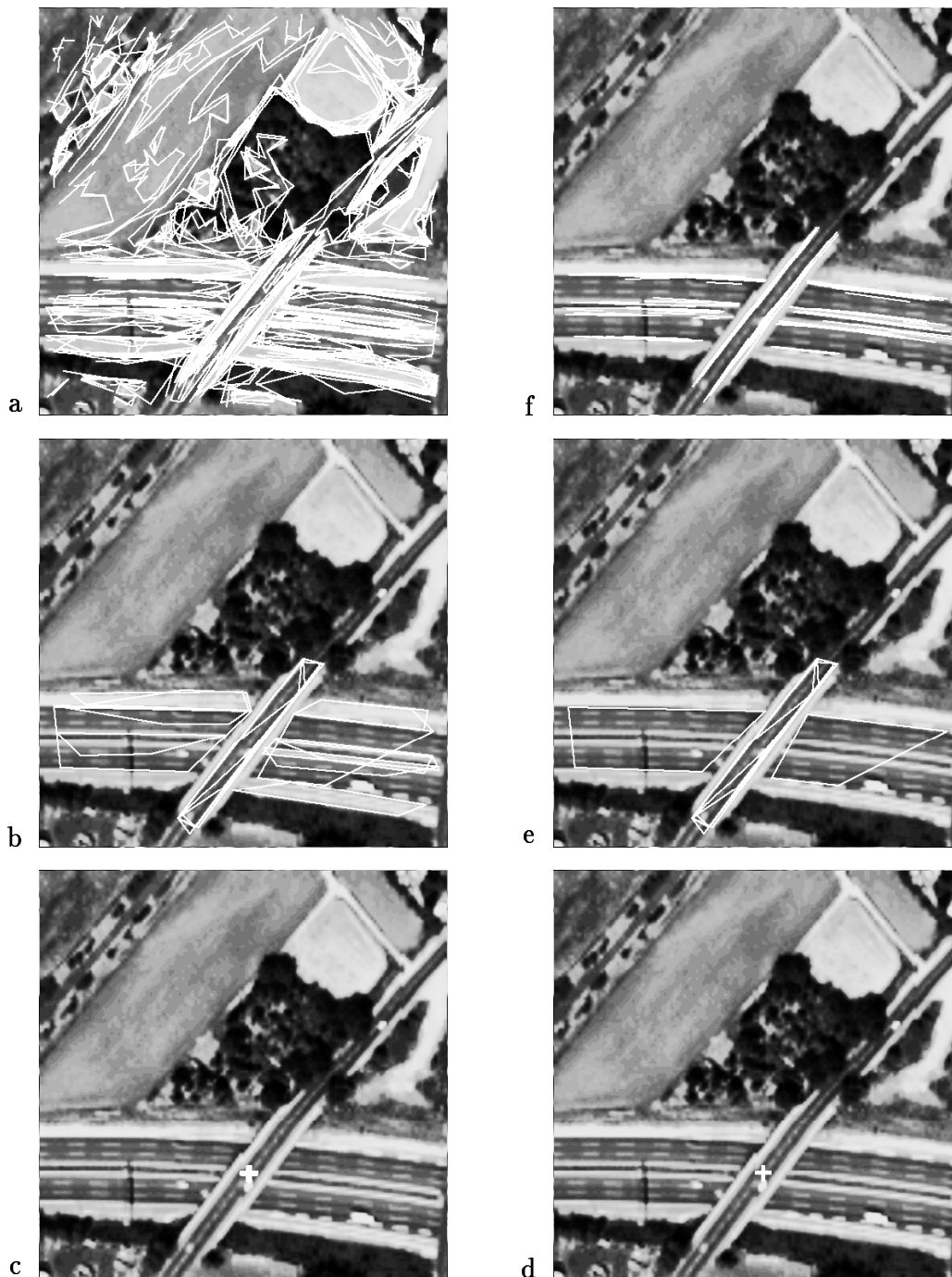
Mit Abbildung 2-12 soll nur das Analyseprinzip erläutert werden. Die eigentlichen Analyseergebnisse werden im Anhang Abschnitt A.3 erläutert.

### Trennung von Objekt und Hintergrund

Die *vollständig ersetzende* Analyse setzt voraus, dass alle zu analysierenden terminalen Instanzen zu einem Startsymbol zusammengesetzt werden. Die Trennung von Objekt und Hintergrund muss in diesem Fall also vor der Analyse, der nur die terminalen Instanzen der zu erkennenden Objekte übergeben werden dürfen, erfolgen. Für die hier betrachteten Bildanalyseaufgaben müßte daher der in vielen Fällen entscheidende Analyseschritt zur Objekt-Hintergrund-Trennung der strukturellen Analyse vorgeschaltet werden, so dass die danach durchgeführte *vollständig ersetzende* strukturelle Analyse nur noch eine untergeordnete Bedeutung hätte. Bezogen auf Abbildung 2-12 hieße dies, dass vor der strukturellen Analyse aus Abbildung 2-12a zuerst Abbildung 2-12f erzeugt werden müßte. Die strukturelle Analyse würde dann aus allen Geradenstücken der Abbildung 2-12f die Brücke in Abbildung 2-12d zusammensetzen.

<sup>6</sup>Hintergrund ist hier die Menge der terminalen Instanzen, die nicht zum Objekt gehört.

<sup>7</sup>Das Kreuz in Abbildung 2-12c entstand durch Überlagerung von 5 Kreuzen und ist deshalb dicker als in Abbildung 2-12d.



**Abb. 2-12:** Analyse mit Konfigurationen Erkennungsverfahren  
 links: alle terminalen und zugeordneten Instanzen  
 rechts: Instanzen, die zum Analyseergebnis gehören  
 oben: Geradenstücke  
 Mitte: Straßen  
 unten: Brücken

Bei der *teilweise ersetzenden* Analyse dürfen nach der Analyse die terminalen Instanzen der Hintergrund-Objekte übrig bleiben. Die Trennung von Objekt und Hintergrund erfordert also teilweise ersetzende Analysen. Abbildung 2-12f zeigt z. B. die durch die strukturelle Analyse von Hintergrund getrennten terminalen Instanzen des Analyseergebnisses.

### Objekterkennung

Die teilweise ersetzende Analyse trennt Objekte vom Hintergrund und beschreibt die Objekte. Dabei können vom Prinzip her Bilder mit beliebigen Inhalten analysiert werden. Um die ab Seite 46 diskutierten Probleme zu meistern, benötigt man Aussagen über die bei speziellen Bildern zu erwartende Qualität der Analyseergebnisse. Dazu müssen die Inhalte der zu analysierenden Bilder eingeschränkt werden. Eine Möglichkeit hierzu ist, die Inhalte bereits analysierter Bilder zusammen mit der jeweils erreichten Analysequalität zu beschreiben und neu zu analysierende Bilder anhand dieser Beschreibungen auszuwählen.

**Trainings- und Testdaten** In der Praxis wird so vorgegangen, dass mit einem ersten Datensatz (Trainingsdaten) ein Analyseverfahren bzw. eine Menge von Assoziationen entwickelt und optimiert wird und dann mit einem zweiten Datensatz (Testdaten), der eine ausreichend große Anzahl von Bildern enthält, die erreichte Qualität überprüft wird. Nachdem Training und Test abgeschlossen sind, werden beim Einsatz der Analyseprogramme die auszuwertenden Bilder wie Testdaten behandelt.

Die zu analysierenden Daten werden mit Generierungsverfahren  $(\mathcal{I}_T, \mathcal{I}_N, \Sigma_S, \mathbf{A}^g)$  beschrieben. Die Startsymbole  $\sigma_S \in \Sigma_S$  stehen für die unterschiedlichen Szenen. Die spezielle Ausprägung einer Szene hängt sowohl von  $\sigma_S$  als auch von der Auswahl der Assoziationen  $\in \mathbf{A}^g$  in der Folge der Ersetzungsschritte zur Generierung der terminalen Instanzen der Szene ab (Gl. 2-15). Die Ausprägungen unterscheiden sich z. B. auf Grund spezieller Objektzusammenstellungen, Objektnordnungen, Beleuchtungen, Atmosphärenbedingungen, Sensoren, Programme zur Extraktion der Primitivobjekte aus den Sensordaten sowie deren Parametrisierung.

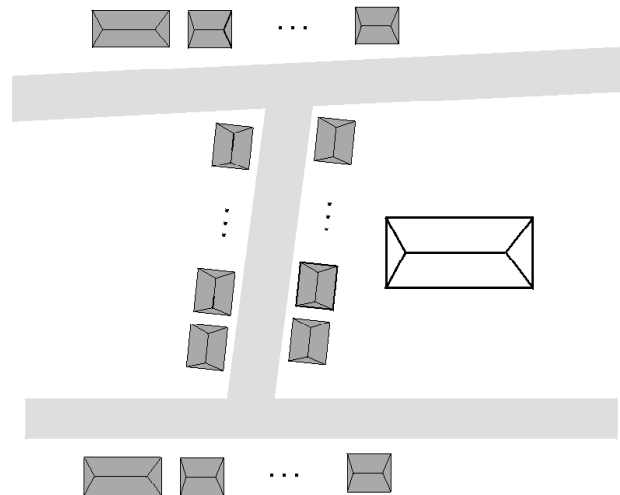
**Objekte zu analysierender Bilder** Die Objekte der Szenen sind Zwischenergebnisse der Ersetzung  $\{\sigma_S\} \xrightarrow[e]{+\mathbf{A}^g} \mathcal{T}$  (Gl. 2-15), die aus dem die Szene definierenden Startsymbol  $\sigma_S \in \Sigma_S$  die schließlich zu analysierenden Primitivobjekte der Menge  $\mathcal{T} \in S_g(\mathcal{I}_T, \mathcal{I}_N, \Sigma_S, \mathbf{A}^g)$  der speziellen Szene erzeugt. Dabei wird eine Menge generierender Assoziationen  $\mathbf{A}^g$  angenommen, die allerdings in der Regel nur zu einem kleinen Teil konkret angegeben werden kann. Oft werden nur die in der Szene enthaltenen Objekte mit ihren Teilen festgehalten, ohne Assoziationen für deren Generierung anzugeben.

Die Objekte der Szene können folgendermaßen eingeteilt werden:

1. Ziel-Objekte  $\gamma^Z \in \mathcal{O}^Z$ , die aufgrund der jeweiligen Aufgabenstellung durch eine Analyse gefunden und beschrieben werden sollen (z. B. Häuser),

2. Kontext-Objekte  $\gamma^K \in \mathcal{O}^K$ , die zusammen mit den unter (1.) genannten Objekten auftreten (im Beispiel Straßen),
3. Verbund-Objekte  $\gamma^V \in \mathcal{O}^V$ , in die Ziel- und Kontext-Objekte eingebettet sind (im Beispiel Siedlungen),
4. in den Trainingsdaten enthaltene, für die Analyse nicht modellierte, aber bekannte Objekte des Hintergrundes  $\gamma_b^H \in \mathcal{O}_b^H$ , die zu keinen Analysefehlern führen, und
5. unbekannte Objekte des Hintergrundes in Testdaten  $\gamma_u^H \in \mathcal{O}_u^H$ , die mit bekannten Objekten verwechselt werden können und dann zu Fehlern führen können.

Die unter 1. bis 4. aufgeführten Objekte treten bei der Ersetzung  $\{\sigma_S\} \xrightarrow[e]{+A^g} \mathcal{T}$  von Trainingsdaten auf (Abbildung 2-13). Solange man sich auf diese Objekte beschränkt, werden bei der Analyse (nach Voraussetzung) keine Objekte verwechselt. Unbekannte Objekte (5. Fall) können bei jedem vorstellbaren Analyseverfahren zu Fehlern führen. Die Erfahrung zeigt, dass auch der Mensch mit seinen überlegenen Erkennungsfähigkeiten, wenn in einer Szene unbekannte Hintergrund-Objekte vorliegen, bei der Analyse bekannter Ziel-Objekte Fehler machen kann.



**Abb. 2-13:** Beispiel für Objekte einer Szene  
 Hintergrundobjekte (große Halle) hell  
 Kontextobjekte (Straßen) hellgrau  
 Zielobjekte (Häuser) dunkelgrau

**Analyse der Trainingsdaten** Die Trainingsdaten enthalten (nach Voraussetzung) nur eine überschaubare Anzahl unterschiedlicher Hintergrund-Objekte, so dass (i) die Hintergrund-Objekte, die mit den Assoziationen zu keinen Analysefehlern führen, festgestellt werden können und (ii) die Assoziationen an die Hintergrund-Objekte angepasst werden können. Wenn beispielsweise in der Ent-

wicklungsphase bei der Analyse der Trainingsdaten Hintergrund-Objekte mit Ziel-, Kontext- oder Verbund-Objekten verwechselt werden, können diese Hintergrund-Objekte zusätzlich modelliert und entweder der Klasse der Kontext-Objekte zuge schlagen oder als konkurrierende Zielobjekte eingeführt werden. Um beispielsweise in Abbildung A-17a mit der Brücke nicht die Häuserformation, sondern die Brücke zuzuordnen, muss auch nach dem Zielobjekt Brücke gesucht werden. Werden beide Zielobjekte mit unterschiedlichen Bewertungen gefunden, kann anhand der Bewertung entschieden werden.

Die Untersuchungen zur Modellierung der hier betrachteten Analyse mit Erzeugungs- und Erkennungsverfahren und zur Einteilung der Szenenobjekte sind noch nicht abgeschlossen. Arbeitshypothese ist, dass in jedem zu analysierenden Datensatz (z. B. Einzelbild, komplette Bildfolge, multispektraler Datensatz) durch die Analyse genau ein Verbund-Objekt, das die Ziel-Objekte als Teile enthält, bestimmt wird. Ziel- und Verbund-Objekt können identisch sein. Die Brücke in Abbildung A-4a ist ein Beispiel für diesen Fall. Kontext-Objekte können fehlen. In Abbildung A-5b ist z. B. die Häuserkonfiguration als Verbund-Objekt ausschließlich durch die Anordnung ihrer Ziel-Objekte (Häuser) definiert.

Mit der Zielsetzung, bei der Analyse genau ein Verbund-Objekt und nicht nur (viele) isolierte Zielobjekte zu erkennen, verbindet sich die Vorstellung, dass für eine robuste Analyse möglichst alle zur Verfügung stehenden Daten und bekannten Zusammenhänge berücksichtigt werden sollten. Beispielsweise konnte die Erkennung der Häuserformation in Abbildung A-21b (oben), die hier als Verbund-Objekt interpretiert wird, fehlerfrei durchgeführt werden. Aus den Zwischenergebnissen in Abbildung A-21b (rechts) sieht man, dass im Gegensatz dazu die isolierte Analyse der Einzelhäuser, die die eigentlichen Ziel-Objekte sein sollen, zu Fehlern geführt hätte.

Die Analyse der Ziel-, Kontext- und Verbund-Objekte wird mit analysierenden Assoziationen einer Menge  $\mathbf{A}^a$  formuliert. Dabei wird  $\mathbf{A}^a$  konzeptionell aus  $\mathbf{A}^g$  abgeleitet<sup>8</sup>. Nach der Analyse sind die Ziel- und Kontext-Objekte Teil der Knoten des Ableitungsgraphen  $AG_N(\gamma_V)$  des erkannten Verbundobjektes  $\gamma_V$ . Damit ergibt sich bei der Analyse der Trainingsdaten der in den Gleichungen 2-25 bis 2-27 formulierte Zusammenhang.

In einer stark vereinfachten Sicht wird ein Startsymbol  $\sigma_S$  so lange durch eine (nicht festgelegte) Folge von Ersetzungsschritten ersetzt, bis eine Menge terminaler Instanzen  $\mathcal{T}$  entsteht. Aus dieser Menge entsteht als Analyseergebnis wieder das Startsymbol.

$$\forall \mathcal{I}_T, \mathcal{I}_N \subseteq \mathcal{I} : \forall \mathbf{A}^g \subseteq \wp(\mathcal{I}) \times \wp(\mathcal{I}) : \forall \Sigma_S \subseteq \mathcal{I}_N : \forall \sigma_S \in \Sigma_S : \forall \mathcal{T} \subseteq \mathcal{I}_T : \\ \{\sigma_S\} \xrightarrow[e]{+\mathbf{A}^g} \mathcal{T} \xrightarrow[e]{+(\mathbf{A}^g)^{-1}} \{\sigma_S\} \leftrightarrow \begin{cases} \sigma_S \text{ ist das zu } \mathcal{T} \\ \text{gehörende Startsymbol} \end{cases} \quad (2 - 25)$$

Bei der Generierung der Trainingsdaten wird angenommen, dass es als Zwischenschritt eine Menge gibt, in der nur ein Verbundobjekt  $\gamma_V$  und bekannte Hintergrund-Objekte der Menge  $\mathcal{O}_b^H$  enthalten sind. Anschließend werden die ter-

<sup>8</sup>Wenn  $\mathbf{A}^g$  bekannt ist, soll  $\mathbf{A}^a \approx (\mathbf{A}^g)^{-1}$  gelten.

minimalen Instanzen des Verbundobjektes  $\gamma_V$  generiert. Dabei entsteht zusätzlich zur Menge  $\mathcal{O}_b^H$  mit bekannten Hintergrund-Objekten eine Menge  $\mathcal{R}$  mit terminalen Instanzen. Aus den terminalen Instanzen wird dann durch die Analyse wieder das Verbundobjekt zusammengesetzt.

$$\begin{array}{l}
\forall \mathcal{I}_T, \mathcal{I}_N \subseteq \mathcal{I} : \forall \mathbf{A}^g \subseteq \wp(\mathcal{I}) \times \wp(\mathcal{I}) : \forall \Sigma_S \subseteq \mathcal{I}_N : \forall \sigma_S \in \Sigma_S : \\
\forall \mathcal{O}^Z, \mathcal{O}^V, \mathcal{O}_b^H \subseteq \mathcal{I}_N : \forall \gamma_V \in \mathcal{O}^V : \forall \mathcal{R} \subseteq \mathcal{I}_T : \\
\left. \begin{array}{l}
\{\sigma_S\} \xrightarrow[e]{+\mathbf{A}^g} (\mathcal{O}_b^H \cup \{\gamma_V\}) \\
(\mathcal{O}_b^H \cup \{\gamma_V\}) \xrightarrow[e]{+\mathbf{A}^g} (\mathcal{O}_b^H \cup \mathcal{R}) \\
(\mathcal{O}_b^H \cup \mathcal{R}) \xrightarrow[e]{+(\mathbf{A}^g)^{-1}} (\mathcal{O}_b^H \cup \{\gamma_V\}) \\
\mathcal{Z} := AG(\gamma_V) \cap \mathcal{O}^Z
\end{array} \right\} \leftrightarrow \left\{ \begin{array}{l}
\gamma_V \text{ ist das zu } \mathcal{R} \\
\text{mit dem Szenenhintergrund } \mathcal{O}_b^H \\
\text{gehörende Verbundobjekt} \\
\text{der Szene } \sigma_S
\end{array} \right.
\end{array} \quad (2-26)$$

Die Konfiguration  $\mathcal{O}_b^H \cup \mathcal{R}$  in Gleichung 2-26 enthält Hintergrund-Objekte und terminale Instanzen der zu analysierenden Objekte. Zu analysierende Datensätze enthalten aber ausschließlich terminale Instanzen, die von Extraktionsprogrammen aus den Sensordaten extrahiert werden. Mit der Menge der analysierenden Assoziationen  $\mathbf{A}^a$  und der generierenden Assoziationen  $\mathbf{A}^g$  erhält man:

$$\begin{array}{l}
\forall \mathcal{I}_T, \mathcal{I}_N \subseteq \mathcal{I} : \forall \mathbf{A}^a, \mathbf{A}^g \subseteq \wp(\mathcal{I}) \times \wp(\mathcal{I}) : \forall \Sigma_S \subseteq \mathcal{I}_N : \forall \sigma_S \in \Sigma_S : \\
\forall \mathcal{O}^Z, \mathcal{O}^V, \mathcal{O}_b^H \subseteq \mathcal{I}_N : \forall \gamma_V \in \mathcal{O}^V : \forall \mathcal{Q} \subseteq \mathcal{I} : \forall \mathcal{T} \subseteq \mathcal{I}_T : \\
\left. \begin{array}{l}
\mathbf{A}^g \approx (\mathbf{A}^a)^{-1} \\
\{\sigma_S\} \xrightarrow[e]{+\mathbf{A}^g} (\mathcal{O}_b^H \cup \{\gamma_V\}) \\
(\mathcal{O}_b^H \cup \{\gamma_V\}) \xrightarrow[e]{+\mathbf{A}^g} \mathcal{T} \\
\mathcal{T} \xrightarrow[e]{+\mathbf{A}^a} \mathcal{Q} \\
\mathcal{Q} \text{ kann nicht weiter ersetzt werden} \\
\mathcal{Q} \text{ ist das beste Analyseergebnis} \\
\mathcal{Z} := \bigcup_{\epsilon \in \mathcal{Q} \cap \mathcal{O}^V} AG(\epsilon) \cap \mathcal{O}^Z
\end{array} \right\} \leftrightarrow \left\{ \begin{array}{l}
\mathcal{Q} \text{ ist das zu } \mathcal{T} \\
\text{gehörende Analyseergebnis} \\
\text{der Szene } \sigma_S \\
\text{mit dem Hintergrund } \mathcal{O}_b^H \\
\text{und dem Verbundobjekt } \gamma_V \\
\text{sowie der Menge der Zielobjekte } \mathcal{Z}
\end{array} \right.
\end{array} \quad (2-27)$$

Der Grad der Übereinstimmung zwischen  $\mathcal{Q} \cap \mathcal{O}^V$  und  $\gamma_V$  gibt die jeweils erreichte Analysegröße an. Dabei sollte in  $\mathcal{Q} \cap \mathcal{O}^V$  nur ein Verbundobjekt  $\epsilon$  enthalten sein. Ersetzend müssen viele unterschiedliche Folgen von Ersetzungsschritten  $\mathcal{T} \xrightarrow[e]{+\mathbf{A}^a} \mathcal{Q}$  probiert werden. Mit der Forderung "Q ist das beste Ergebnis" wird für die Analyse eine Ersetzungsfolge ausgewählt (vgl. auch Erläuterung zu Gleichung 2-29).

**Problemfälle bei der Analyse mit Konfigurationen-Erkennungsverfahren** Erkennungsverfahren für Konfigurationen dienen zur strukturellen Musteranalyse, die allerdings in vielen Fällen allein aufgrund der zu analysierenden Szene nicht erfolgreich durchgeführt werden kann. Aus Sicht der strukturellen Musteranalyse können u. a. folgende Probleme auftreten:



1. Fehlende Terminale: Wenn in einer Szene zu einem Objekt keine terminalen Instanzen gehören, kann das Objekt nicht erkannt werden. Der Fall tritt z. B. bei völlig verdeckten Objekten auf.
2. Mehrdeutige Modelle durch eine vexierende Erzeugung: Die Erzeugung heißt *vexierend*, wenn die terminalen Instanzen keine eindeutigen Rückschlüsse auf die Objekte erlauben. In dem Vexierbild der Abbildung 1-1 werden z. B. zwei Gesichter mit einer Vase verwechselt, weil zu zwei Gesichtern im richtigen Abstand die gleichen Primitivobjekte gehören wie zu einer Vase.
3. Verwechelte Startsymbole: Wenn es mehrere Objekte gibt, zu denen die gleichen Primitivobjekte gehören, dann werden bei der Analyse Objekte der Szene verwechselt. Dieser Fall tritt regelmäßig bei teilweise verdeckten oder schlecht aufgelösten Objekten auf.
4. Verschmolzene Instanzen: Instanzen können Teil mehrerer Zwischenergebnisse sein. Derartige Muster werden konzeptionell durch mehrfach assoziierende Assoziationen generiert (Satz B.86). Sie können ersetzend nicht erkannt werden (Satz B.88), erzeugen zuordnend bei der Analyse Ableitungsgraphen (Lemma B.89) und entstehen z. B. bei aneinanderstoßenden Objekten mit verschmolzenen Objekträndern (vgl. auch Abschnitt 3.1).

Die Auflistung zeigt die Ausprägung allgemeiner Analyseprobleme, die bei einer Vielzahl unterschiedlicher Analyseansätze auftreten, aus der Perspektive des hier dargestellten, auf Konfigurationen aufbauenden strukturellen Analyseansatzes.

**Erweiterung der Analyse** Für die Bewertung, Anpassung und Weiterentwicklung von Analyseverfahren werden im ersten Schritt Mengen terminaler Instanzen aus geeigneten Trainings-Datensätzen extrahiert und strukturell analysiert. Anschließend werden die erzielten realen Ergebnisse mit interaktiv vorgegebenen idealen Analysedaten (ground truth) verglichen.

Bei Analysefehlern muss für die Anpassung und Weiterentwicklung im zweiten Schritt versucht werden, zusätzliche Kontext- und Verbund-Objekte, konkurrierende Zielobjekte sowie analysierende Assoziationen zu definieren. Falls auch zusätzliche terminale Instanzen benötigt werden, so können diese z. B. aus Daten zusätzlicher Sensoren oder aus Daten, die von anderen Standorten aus bzw. mit anderen Sensoreinstellungen aufgenommen werden, extrahiert werden. Außerdem kann die Parametrisierung der Extraktionsprogramme zur Generierung der terminalen Instanzen aus den Sensordaten geändert werden (Abschnitt A.2). In vielen Fällen müssen zusätzliche Extraktionsprogramme bereitgestellt werden.

**Analyse unbekannter Testdaten** Nach der Optimierung der Analyseverfahren mit den Trainingsdaten wird das entwickelte Verfahren in der Regel bewertet. Dazu wird in Gleichung 2-27 die Menge der bekannten Hintergrund-Objekte  $\mathcal{O}_b^H$  der Trainingsdaten durch die Menge der unbekanntenen Hintergrund-Objekte  $\mathcal{O}_u^H$  der

Testdaten ersetzt:

$$\begin{array}{l}
\forall \mathcal{I}_T, \mathcal{I}_N \subseteq \mathcal{I} : \forall \mathbf{A}^a, \mathbf{A}^g \subseteq \wp(\mathcal{I}) \times \wp(\mathcal{I}) : \\
\forall \mathcal{O}^Z, \mathcal{O}^V, \mathcal{O}_u^H \subseteq \mathcal{I}_N : \forall \gamma_V \in \mathcal{O}^V : \forall \mathcal{Q} \subseteq \mathcal{I} : \forall \mathcal{T} \subseteq \mathcal{I}_T : \\
\left. \begin{array}{l}
\mathbf{A}^g \approx (\mathbf{A}^a)^{-1} \\
(\mathcal{O}_u^H \cup \{\gamma_V\}) \xrightarrow[e]{+\mathbf{A}^g} \mathcal{T} \\
\mathcal{T} \xrightarrow[e]{+\mathbf{A}^a} \mathcal{Q} \\
\mathcal{Q} \text{ kann nicht weiter ersetzt werden} \\
\mathcal{Q} \text{ ist das beste Analyseergebnis} \\
\mathcal{Z} := \bigcup_{\epsilon \in \mathcal{Q} \cap \mathcal{O}^V} AG(\epsilon) \cap \mathcal{O}^Z
\end{array} \right\} \leftrightarrow \left\{ \begin{array}{l}
\mathcal{Q} \text{ ist das zu } \mathcal{T} \\
\text{gehörende Analyseergebnis} \\
\text{der Szene } \mathcal{O}_u^H \cup \{\gamma_V\} \\
\text{mit dem Hintergrund } \mathcal{O}_u^H \\
\text{und dem Verbundobjekt } \gamma_V \\
\text{sowie der Menge der Zielobjekte } \mathcal{Z}
\end{array} \right.
\end{array} \quad (2-28)$$

$\mathcal{Q} \cap \mathcal{O}^V$  ist die Menge der Verbundobjekte,  $\mathcal{Z}$  die Menge der zu erkennenden Zielobjekte. Das Ergebnis wird bewertet, indem  $\mathcal{Q} \cap \mathcal{O}^V$  mit der Beschreibung des Datensatzes - d.h.  $\gamma_V$  und  $\mathcal{O}_u^H$  - verglichen wird. Wenn in  $\mathcal{Q} \cap \mathcal{O}^V$  mehr als ein Verbundobjekt  $\epsilon$  enthalten sein sollte, müssen anwendungsspezifisch die relevanten Analyseergebnisse extrahiert werden.

Dieser Abschnitt erläutert grundsätzliche Zusammenhänge einer Analyse, die Muster mit ungeordneten Mengen beschreibt. Da ersetzend nicht alle generierten Konfigurationen erkannt werden können (Satz B.84) und um den Aufwand zu reduzieren, wird in der Praxis nicht ersetzend nach Gleichung 2-28, sondern zuordnend analysiert.

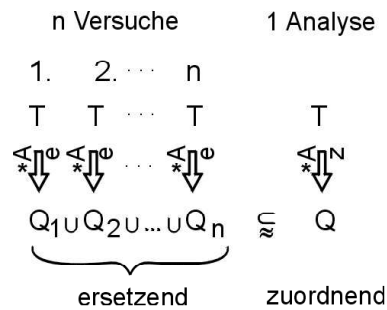
### 2.4.3 Zuordnende Erkennung erzeugter Konfigurationen

Bei der zuordnenden Erkennung werden die Zwischenergebnisse abgespeichert. Teilweise zuordnend nach Gleichung 2-19 gibt es zu jeder generierten Konfiguration  $\mathcal{R} \in S_g(\mathcal{I}_T, \mathcal{I}_N, \Sigma, \mathbf{A}^g)$  mit ihrer Folge von Ersetzungsschritten eine Folge zuordnender Analyseschritte, so dass das generierte Szenario  $S_g(\mathcal{I}_T, \mathcal{I}_N, \Sigma, \mathbf{A}^g)$  im teilweise zuordnend erkannten Szenario  $S_a^{tz}(\mathcal{I}_T, \mathcal{I}_N, \Sigma, (\mathbf{A}^g)^{-1})$  enthalten ist (Satz B.81).

Aus Gleichung 2-28, die die ersetzende Erkennung auszuwertender Bilder beschreibt, wird zuordnend analysierend Gleichung 2-29. Dabei wird aus der Ersetzung  $\mathcal{T} \xrightarrow[e]{+\mathbf{A}^a} \mathcal{Q}$  mit den Bedingungen  $\mathcal{Q}$  kann nicht weiter ersetzt werden und  $\mathcal{Q}$  ist das beste Analyseergebnis die abschließende Zuordnung  $\mathcal{T} \xrightarrow[z]{\dagger \mathbf{A}^a} \mathcal{Q}$ , bei der die Mengen  $\mathcal{Q}$  und  $\mathcal{Z}$  wesentlich mehr Elemente als in Gleichung 2-28 enthalten können. Zuordnend wird mit  $\mathcal{T} \xrightarrow[z]{\dagger \mathbf{A}^a} \mathcal{Q}$  nur ein  $\mathcal{Q}$  bestimmt, in dem alle aus  $\mathcal{T}$  ableitbaren End- und Zwischenergebnisse enthalten sind. Ersetzend müssen dagegen für die Suche des besten  $\mathcal{Q}$  nacheinander viele unterschiedliche Folgen von Ersetzungsschritten  $\mathcal{T} \xrightarrow[e]{+\mathbf{A}^a} \mathcal{Q}$  durchprobiert werden, zu denen jeweils eine Menge  $\mathcal{Q}$  gehört

(Abbildung 2-14).

$$\begin{array}{l}
 \forall \mathcal{I}_T, \mathcal{I}_N \subseteq \mathcal{I} : \forall \mathbf{A}^a, \mathbf{A}^g \subseteq \wp(\mathcal{I}) \times \wp(\mathcal{I}) : \\
 \forall \mathcal{O}^Z, \mathcal{O}^V, \mathcal{O}_u^H \subseteq \mathcal{I}_N : \forall \gamma_V \in \mathcal{O}^V : \forall Q \subseteq \mathcal{I} : \forall T \subseteq \mathcal{I}_T : \\
 \left. \begin{array}{l}
 \mathbf{A}^g \approx (\mathbf{A}^a)^{-1} \\
 (\mathcal{O}_u^H \cup \{\gamma_V\}) \xrightarrow[e]{+\mathbf{A}^g} T \\
 T \xrightarrow[z]{+\mathbf{A}^a} Q \\
 \mathcal{Z} := \bigcup_{\epsilon \in \mathcal{Q} \cap \mathcal{O}^V} AG(\epsilon) \cap \mathcal{O}^Z
 \end{array} \right\} \leftrightarrow \left\{ \begin{array}{l}
 Q \text{ ist das zu } T \\
 \text{gehörende Analyseergebnis} \\
 \text{der Szene } \mathcal{O}_u^H \cup \{\gamma_V\} \\
 \text{mit dem Hintergrund } \mathcal{O}_u^H \\
 \text{und dem Verbundobjekt } \gamma_V \\
 \text{sowie der Menge der Zielobjekte } \mathcal{Z}
 \end{array} \right. \quad (2 - 29)
 \end{array}$$



**Abb. 2-14:** Ersetzende und zuordnende Erkennung

In  $Q \cap \mathcal{O}^V$  werden in der Regel mehrere Verbundobjekte  $\epsilon$  enthalten sein, aus denen anwendungsspezifisch die besten Analyseergebnisse extrahiert werden müssen. Bewertet wird, indem die Elemente von  $Q \cap \mathcal{O}^V$  mit  $\gamma_V$  verglichen werden.

### 3 Diskussion

Anschließend werden die im 2. Kapitel definierten Systeme, Automaten und Verfahren diskutiert. Im Vordergrund steht dabei der Analyseaufwand.

Ersetzende Systeme erzeugen bei der für die Analyse durchgeführten Suche Zwischenergebnisse mehrfach, so dass durch die Mehrfachgenerierung in der Regel unnötiger Aufwand entsteht, der mit zuordnenden Systemen, die alle Zwischenergebnisse speichern und nur ein Mal generieren<sup>1</sup>, vermieden wird (Abschnitt 3.1). Beispiel für ersetzende Systeme sind das ERNEST-System /Sag85/ und daraus abgeleitete. Zum Vergleich mit derartigen Systemen (siehe auch Abschnitt 4.2; Seite 92) werden im Abschnitt 3.3 spezielle Algorithmen aufgeführt und ihr Aufwand diskutiert.

Ein weiterer Punkt ist die Frage der kontextfreien bzw. kontextsensitiven Erkennung. Es zeigt sich, dass diese Eigenschaften bei zuordnender Erkennung sekundär sind und statt dessen die Eigenschaft mehrfach assoziierend betrachtet werden muss (Abschnitt 3.2).

Schließlich werden Ansätze zur Begrenzung und Minimierung des Analyseaufwandes erläutert (Abschnitte 3.5 bis 3.7).

Zentrale Aussage dieses Kapitels ist, dass klassische wissensbasierte Systeme, die ersetzend arbeiten und dann mehrfach generieren, bei den hier betrachteten Bildanalyseaufgaben mit Teil-von-Relationen<sup>2</sup> einen höheren Aufwand erfordern als die mit dieser Arbeit eingeführten Zuordnungsautomaten, die Ergebnisse nur einfach speichern.

#### 3.1 Ersetzende und zuordnende Erkennung

Die in diesem Kapitel diskutierten Verfahren setzen Assoziationen ersetzend (Gl. 2-7) oder zuordnend (Gl. 2-8) ein. Die zuordnend eingesetzten Assoziationen  $(\mathcal{X}, \mathcal{Y}) \in \mathbf{A}$  werden vom Entwickler in der Regel ersetzend formuliert, mit der Intention, dass alle Instanzen von  $\mathcal{X}$  nach dem Analyseschritt mit  $(\mathcal{X}, \mathcal{Y})$  entsprechend Gleichung 2-7 nicht mehr betrachtet werden müssen und durch neue Instanzen der Menge  $\mathcal{Y}$  beschrieben werden.

Bei Zuordnungen (Gl. 2-8) werden keine Zwischenergebnisse verändert, ersetzt oder gelöscht. Es werden nur neue Zwischenergebnisse hinzugefügt. Zuordnend eingesetzte Assoziationen  $(\mathcal{X}, \mathcal{Y}) \in \mathbf{A}$  wirken sich deshalb wie kontextsensitive Assoziationen  $(\mathcal{X}, \mathcal{X} \cup \mathcal{Y})$  aus (Gl. B-69). Daher entstehen bei der zuordnenden Erkennung von Mustern (Gl. 2-19, Gl. 2-21) zusätzlich zu den ersetzend erkannten Szenarien  $S_a^{ve}$  bzw.  $S_a^{te}$  (Gl. 2-17, Gl. 2-18) noch die Mengen  $S_a^{vz} \setminus S_a^{ve}$  bzw.  $S_a^{tz} \setminus S_a^{te}$ .

---

<sup>1</sup>Die Ergebnisse können - solange sie nur ein Mal gespeichert werden - auch mehrfach generiert werden (Zuordnungsautomaten generieren z. B. mehrfach, siehe Abschnitt 3.4.1); entscheidend ist, dass jedes Zwischenergebnis nur ein Mal Überprüfungen anstößt.

<sup>2</sup>Die Analyse mit Teil-von-Relationen soll dabei Ableitungsbäume und keine Ableitungsketten erzeugen.

Dabei können Erkennungen erfolgen, die nicht der Intention des Entwicklers entsprechen.

Um die im Sinne der Analyseaufgabe "richtigen" Ergebnisse zu erhalten, kann bei Instanzen, die als Ableitungen formuliert sind, der Ableitungsgraph (Gl. B-59) mit seinen Zwischenergebnissen untersucht werden. Durch diese Nachbehandlung können zuordnend entstandene Startsymbole, die nicht der Intention des Entwicklers entsprechen, nachträglich verworfen werden. Darüber hinaus wird von der Arbeitshypothese ausgegangen, dass auf die Analyse der Ableitungsgraphen verzichtet werden kann, wenn Instanzen mit Ausdehnungen und anlagernde Assoziationen verwendet werden (vgl. Diskussion der Gleichung 2-24 auf Seite 37).

Vorteil der zuordnenden Erkennung ist, dass auch bei verschmolzenen Terminalen, die Teile mehrerer anderer Instanzen sein können (vgl. Seite 47), oder bei Konfigurationen, die die Vereinigung generierter Konfigurationen sind, Startsymbole erkannt werden (Satz B.86 und Satz B.88). Ersetzend kann in diesen Fällen nicht erkannt werden. Dieses Problem bei der ersetzenden Erkennung entsteht formal durch die Mengenrepräsentation der Konfigurationen und wird mit *mehrfach assoziierenden Assoziationen* (Gl. B-72) erfasst. Die Untersuchungen hierzu sind noch nicht abgeschlossen und werden im Ausblick (Abschnitt 5.2) aufgegriffen.

Außerdem veranlassen zuordnend entstandene Zwischenergebnisse, die nur einfach abgespeichert sind, keine mehrfachen Überprüfungen. Deshalb ist die zuordnende Erkennung in der Regel deutlich effizienter als die ersetzende. Im Abschnitt 3.3 wird der Aufwand für beide Varianten diskutiert.

### 3.2 Kontextfreie und kontextsensitive Erkennung

Ersetzende Verfahren generieren mit kontextfreien Assoziationen (Gl. B-69) Ergebnisse, die man so auch bei der Definition der Assoziationen intuitiv erwartet: mit jedem Ersetzungsschritt entstehen bei der Generierung und Analyse voneinander unabhängige Teilbäume, deren Generierung lokal an jedem Knoten anhand der jeweils verwendeten Assoziation erklärt werden kann. In Abbildung 3-1 entstehen beispielsweise aus  $\delta$  mit der Assoziation  $(\{\delta\}, \{\gamma_{1,1}, \gamma_{1,2}\})$   $\gamma_{1,1}$  und  $\gamma_{1,2}$ . Entsprechend entstehen aus  $\gamma_{1,1}$  bzw.  $\gamma_{1,2}$  mit  $(\{\gamma_{1,1}\}, \{\alpha_{2,1}, \alpha_{2,2}\})$   $\alpha_{2,1}$  und  $\alpha_{2,2}$  bzw. mit  $(\{\gamma_{1,2}\}, \{\alpha_{2,3}, \alpha_{2,4}\})$   $\alpha_{2,3}$  und  $\alpha_{2,4}$ .

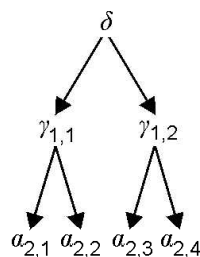


Abb. 3-1: Teilbäume bei kontextfreien Assoziationen

Bei kontextsensitiven Assoziationen gibt es ersetzend eine Abhängigkeit der

Anwendbarkeit einer Assoziation von der Reihenfolge, in der Assoziationen eingesetzt werden. Arbeitshypothese ist, dass kontextsensitive Assoziationen - genauso wie Produktionen kontextsensitiver Chomsky-Grammatiken - ersetzend primär den Analyseablauf beeinflussen und die Überprüfung des Kontextes bei den einzelnen Analyseschritten in den Hintergrund tritt (vgl. hierzu z. B. /Sal78/).

Ersetzend sind sowohl Erzeugungs- als auch Erkennungsverfahren definiert. Die der zuordnenden Erkennung unmittelbar entsprechende zuordnende Erzeugung ist dagegen nicht definiert, so dass hier ein Defizit besteht.

Zuordnende Erkennungsverfahren mit kontextfreien und kontextsensitiven Assoziationen werden folgendermaßen gerechtfertigt:

1. Alle Ergebnisse, die ersetzend generiert werden, werden zuordnend erkannt (Satz B.81). Dabei sind die Analyseergebnisse bei kontextfreien Assoziationen mit den jeweils voneinander unabhängigen Teilbäumen intuitiv unmittelbar nachzuvollziehen (Abbildung 3-1).
2. Jede kontextfreie Menge von Assoziationen  $\mathbf{B}^F$  erzeugt zuordnend die gleichen Ergebnisse<sup>3</sup> wie die Menge kontextsensitiver Assoziationen  $\mathbf{B}^K$  mit

$$\mathbf{B}^K := \{(\mathcal{X}, \mathcal{Y} \cup \mathcal{X}) \mid (\mathcal{X}, \mathcal{Y}) \in \mathbf{B}^F\}. \quad (3-1)$$

3. Jede (lokal) kontextsensitive Menge von Assoziationen  $\mathbf{A}^K$  erzeugt zuordnend die gleichen Ergebnisse wie die Menge (lokal) kontextfreier Assoziationen  $\mathbf{A}^F$  mit

$$\mathbf{A}^F := \{(\mathcal{X}, \mathcal{Y} \setminus \mathcal{X}) \mid (\mathcal{X}, \mathcal{Y}) \in \mathbf{A}^K\}. \quad (3-2)$$

Arbeitshypothese ist, dass bei Zuordnungen die für Zeichenketten-Grammatiken zentralen Eigenschaften *kontextfrei* und *kontextsensitiv* sekundär sind und statt dessen die für Konfigurationen definierte Eigenschaft *mehrfach assoziierend* diskutiert werden muss. Die Untersuchungen hierzu sind noch nicht abgeschlossen und werden im Ausblick (Abschnitt 5.2) aufgegriffen.

### 3.3 Aufwand für die Musteranalyse mit Produktionsnetzen

In den folgenden Abschnitten werden Grenzfälle für und Einflüsse auf den Analyseaufwand diskutiert. Dazu werden zusätzliche Algorithmen, die nicht auf Zuordnungsautomaten aufbauen und Mengen terminaler Instanzen vollständig durchsuchen, definiert, mit denen auch einfache Rechnerprogramme für erste Untersuchungen zur strukturellen Musteranalyse mit Produktionsnetzen implementiert werden könnten.

Der Aufwand für strukturelle Analysen hängt von den zu analysierenden Strukturen und dem Analyseverfahren ab. Als Grenzfälle der zu analysierenden Strukturen werden Aneinanderreihungen, für die Ableitungsketten entstehen, und Strukturen mit Teil-von-Hierarchien, für die Ableitungsbäume entstehen, untersucht.

---

<sup>3</sup>Es ist zuordnend kein Unterschied, ob die überprüfte und bereits vorliegende Konfiguration  $\mathcal{X}$  durch die Zuordnung in die resultierende neue Konfiguration (Menge) mit eingetragen wird oder nicht.

Motivation für die Untersuchungen dieses Abschnitts war, Grundlagen für Aufwandsabschätzungen bei Analysen mit Zuordnungsautomaten zu liefern; auch im Vergleich zu Verfahren, die Analysezustände speichern und bewerten sowie mit dem A\*-Algorithmus optimieren /Qui97, Sag85/ (vgl. Abschnitte 4.2). Außerdem sollte der Bezug zu allgemeinen Suchverfahren /Rus95, Ben96, Poo98/ hergestellt werden.

### Analyseaufwand und Anzahl der Zwischenergebnisse

Bei der strukturellen Analyse mit Produktionsnetzen hängt der Analyseaufwand in erster Linie von der Anzahl der zu analysierenden terminalen Instanzen und der dazu generierten Zwischenergebnisse ab. Aufwand wird dabei mit der Anzahl der durchgeführten Überprüfungen gleichgesetzt. Der bei der Ausführung auf einem Rechner benötigte Speicherplatz und die Rechenzeit zur Ausführung einer einzelnen Überprüfung werden ignoriert.

Es wird vorausgesetzt, dass im Rahmen der Verfahrensentwicklung Analysemodelle mit ausreichend *selten auftretenden* Zwischenergebnissen gefunden werden können. Ein Ziel der Verfahrensentwicklung ist, mit geeignet gewählten Zwischenergebnissen (d.h. Konzepten in den Produktionsnetzen) die Anzahl der während einer Analyse generierten Zwischenergebnisse (d.h. Instanzen in den Ableitungsgraphen) zu minimieren. Die Produktionsnetze müssen also an die Statistik<sup>4</sup> der Daten angepasst werden. Aus der Statistik der Szene bzw. der Eingangsdaten kann allerdings nur eine untere Grenze für die Anzahl der generierten Zwischenergebnisse abgeleitet werden (vgl. Abschnitt 3.4).

Aus der Anzahl der während der Analyse tatsächlich generierten Zwischenergebnisse folgt unmittelbar die Anzahl der Überprüfungen. Die Anzahl der Zwischenergebnisse kann entweder vor einer Analyse geschätzt werden, oder es können Soll- bzw. Grenzwerte für die Anzahl vorgegeben werden (vgl. Abschnitt 3.5).

### Aufgabenstellung und Randbedingungen

Bei den Algorithmen wird als Aufgabe zu Grunde gelegt, dass in der Menge der zu analysierenden terminalen Instanzen mehrere Ergebnismuster enthalten sein können und die Analyse alle feststellen soll.

Für die Aufwandsabschätzungen wird angenommen, dass die Anzahl der zu analysierenden Instanzen der Menge  $\mathcal{T}$  wesentlich größer ist als die Anzahl der zu einem einzelnen Muster bzw. Startsymbol zusammenfassenden terminalen Instanzen der Menge  $\mathcal{R}$  (Abbildungen 2-12 a mit  $\mathcal{T}$  und 2-12 f mit  $\mathcal{R}$  sowie 2-10 oben). Außerdem wird angenommen, dass die Analyse nicht durch die Anzahl der von der Analyse berücksichtigten Instanzen in  $\mathcal{I}_a(\mathbf{A})$  (Gl. B-2) eingeschränkt ist (vgl. Seite 69).

---

<sup>4</sup>Statistik meint hier die Anzahl der in den Sensordaten abgebildeten Objekte sowie Teilobjekte mit ihren Instanzen.

Für die die Analysemodelle definierenden Assoziationen wird angenommen, dass sie als Analyseergebnisse Ableitungsketten<sup>5</sup> oder regulär binäre balancierte Ableitungsbäume (rbb-Ableitungsbäume)<sup>6</sup> aufbauen /Dör88/ und in der Form  $(\{\alpha, \beta\}, \{\gamma\})$  vorliegen, so dass bei der Analyse immer zwei Instanzen  $\alpha$  und  $\beta$  zu einer neuen Instanz  $\gamma$  zusammengefasst werden. Außerdem wird vorausgesetzt, dass es zu jeder Menge mit zwei Instanzen  $\{\alpha, \beta\}$  maximal eine Assoziation gibt. Assoziation  $(\mathcal{X}, \mathcal{Y})$  mit  $|\mathcal{Y}| > 1$  oder lokal kontextsensitive Assoziationen mit  $\mathcal{X} \cap \mathcal{Y} \neq \emptyset$  (Gl. B-69) werden nicht betrachtet.

In den Algorithmen (z.B. in Algorithmus 3-4) steht die Abfrage  $(\{\alpha, \beta\}, \{\gamma\}) \in \mathbf{A}$  für die Überprüfung des die Menge  $\mathbf{A}$  definierenden Prädikates  $P_{\mathbf{A}}$ , mit dem  $\mathbf{A}$  üblicherweise anwendungsspezifisch definiert wird:  $\mathbf{A} := \{(\{\alpha, \beta\}, \{\gamma\}) \mid P_{\mathbf{A}}(\{\alpha, \beta\}, \{\gamma\})\}$  (vgl. die Definition von  $\mathbf{A}_K(\mathcal{I}, \mathbf{P}, i)$  auf Seite 24 ff.). Ein Beispiel für  $P_{\mathbf{A}}$  wäre die Überprüfung, ob  $\alpha$  und  $\beta$  zueinander parallele Geradenstücke sind und durch den Streifen  $\gamma$  beschrieben werden können (vgl. Abbildung 1-4).

Ersetzende Verfahren werden anschließend anhand von Graph-Suchalgorithmen /Poo98, Ben96, Rus95/ und zuordnende Verfahren anhand speichernder Suchalgorithmen diskutiert.

### 3.3.1 Algorithmen ohne Bewertungen

Zur Realisierung von Erkennungsverfahren und zur Abschätzung des Aufwandes für die strukturelle Musteranalyse mit Produktionsnetzen werden mehrere Algorithmen, die mit geschachtelten Schleifen Mengen von Instanzen vollständig durchsuchen, definiert.

Es werden Algorithmen zur Analyse mit Ableitungsketten und zur Analyse mit rbb-Ableitungsbäumen aufgeführt. Die beiden Algorithmen gibt es jeweils in einer Variante, die ohne Speicherung arbeitet, und in einer Variante, die Zwischenergebnisse abspeichert.

In der Regel werden vor der Suche alle terminalen Instanzen bereitgestellt. Zusätzlich wird für rbb-Bäume ein Algorithmus diskutiert, der nach einer Vorauswahl, die auf bereits vorliegenden Zwischenergebnissen aufbaut, in den Suchschleifen reduzierte Mengen durchsucht. Diese Mengen werden *Suchbereiche* genannt. Überprüfungsaufträge  $(\gamma, (\mathcal{A}, \mathcal{B}))$  (Gl. B-83) definieren mit der Produktion  $(\mathcal{A}, \mathcal{B})$  Suchbereiche passend zu der weiter zu analysierenden Instanz  $\gamma$ . Das BPI-System (siehe Abschnitt 1.4) stellt diese Suchbereiche mit einem Assoziativspeicher zur Verfügung. In Systemen mit semantischen Netzen (siehe Abschnitt 4.2.5) werden die Suchbereiche mit *modifizierten Konzepten* /Sag97/ definiert. Dabei wird angenommen, dass der Aufwand zur Bereitstellung der Suchbereiche vernachlässigt werden kann. Ein Ansatz hierzu ist, invertierte Dateien einzurichten /Schl83/.

<sup>5</sup>Ableitungsketten und -bäume sind Sonderfälle von Ableitungsgraphen (Gl. B-59); sie sind keine Zeichenketten.

<sup>6</sup>Wenn ein Knoten mit anderen verzweigt ist, dann sollen bei dem Knoten genau zwei Söhne vorliegen. Außerdem haben alle Blätter gleiches Niveau /Dör88/.



### Analyse mit Ableitungsketten

Bei der strukturellen Analyse einer Menge terminaler Instanzen  $\mathcal{T}$  mit Ableitungsketten werden die  $|\mathcal{R}|$  Elemente einer zu findenden Konfiguration  $\mathcal{R} = \{\rho_1, \rho_2, \dots, \rho_{|\mathcal{R}|}\}$  Schritt für Schritt überprüft. Hierzu wird konzeptionell die Menge der Assoziationen

$$\mathbf{A}_k := \{(\{\rho_1, \rho_2\}, \{\mu_1\}), (\{\rho_3, \mu_1\}, \{\mu_2\}), \dots, (\{\rho_{i+1}, \mu_{i-1}\}, \{\mu_i\}), \dots, (\{\rho_{|\mathcal{R}|-1}, \mu_{|\mathcal{R}|-3}\}, \{\mu_{|\mathcal{R}|-2}\}), (\{\rho_{|\mathcal{R}|}, \mu_{|\mathcal{R}|-2}\}, \{\sigma\})\} \quad (3-3)$$

eingesetzt, mit der Zwischenergebnisse  $\mu_i$  in aufsteigender Reihenfolge überprüft und gegebenenfalls generiert werden. Dabei liegt die Menge der Assoziationen nicht explizit vor, sondern wird implizit durch die Produktionen bzw. die zur Analyse eingesetzten Rechnerprogramme definiert.

Bei der Analyse entsteht als Ableitungsgraph eine Kette von Ableitungen.

Mit einer erweiterten Menge von Assoziationen  $\mathbf{A}$  (bzw. mit geeigneten Rechnerprogrammen) kann darüber hinaus nicht nur eine feste Konfiguration  $\mathcal{R}$ , sondern eine Familie von Konfigurationen  $\left\{ \mathcal{R} \left| \begin{array}{l} \exists \sigma \in \Sigma : \\ \mathcal{R} \xrightarrow[e]{*\mathbf{A}} \{\sigma\} \end{array} \right. \right\}$  gesucht werden.

**Suche ohne Speicherung der Zwischenergebnisse** Die Elemente einer in der Menge  $\mathcal{T}$  zu findenden Konfiguration  $\mathcal{R}$  können mit  $|\mathcal{R}|$  geschachtelten Schleifen gesucht werden. Diese Analyse ohne Speicherung der Zwischenergebnisse kann als (teilweise) ersetzende Analyse beschrieben werden (Gl. 2-7, 2-18). Die gesuchte Teilmenge aller terminaler Instanzen der Menge  $\mathcal{T}$  ist die Menge  $\mathcal{R}$  der ersetzten terminalen Instanzen  $\gamma_i$ , die durch die Laufindices  $l_i$  der geschachtelten Suchschleifen definiert sind. Die aktuell jeweils vorliegende Konfiguration  $\mathcal{Q}_i$  ergibt sich aus der in Algorithmus 3-4 erreichten Zeile. Dabei erhält man die Folge  $\mathcal{Q}_0 = \mathcal{T}$ ,  $\mathcal{Q}_1 = (\mathcal{Q}_0 \setminus \{\rho_1, \rho_2\}) \cup \{\mu_1\}$ ,  $\mathcal{Q}_{i+1} = (\mathcal{Q}_i \setminus \{\rho_{i+1}, \mu_{i-1}\}) \cup \{\mu_i\}$ ,  $i = 2, 3, \dots, |\mathcal{R}| - 1$ ,  $\mathcal{Q}_{|\mathcal{R}|} = \{\sigma\}$ .

$$\begin{array}{l} \text{für } l_1 := 1, |\mathcal{T}| - 1 \\ \quad \text{für } l_2 := l_1 + 1, |\mathcal{T}| \\ \quad \quad \text{falls } (\{\gamma_{l_1}, \gamma_{l_2}\}, \{\mu_1\}) \in \mathbf{A} \\ \quad \quad \quad \mu_1 \text{ liegt vor} \\ \quad \quad \quad (\text{weil } \gamma_{l_1} \text{ und } \gamma_{l_2} \text{ vorliegen und } (\{\gamma_{l_1}, \gamma_{l_2}\}, \{\mu_1\}) \in \mathbf{A}) \\ \quad \quad \text{für } l_3 := 1, |\mathcal{T}| \text{ ohne } l_1, l_2 \\ \quad \quad \quad \text{falls } (\{\gamma_{l_3}, \mu_1\}, \{\mu_2\}) \in \mathbf{A} \\ \quad \quad \quad \quad \mu_2 \text{ liegt vor} \\ \quad \quad \quad \quad \text{für } l_4 := 1, |\mathcal{T}| \text{ ohne } l_1, l_2, l_3 \\ \quad \quad \quad \quad \quad \text{falls } (\{\gamma_{l_4}, \mu_2\}, \{\mu_3\}) \in \mathbf{A} \\ \quad \quad \quad \quad \quad \quad \mu_3 \text{ liegt vor} \\ \quad \quad \quad \quad \quad \quad \dots \\ \quad \quad \quad \quad \quad \quad \text{für } l_{|\mathcal{R}|} := 1, |\mathcal{T}| \text{ ohne } l_1, l_2, \dots, l_{|\mathcal{R}|-1} \\ \quad \quad \quad \quad \quad \quad \quad \text{falls } (\{\gamma_{l_{|\mathcal{R}|}}, \mu_{|\mathcal{R}|-2}\}, \{\sigma\}) \in \mathbf{A} \\ \quad \quad \quad \quad \quad \quad \quad \quad \text{ein Analyseergebnis } \sigma \text{ liegt vor} \end{array} \quad (3-4)$$

Ausgehend von der Anzahl  $t_0 = |\mathcal{T}|$  der terminalen Instanzen und  $t_i$  der Zwischenergebnisse  $\mu_i$  sind  $n_{tkp}$  bzw. für  $|\mathcal{T}| \gg |\mathcal{R}|$  mit guter Näherung  $n_{tk}$  Überprüfungen

notwendig:

$$n_{tkp} := \frac{t_0}{2}(t_0 - 1) + t_1(t_0 - 2) + t_2(t_0 - 3) + \cdots + t_{|\mathcal{R}-2|}(t_0 - |\mathcal{R}| + 1) \quad (3-5)$$

$$n_{tk} := t_0 \cdot \left( \frac{t_0}{2} + \sum_{i=1}^{|\mathcal{R}|-2} t_i \right) \quad (3-6)$$

In Algorithmus 3-4 wird die erste Überprüfung

$$\text{falls } (\{\gamma_{l_1}, \gamma_{l_2}\}, \{\mu_1\}) \in \mathbf{A}$$

$\frac{1}{2} \cdot t_0 \cdot (t_0 - 1)$  Mal ausgeführt (Lemma B.100). Daraus ergibt sich der 1. Summand  $t_0/2$  in Gleichung 3-6. Gleichung 3-6 vernachlässigt, dass in der Suchschleife mit der Laufvariablen  $l_i, i \in \{3, 4, \dots, |\mathcal{R}|\}$  die Indices  $l_1, l_2, \dots, l_{i-1}$  ausgeschlossen werden können und statt  $t_0$  nur  $t_0 - i + 1$  Durchläufe benötigt werden. Da  $|\mathcal{R}| \ll |\mathcal{T}| = t_0$  gelten soll, ist diese Vereinfachung gerechtfertigt. Außerdem werden die Indices nur aus logischen Gründen ausgeschlossen<sup>7</sup> und nicht zur Reduzierung des Aufwandes.

**Suche mit Speicherung der Zwischenergebnisse** Bei Speicherung der Zwischenergebnisse können die  $|\mathcal{R}|$  ineinander geschachtelten Schleifen von Algorithmus 3-4 durch  $|\mathcal{R}| - 1$  Schleifenpaare ersetzt werden. Diese Analyse mit Speicherung der Zwischenergebnisse kann als (teilweise) zuordnende Analyse beschrieben werden (Gl. 2-8, 2-19).

$$\begin{aligned}
& \mathcal{N}_1 := \mathcal{N}_2 := \cdots := \mathcal{N}_{|\mathcal{R}-2} := \emptyset \\
& \text{für } l_1 := 1, |\mathcal{T}| - 1 \\
& \quad \text{für } l_2 := l_1 + 1, |\mathcal{T}| \\
& \quad \quad \text{falls } (\{\gamma_{l_1}, \gamma_{l_2}\}, \{\mu_1\}) \in \mathbf{A} \text{ mit } \gamma_{l_1}, \gamma_{l_2} \in \mathcal{T} \\
& \quad \quad \quad \text{speichere Zwischenergebnis der 1. nicht terminalen Ebene} \\
& \quad \quad \quad \mathcal{N}_1 := \mathcal{N}_1 \cup \{\mu_1\} \\
& \quad \quad \quad \text{speichere Indices } l_1 \text{ und } l_2 \text{ bei } \mu_1 \\
& \text{für } l_1 := 1, |\mathcal{N}_1| \\
& \quad \text{für } l_3 := 1, |\mathcal{T}| \text{ ohne Vorgängerindices von } \mu_{1,l_1} \in \mathcal{N}_1 \\
& \quad \quad \text{falls } (\{\gamma_{l_3}, \mu_{1,l_1}\}, \{\mu_2\}) \in \mathbf{A} \text{ mit } \gamma_{l_3} \in \mathcal{T} \text{ und } \mu_{1,l_1} \in \mathcal{N}_1 \\
& \quad \quad \quad \text{speichere Zwischenergebnis der 2. nicht terminalen Ebene} \\
& \quad \quad \quad \mathcal{N}_2 := \mathcal{N}_2 \cup \{\mu_2\} \\
& \quad \quad \quad \text{speichere Indices von } \mu_1 \text{ und Index } l_3 \text{ bei } \mu_2 \\
& \quad \quad \quad \dots \\
& \text{für } l_1 := 1, |\mathcal{N}_{|\mathcal{R}-2}| \\
& \quad \text{für } l_{|\mathcal{R}|} := 1, |\mathcal{T}| \text{ ohne Vorgängerindices von } \mu_{|\mathcal{R}-2,l_1} \in \mathcal{N}_{|\mathcal{R}-2} \\
& \quad \quad \text{falls } (\{\gamma_{l_{|\mathcal{R}|}}, \mu_{|\mathcal{R}-2,l_1}\}, \{\sigma\}) \in \mathbf{A} \text{ mit } \gamma_{l_{|\mathcal{R}|}} \in \mathcal{T} \text{ und } \mu_{|\mathcal{R}-2,l_1} \in \mathcal{N}_{|\mathcal{R}-2} \\
& \quad \quad \quad \text{ein Analyseergebnis } \sigma \text{ wurde gefunden}
\end{aligned} \quad (3-7)$$

Die Mengen  $\mathcal{N}_i$  mit den Zwischenergebnissen werden wie folgt geschrieben:  $\mathcal{N}_i := \{\mu_{i,1}, \mu_{i,2}, \dots, \mu_{i,|\mathcal{N}_i|}\}$ ,  $i \in \{1, 2, \dots, |\mathcal{R}| - 2\}$ . Mit  $t_0 = |\mathcal{T}|$  sowie  $t_i = |\mathcal{N}_i|$  erhält man wie für Algorithmus 3-4  $n_{tkp}$  (Gl. 3-5) bzw. für  $|\mathcal{T}| \gg |\mathcal{R}|$  mit guter Näherung  $n_{tk}$  Überprüfungen (Gl. 3-6).

<sup>7</sup>Damit wird sichergestellt, dass der Algorithmus ein Ersetzungsverfahren realisiert.

Bei Ableitungsketten ist die Anzahl der Zwischenergebnisse und Überprüfungen also unabhängig davon, ob die Zwischenergebnisse abgespeichert werden oder nicht.

### Analyse mit Ableitungsbäumen

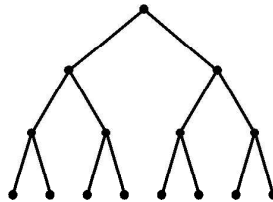
Die strukturelle Analyse mit Produktionsnetzen analysiert Teil-von-Relationen von Objekten, für die in der Regel Ableitungsbäume generiert werden. Zur Vereinfachung der anschließend formulierten Zusammenhänge werden nur regulär binäre balancierte Bäume (rbb-Bäume) berücksichtigt (Abbildung 3-2). Dabei wird die Menge der Assoziationen

$$\mathbf{A}_b := \bigcup_{i=1}^{ld(|\mathcal{R}|)} \left( \bigcup_{j=1}^{|\mathcal{R}|/2^i} \{ (\{\mu_{i-1,2j-1}, \mu_{i-1,2j}\}, \{\mu_{i,j}\}) \} \right) \quad (3-8)$$

betrachtet, die für  $|\mathcal{R}| = 8$  zusammenzufassende terminale Instanzen gleich

$$\begin{aligned} & \{ (\{\mu_{2,1}, \mu_{2,2}\}, \{\sigma\}) , \\ & (\{\mu_{1,1}, \mu_{1,2}\}, \{\mu_{2,1}\}) , (\{\mu_{1,3}, \mu_{1,4}\}, \{\mu_{2,2}\}) , \\ & (\{\rho_1, \rho_2\}, \{\mu_{1,1}\}) , (\{\rho_3, \rho_4\}, \{\mu_{1,2}\}) , (\{\rho_5, \rho_6\}, \{\mu_{1,3}\}) , (\{\rho_7, \rho_8\}, \{\mu_{1,4}\}) \} \end{aligned} \quad (3-9)$$

ist. Die terminalen Instanzen  $\rho$  und das Startsymbol  $\sigma$  von Gleichung 3-9 werden



**Abb. 3-2:** rbb-Ableitungsbaum  
(Beispiel mit 8 Blättern)

in Gleichung 3-8 geschrieben  $\rho_j := \mu_{0,j} \in \mathcal{R}; j = 1, \dots, |\mathcal{R}|$  und  $\mu_{ld(|\mathcal{R}|),1} = \sigma$ .

**Suche ohne Speicherung der Zwischenergebnisse** Die Elemente einer in der Menge  $\mathcal{T}$  zu findenden Konfiguration  $\mathcal{R}$  können mit  $|\mathcal{R}|$  geschachtelten Schleifen gesucht werden. Dabei werden die die Suche begrenzenden Überprüfungen als Baum entsprechend Gleichung 3-9 bzw. Abbildung 3-2 organisiert.

In /Poo98/ Seite 119 und 125 wird gezeigt, dass eine Tiefensuche grundsätzlich ersetzend arbeitet, so dass die Analyse ohne Speicherung der Zwischenergebnisse als (teilweise) ersetzende Analyse beschrieben werden kann (Gl. 2-17, 2-18). Die ersetzten terminalen Instanzen sind durch die Laufindices der erreichten Suchschleifen definiert. Die aktuell jeweils vorliegende Instanz ergibt sich aus der in Algorithmus 3-10 erreichten Zeile.

Die ohne permanente Speicherung der Zwischenergebnisse  $\nu_{i,j}$  notwendige temporäre Zwischenspeicherung kann mit einem Kellerspeicher /Ste88/ erfolgen. Dazu muss in den Algorithmen 3-10 und 3-17 jede Zeile

$\nu_{i,j}$  liegt vor

ersetzt werden durch die Zeile

erhöhe Kellerzeiger und speichere  $v_{i,j}$  im Keller ab.

Dies wird oft mit "push  $v_{i,j}$ " abgekürzt. Außerdem müssen bei Einsatz eines Kellerspeichers vor der Ausführung der Zeile

falls  $(\{\nu_{i,j}, \nu_{i,j+1}\}, \{\nu_{i+1,(j+1)/2}\}) \in \mathbf{A}$

die Zeilen

hole  $\nu_{i,j+1}$  aus dem Kellerspeicher und erniedrige Kellerzeiger  
hole  $\nu_{i,j}$  aus dem Kellerspeicher und erniedrige Kellerzeiger

ausgeführt werden. Diese Folge wird oft mit "pop  $\nu_{i,j+1}$  und pop  $\nu_{i,j}$ " abgekürzt.

Im einfachsten Fall wird in jeder der geschachtelten Suchschleifen jede terminale Instanz überprüft. Zusätzlich werden für rbb-Bäume Algorithmen diskutiert, die mit Suchbereichen arbeiten. Diese Differenzierung dient u. a. zur Beschreibung spezieller Strategien von Verfahren mit bewerteten Analysezuständen, die mit dem A\*-Algorithmus optimieren /Qui97, Sag85/.

**Vollständige Suche ohne Speicherung der Zwischenergebnisse** Der folgende Algorithmus geht davon aus, dass vor der Suche alle zu analysierenden terminalen Instanzen  $\gamma_i$  von  $\mathcal{T}$  vorliegen.

für  $l_1 := 1, |\mathcal{T}| - 1$   
 für  $l_2 := l_1 + 1, |\mathcal{T}|$   
 falls  $(\{\gamma_{l_1}, \gamma_{l_2}\}, \{\nu_{1,1}\}) \in \mathbf{A}$   
 $\nu_{1,1}$  liegt vor,  
 für  $l_3 := 1, |\mathcal{T}| - 1$  ohne  $l_1, l_2$   
 für  $l_4 := l_3 + 1, |\mathcal{T}|$  ohne  $l_1, l_2$   
 falls  $(\{\gamma_{l_3}, \gamma_{l_4}\}, \{\nu_{1,2}\}) \in \mathbf{A}$   
 $\nu_{1,2}$  liegt vor  
 falls  $(\{\nu_{1,1}, \nu_{1,2}\}, \{\nu_{2,1}\}) \in \mathbf{A}$   
 $\nu_{2,1}$  liegt vor  
 ... (3 - 10)

für  $l_{|\mathcal{R}|-1} := 1, |\mathcal{T}| - 1$  ohne  $l_1, l_2, \dots, l_{|\mathcal{R}|-2}$   
 für  $l_{|\mathcal{R}|} := l_{|\mathcal{R}|-1} + 1, |\mathcal{T}|$  ohne  $l_1, l_2, \dots, l_{|\mathcal{R}|-2}$   
 falls  $(\{\gamma_{l_{|\mathcal{R}|-1}}, \gamma_{l_{|\mathcal{R}|}}\}, \{\nu_{1,|\mathcal{R}|/2}\}) \in \mathbf{A}$   
 $\nu_{1,|\mathcal{R}|/2}$  liegt vor  
 falls  $(\{\nu_{1,|\mathcal{R}|/2-1}, \nu_{1,|\mathcal{R}|/2}\}, \{\nu_{2,|\mathcal{R}|/4}\}) \in \mathbf{A}$   
 $\nu_{2,|\mathcal{R}|/4}$  liegt vor  
 ...  
 falls  $(\{\nu_{id(|\mathcal{R})-1,1}, \nu_{id(|\mathcal{R})-1,2}\}, \{\sigma\}) \in \mathbf{A}$   
 ein Analyseergebnis  $\sigma$  wurde gefunden

Mit  $t_0 = t_{0,j} = |\mathcal{T}|; j = 1, 2, \dots, |\mathcal{R}|$  und  $t_{i,j}^m$ , der Anzahl der Zwischenergebnisse  $\nu_{i,j}; i = 1, 2, \dots, \text{ld}(|\mathcal{R}|) - 1; j = 1, 2, \dots, |\mathcal{R}|/2^i$ , werden  $n_{tbop}$

$$n_{tbop} := \sum_{i=1}^{\text{ld}(|\mathcal{R}|)-1} \left( \sum_{j=1}^{|\mathcal{R}|/2^{(i+1)}} t_{i,2j}^m \right) + \frac{t_0 \cdot (t_0 - 1)}{2} \cdot \left[ 1 + \sum_{i=1}^{\text{ld}(|\mathcal{R}|)-1} \left( \sum_{j=1}^{|\mathcal{R}|/2^{(i+1)}} (t_{i,2j-1}^m - (2j-1) \cdot 2^i) \right) \right] \quad (3-11)$$

bzw. für  $|\mathcal{T}| \gg |\mathcal{R}|$  mit guter Näherung  $n_{tbo}$  Überprüfungen

$$n_{tbo} := \sum_{i=1}^{\text{ld}(|\mathcal{R}|)-1} \left( \sum_{j=1}^{|\mathcal{R}|/2^{(i+1)}} t_{i,2j}^m \right) + \frac{t_0^2}{2} \cdot \left[ 1 + \sum_{i=1}^{\text{ld}(|\mathcal{R}|)-1} \left( \sum_{j=1}^{|\mathcal{R}|/2^{(i+1)}} t_{i,2j-1}^m \right) \right] \quad (3-12)$$

benötigt.  $\nu_{i,j}$  sind die im Ablauf von Algorithmus 3-10 bearbeiteten Zwischenergebnisse. Für  $|\mathcal{T}| \gg |\mathcal{R}|$  gilt näherungsweise Gleichung 3-12, die vernachlässigt, dass in der Suchschleife mit der Laufvariablen  $l_k; k = 2, 3, \dots, |\mathcal{R}|$  statt  $t_0$  nur  $t_0 - k + 1$  Durchläufe benötigt werden (vgl. Bemerkung zu Gleichung 3-6).

Gleichung 3-12 wird mit Abbildung 3-3, die ihrerseits Algorithmus 3-10 beschreibt, erläutert: In der untersten Zeile bzw. der 0. Ebene ( $i = 0$ ) sind  $|\mathcal{R}|$  terminale Instanzen aufgeführt, aus denen paarweise Zwischenergebnisse  $\nu_{1,j}; j = 1, 2, \dots, |\mathcal{R}|/2$  entstehen. Der erste Durchlauf mit den Laufvariablen  $l_1$  und  $l_2$ , der durch den Summanden 1 in Gleichung 3-12 ( $\dots[1 + \dots]$ ) berücksichtigt wird, generiert Zwischenergebnisse  $\nu_{1,1}$ . Alle ungerade indizierten Zwischenergebnisse  $\nu_{i,2j-1}$  der Ebene  $i$  veranlassen, dass die zu analysierenden Instanzen von  $\mathcal{T}$  ein weiteres Mal durchsucht werden, um ein Zwischenergebnis der 1. Ebene zu suchen, so dass die  $\left(\frac{t_0^2}{2}\right) \cdot t_{i,2j-1}^m$  Überprüfungen des zweiten Terms von Gleichung 3-12 durchgeführt werden. Die Indices  $k_1$  und  $k_2$  der Laufvariablen der zu überprüfenden terminalen Instanzen berechnen sich dabei aus den Indices  $m$  und  $n$  der Zwischenergebnisse  $\nu_{m,n}$  zu  $k_1 = n \cdot 2^m + 1$  und  $k_2 = n \cdot 2^m + 2$  (bzw. mit  $\nu_{i,2j-1}$  zu  $k_1 = (2j-1) \cdot 2^i + 1$  und  $k_2 = (2j-1) \cdot 2^i + 2$  vgl. Gl. B-159). Wenn ein gerade indiziertes Zwischenergebnis  $\nu_{i,2j}$  generiert wird, wird geprüft, ob ein weiteres Zwischenergebnis der nächst höheren Ebene vorliegt, so dass die  $t_{i,2j}^m$  Überprüfungen des ersten Terms von Gleichung 3-12 durchgeführt werden (vgl. auch Abbildungen B-1 bis B-3).

Die Anzahl  $t_{i,j}^m$  der im Ablauf von Algorithmus 3-10 bearbeiteten Zwischenergebnisse  $\nu_{i,j}$  ist größer als die Anzahl  $t_{i,j}$  der aus der zu analysierenden Konfiguration  $\mathcal{T}$  ableitbaren (unterschiedlichen) Zwischenergebnisse, die in der Diskussion der anderen Algorithmen (z. B. Algorithmus 3-19) mit  $\mu_{i,j}$  bezeichnet werden. Die Anzahl der Mehrfachgenerierungen der  $t_{i,j}$  Zwischenergebnisse  $\nu_{i,j}$  ist  $m_{i,j}$ :

$$m_{i,j} = t_{i,j}^m / t_{i,j} \quad (3-13)$$

Im Abschnitt B.4 wird ein Algorithmus zur Berechnung der Mehrfachgenerierungskoeffizienten  $m_{i,j}$  angegeben (Algorithmus B-161). Mit der Anzahl  $t_{i,j}$  der einzelnen Zwischenergebnisse und den Gleichungen B-154 ff. (bzw. Algorithmus B-161)

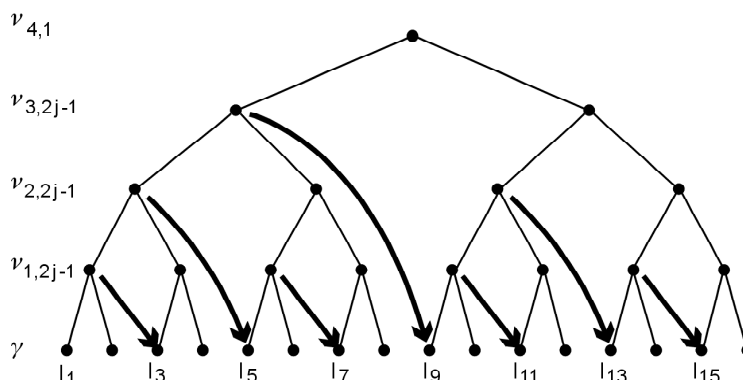


Abb. 3-3: Zur Anzahl der Überprüfungen mit Algorithmus 3-10

ergibt sich für die Mehrfachgenerierungskoeffizienten  $m_{i,j}$  der ersten vier Ebenen:

$$\begin{aligned}
 m_{4,1} &= m_{3,1} = m_{2,1} = m_{1,1} = 1 && = 1 \\
 & & m_{1,2} &= t_{1,1} && = t_{1,1} \\
 m_{2,2} &= m_{1,3} = m_{2,1} \cdot t_{2,1} && = t_{2,1} \\
 & & m_{1,4} &= m_{1,3} \cdot t_{1,3} && = t_{1,3} \cdot t_{2,1} \\
 m_{3,2} &= m_{2,3} = m_{1,5} = m_{3,1} \cdot t_{3,1} && = t_{3,1} \\
 & & m_{1,6} &= m_{1,5} \cdot t_{1,5} && = t_{1,5} \cdot t_{3,1} \\
 m_{2,4} &= m_{1,7} = m_{2,3} \cdot t_{2,3} && = t_{2,3} \cdot t_{3,1} \\
 & & m_{1,8} &= m_{1,7} \cdot t_{1,7} && = t_{1,7} \cdot t_{2,3} \cdot t_{3,1}
 \end{aligned} \tag{3-14}$$

In Abhängigkeit von den in Tabelle 3-1 angenommenen Werten für die Anzahl der Instanzen  $t_0$ ,  $t_1$ ,  $t_2$  und  $t_3$  (mit  $t_i := t_{i,1}$  und  $t_{i,1} \approx t_{i,j}; i = 1, 2, 3; j = 2, 3, \dots$ ) erhält man bei ersetzenden Systemen die in der Tabelle angegebenen Mehrfachgenerierungskoeffizienten. Die in dieser Arbeit betrachteten Anwendungen liegen in der Regel zwischen dem 5. und 6. Fall. Der 7. Fall arbeitet mit der maximalen Anzahl von Zwischenergebnissen und ist maschinell in der Regel nicht mehr zu bewältigen. Die Fälle 1. bis 4. geben Zahlenbeispiele.

Aus den Mehrfachgenerierungskoeffizienten  $m_{i,j}$  kann allerdings nicht unmittelbar der tatsächliche Mehraufwand, der durch Mehrfachgenerierungen entsteht, abgelesen werden. Beispielsweise ist der Aufwand im 7. Fall trotz der extremen Mehrfachgenerierungen bei einigen Zwischenergebnissen praktisch unabhängig von der Mehrfachgenerierung, d. h. davon, ob Zwischenergebnisse gespeichert werden oder nicht (vgl. Tabelle 3-2). Auch im 1. Fall ergeben sich nur relativ kleine Unterschiede. Hier dominiert die Anzahl der Zwischenergebnisse mit  $m \approx 1$ .

**Suche ohne Speicherung der Zwischenergebnisse mit Suchbereichen** Im vorhergehenden Abschnitt werden zur Analyse einer Instanz  $\alpha$  alle Instanzen der gleichen Stufe überprüft. Durch die Einführung von Suchbereichen soll die Anzahl der zu überprüfenden Instanzen reduziert werden.

Die in Algorithmus 3-17 verwendete Funktion *SuchbereichPartner*( $\alpha$ ) bestimmt die Instanzen des Suchbereichs passend zu dem bereits vorliegenden Zwi-

angenommene	$t_0$	30	100	$10^3$	$10^5$	$t_0$	$t_0$	$t_0$
Anzahl	$t_1$	10	30	100	$10^5$	$\frac{t_0}{10}$	$t_0$	$t_0^2$
der	$t_2$	3	10	10	$10^5$	$\frac{t_0}{100}$	$t_0$	$t_0^4$
Instanzen	$t_3$	1	3	1	100	$\frac{t_0}{1000}$	$t_0$	$t_0^8$
Fall		1.	2.	3.	4.	5.	6.	7.
aus	$m_{4,1}$	$m_{3,1}$	$m_{2,1}$	$m_{1,1}$	1	1	1	1
$t_0, t_1, t_2, t_3$				$m_{1,2}$	10	30	100	$10^5$
jeweils berechnete			$m_{2,2}$	$m_{1,3}$	3	10	10	$10^5$
Anzahl der				$m_{1,4}$	30	300	$10^3$	$10^{10}$
Mehrfach-	$m_{3,2}$	$m_{2,3}$	$m_{1,5}$		1	3	1	100
generierungen			$m_{1,6}$		10	90	100	$10^7$
mit		$m_{2,4}$	$m_{1,7}$		3	30	10	$10^7$
$t_{i,j}^m = m_{i,j} \cdot t_{i,j}$			$m_{1,8}$		30	900	$10^3$	$10^{13}$

Mehrfachgenerierungen mit dem ersetzenden Algorithmus 3-10  
**Tabelle 3-1:** und rbb-Ableitungsbäumen

schenergebnis  $\alpha$ :

$$\begin{aligned}
 & \text{SuchbereichPartner} : \mathcal{I} \rightarrow \wp(\mathcal{I}) \\
 & \alpha \mapsto \text{SuchbereichPartner}(\alpha) \\
 & \text{SuchbereichPartner}(\alpha) := \left\{ \beta \mid \begin{array}{l} \exists \gamma \in \mathcal{I} : \\ (\{\alpha, \beta\}, \{\gamma\}) \in \mathbf{A} \end{array} \right\} \tag{3-15}
 \end{aligned}$$

Die Funktion  $Teile(\mathcal{X})$  bestimmt diejenigen Instanzen, aus denen Instanzen der Menge  $\mathcal{X}$  abgeleitet werden:

$$\begin{aligned}
 & \text{Teile} : \wp(\mathcal{I}) \rightarrow \wp(\mathcal{I}) \\
 & \mathcal{X} \mapsto \text{Teile}(\mathcal{X}) \\
 & \text{Teile}(\mathcal{X}) := \left\{ \alpha, \beta \mid \begin{array}{l} \exists \gamma \in \mathcal{X} : \\ (\{\alpha, \beta\}, \{\gamma\}) \in \mathbf{A} \end{array} \right\} \tag{3-16}
 \end{aligned}$$

In realisierten Anwendungen wird man die Funktionen *SuchbereichPartner* und *Teile* durch vereinfachte anwendungsspezifische Funktionen ersetzen, mit denen Mengen  $\mathcal{T}'$  bereitgestellt werden, die die mit *SuchbereichPartner* und *Teile* jeweils generierten Mengen  $\mathcal{T}$  enthalten.

$\mathcal{T}_1$  ist die zur Analyse der ersten Zwischenergebnisse  $\nu_{1,1}$  bereitgestellte Menge terminaler Instanzen.

$$\begin{aligned}
& \text{für } l_1 := 1, |\mathcal{T}_1| \\
& \quad \mathcal{T}_2 := \text{SuchbereichPartner}(\gamma_{l_1}) \\
& \text{für } l_2 := 1, |\mathcal{T}_2| \\
& \quad \text{falls } (\{\gamma_{l_1}, \gamma_{l_2}\}, \{\nu_{1,1}\}) \in \mathbf{A} \text{ mit } \gamma_{l_1} \in \mathcal{T}_1 \text{ und } \gamma_{l_2} \in \mathcal{T}_2 \\
& \quad \quad \mathcal{T}_{\nu_{1,2}} := \text{SuchbereichPartner}(\nu_{1,1}) \\
& \quad \quad \mathcal{T}_3 := \text{Teile}(\mathcal{T}_{\nu_{1,2}}) \\
& \quad \text{für } l_3 := 1, |\mathcal{T}_3| \\
& \quad \quad \mathcal{T}_4 := \text{SuchbereichPartner}(\gamma_{l_3}) \\
& \quad \text{für } l_4 := 1, |\mathcal{T}_4| \\
& \quad \quad \text{falls } (\{\gamma_{l_3}, \gamma_{l_4}\}, \{\nu_{1,2}\}) \in \mathbf{A} \text{ mit } \gamma_{l_3} \in \mathcal{T}_3 \text{ und } \gamma_{l_4} \in \mathcal{T}_4 \\
& \quad \quad \quad \text{falls } (\{\nu_{1,1}, \nu_{1,2}\}, \{\nu_{2,1}\}) \in \mathbf{A} \\
& \quad \quad \quad \quad \mathcal{T}_{\nu_{2,2}} := \text{SuchbereichPartner}(\nu_{2,1}) \\
& \quad \quad \quad \quad \mathcal{T}_{5,6} := \text{Teile}(\mathcal{T}_{\nu_{2,2}}) \\
& \quad \quad \quad \quad \mathcal{T}_5 := \text{Teile}(\mathcal{T}_{5,6}) \\
& \quad \quad \quad \text{für } l_5 := 1, |\mathcal{T}_5| \\
& \quad \quad \quad \quad \mathcal{T}_6 := \text{SuchbereichPartner}(\gamma_{l_5}) \\
& \quad \quad \quad \text{für } l_6 := 1, |\mathcal{T}_6| \\
& \quad \quad \quad \vdots \\
& \quad \mathcal{T}_{\nu_{1,|\mathcal{R}|/2}} := \text{SuchbereichPartner}(\nu_{1,|\mathcal{R}|/2-1}) \\
& \quad \mathcal{T}_{|\mathcal{R}|-1} := \text{Teile}(\mathcal{T}_{\nu_{1,|\mathcal{R}|/2}}) \\
& \text{für } l_{|\mathcal{R}|-1} := 1, |\mathcal{T}_{|\mathcal{R}|-1}| \\
& \quad \mathcal{T}_{|\mathcal{R}|} := \text{SuchbereichPartner}(\gamma_{l_{|\mathcal{R}|-1}}) \\
& \text{für } l_{|\mathcal{R}|} := 1, |\mathcal{T}_{|\mathcal{R}|}| \\
& \quad \text{falls } (\{\gamma_{l_{|\mathcal{R}|-1}}, \gamma_{l_{|\mathcal{R}|}}\}, \{\nu_{1,|\mathcal{R}|/2}\}) \in \mathbf{A} \text{ mit } \gamma_{l_{|\mathcal{R}|-1}} \in \mathcal{T}_{|\mathcal{R}|-1} \text{ und } \gamma_{l_{|\mathcal{R}|}} \in \mathcal{T}_{|\mathcal{R}|} \\
& \quad \quad \text{falls } (\{\nu_{1,|\mathcal{R}|/2-1}, \nu_{1,|\mathcal{R}|/2}\}, \{\nu_{2,|\mathcal{R}|/4}\}) \in \mathbf{A} \\
& \quad \quad \dots \\
& \quad \quad \text{falls } (\{\nu_{id(|\mathcal{R}|-1,1)}, \nu_{id(|\mathcal{R}|-1,2)}\}, \{\sigma\}) \in \mathbf{A} \\
& \quad \quad \quad \text{ein Analyseergebnis } \sigma \text{ wurde gefunden}
\end{aligned} \tag{3-17}$$

Unter der Annahme, dass in den Suchbereichen  $\mathcal{T}_i; i > 1$  näherungsweise konstant  $s$  Instanzen enthalten sind und in der Startkonfiguration  $\mathcal{T}_1$  die Anzahl der terminalen Instanzen  $t_0$  ist, erhält man analog Gleichung 3-12

$$n_{tbos}(s) := \sum_{i=1}^{id(|\mathcal{R}|-1)} \left( \sum_{j=1}^{|\mathcal{R}|/2^{(i+1)}} t_{i,2j}^m \right) + s \cdot t_0 + s^2 \cdot \sum_{i=1}^{id(|\mathcal{R}|-1)} \left( \sum_{j=1}^{|\mathcal{R}|/2^{(i+1)}} t_{i,2j-1}^m \right) \tag{3-18}$$

Überprüfungen. Der Summand  $s \cdot t_0$  in Gleichung 3-18 entsteht dabei aus  $\frac{t_0^2}{2}[1 \dots$  in Gleichung 3-12.

**Mit Speicherung der Zwischenergebnisse** Bei Speicherung der Zwischenergebnisse können die  $|\mathcal{R}|$  geschachtelten Schleifen von Algorithmus 3-10 durch mehrere nacheinander auszuführende Schleifenpaare ersetzt werden. Diese Organisation entspricht dem bei der Analyse von  $\mathcal{T}$  entstehenden Ableitungsbaum. Die Analyse mit Speicherung der Zwischenergebnisse kann als zuordnende Analyse be-



geschrieben werden (Gl. 2-8).

**Vollständige Suche mit Speicherung der Zwischenergebnisse** Der folgende Algorithmus geht davon aus, dass zur Überprüfung einer Instanz einer Stufe  $i$  alle anderen Instanzen dieser Stufe überprüft werden.

$$\begin{aligned}
& \mathcal{N}_{i,j} := \emptyset; j = 1, 2, \dots, \frac{|\mathcal{R}|}{2^i}; i = 1, 2, \dots, ld(|\mathcal{R}|) - 1 \\
& \text{für } l_1 := 1, |\mathcal{T}| - 1 \\
& \quad \text{für } l_2 := l_1 + 1, |\mathcal{T}| \\
& \quad \quad \text{mit } \gamma_{l_1}, \gamma_{l_2} \in \mathcal{T} \\
& \quad \quad \text{falls } (\{\gamma_{l_1}, \gamma_{l_2}\}, \{\mu_{1,1}\}) \in \mathbf{A} \\
& \quad \quad \quad \mathcal{N}_{1,1} := \mathcal{N}_{1,1} \cup \{\mu_{1,1}\} \\
& \quad \quad \text{falls } (\{\gamma_{l_1}, \gamma_{l_2}\}, \{\mu_{1,2}\}) \in \mathbf{A} \\
& \quad \quad \quad \mathcal{N}_{1,2} := \mathcal{N}_{1,2} \cup \{\mu_{1,2}\} \\
& \quad \quad \quad \dots \\
& \quad \quad \text{falls } (\{\gamma_{l_1}, \gamma_{l_2}\}, \{\mu_{1,|\mathcal{R}|/2}\}) \in \mathbf{A} \\
& \quad \quad \quad \mathcal{N}_{1,|\mathcal{R}|/2} := \mathcal{N}_{1,|\mathcal{R}|/2} \cup \{\mu_{1,|\mathcal{R}|/2}\} \\
& \text{für } l_1 := 1, |\mathcal{N}_{1,1}| \\
& \quad \text{für } l_2 := 1, |\mathcal{N}_{1,2}| \\
& \quad \quad \text{falls } (\{\mu_{1,1,l_1}, \mu_{1,2,l_2}\}, \{\mu_{2,1}\}) \in \mathbf{A} \text{ mit } \mu_{1,1,l_1} \in \mathcal{N}_{1,1} \text{ und } \mu_{1,2,l_2} \in \mathcal{N}_{1,2} \\
& \quad \quad \quad \mathcal{N}_{2,1} := \mathcal{N}_{2,1} \cup \{\mu_{2,1}\} \\
& \quad \quad \quad \dots \\
& \text{für } l_1 := 1, |\mathcal{N}_{ld(|\mathcal{R})-1,1}| \\
& \quad \text{für } l_2 := 1, |\mathcal{N}_{ld(|\mathcal{R})-1,2}| \\
& \quad \quad \text{falls } (\{\mu_{ld(|\mathcal{R})-1,1,l_1}, \mu_{ld(|\mathcal{R})-1,2,l_2}\}, \{\sigma\}) \in \mathbf{A} \\
& \quad \quad \quad \text{mit } \mu_{ld(|\mathcal{R})-1,1,l_1} \in \mathcal{N}_{ld(|\mathcal{R})-1,1} \text{ und } \mu_{ld(|\mathcal{R})-1,2,l_2} \in \mathcal{N}_{ld(|\mathcal{R})-1,2} \\
& \quad \quad \quad \text{ein Analyseergebnis } \sigma \text{ wurde gefunden}
\end{aligned} \tag{3-19}$$

Die Mengen  $\mathcal{N}_{i,j}$  mit den Zwischenergebnissen werden wie folgt geschrieben:  
 $\mathcal{N}_{i,j} := \{\mu_{i,j,1}, \mu_{i,j,2}, \dots, \mu_{i,j,|\mathcal{N}_{i,j}|}\}; j = 1, 2, \dots, |\mathcal{R}|/2^{i-1}; i = 1, 2, \dots, ld(|\mathcal{R}|).$

Mit  $ld(|\mathcal{R}|)$  Stufen und der Festlegung  $t_0 := t_{0,j} := |\mathcal{T}|; j = 1, 2, \dots, |\mathcal{R}|/2$  sowie  $t_{i,j} := |\mathcal{N}_{i,j}|; j = 1, 2, \dots, |\mathcal{R}|/2^{i-1}; i = 1, 2, \dots, ld(|\mathcal{R}|)$  und  $t_0 \gg 1$  ist die Anzahl der Überprüfungen näherungsweise gleich  $n_{t_{bm}}$ :

$$n_{t_{bm}} := \frac{|\mathcal{R}|}{2} \cdot \frac{t_0^2}{2} + \sum_{i=2}^{ld(|\mathcal{R}|)} \left( \sum_{j=1}^{|\mathcal{R}|/2^i} t_{i-1,2j-1} \cdot t_{i-1,2j} \right) \tag{3-20}$$

Wird die Forderung  $t_0 \gg 1$  fallengelassen, so muss statt  $\frac{t_0^2}{2}$  geschrieben werden  $\frac{t_0 \cdot (t_0 - 1)}{2}$  und es gilt:

$$n_{t_{bmp}} := \frac{|\mathcal{R}|}{2} \cdot \frac{t_0 \cdot (t_0 - 1)}{2} + \sum_{i=2}^{ld(|\mathcal{R}|)} \left( \sum_{j=1}^{|\mathcal{R}|/2^i} t_{i-1,2j-1} \cdot t_{i-1,2j} \right) \tag{3-21}$$

**Suche mit Speicherung der Zwischenergebnisse und mit Suchbereichen**  
 Analog Algorithmus 3-17 und Algorithmus 3-19 erhält man bei Speicherung der Zwischenergebnisse und Suchbereichen den folgenden Algorithmus:

$$\begin{aligned}
& \mathcal{N}_{i,j} := \emptyset; j = 1, 2, \dots, \frac{|\mathcal{R}|}{2^i}; i = 1, 2, \dots, ld(|\mathcal{R}|) - 1 \\
& \text{für } l_1 := 1, |\mathcal{T}_{0,1}| \\
& \quad \mathcal{T}_{0,2} := \text{SuchbereichPartner}(\gamma_{l_1}) \\
& \quad \text{für } l_2 := 1, |\mathcal{T}_2| \\
& \quad \quad \text{mit } \gamma_{l_1} \in \mathcal{T}_{0,1} \text{ und } \gamma_{l_2} \in \mathcal{T}_{0,2} \\
& \quad \quad \text{falls } (\{\gamma_{l_1}, \gamma_{l_2}\}, \{\mu_{1,1}\}) \in \mathbf{A} \\
& \quad \quad \quad \mathcal{N}_{1,1} := \mathcal{N}_{1,1} \cup \{\mu_{1,1}\} \\
& \quad \quad \text{falls } (\{\gamma_{l_1}, \gamma_{l_2}\}, \{\mu_{1,2}\}) \in \mathbf{A} \\
& \quad \quad \quad \mathcal{N}_{1,2} := \mathcal{N}_{1,2} \cup \{\mu_{1,2}\} \\
& \quad \quad \quad \dots \\
& \quad \quad \text{falls } (\{\gamma_{l_1}, \gamma_{l_2}\}, \{\mu_{1,|\mathcal{R}|/2}\}) \in \mathbf{A} \\
& \quad \quad \quad \mathcal{N}_{1,|\mathcal{R}|/2} := \mathcal{N}_{1,|\mathcal{R}|/2} \cup \{\mu_{1,|\mathcal{R}|/2}\} \\
& \text{für } l_1 := 1, |\mathcal{N}_{1,1}| \\
& \quad \mathcal{T}_{1,1} := \text{SuchbereichPartner}(\mu_{1,1,l_1}) \\
& \quad \text{für } l_2 := 1, |\mathcal{T}_{1,1}| \\
& \quad \quad \text{falls } (\{\mu_{1,1,l_1}, \mu_{1,2,l_2}\}, \{\mu_{2,1}\}) \in \mathbf{A} \text{ mit } \mu_{1,1,l_1} \in \mathcal{N}_{1,1} \text{ und } \mu_{1,2,l_2} \in \mathcal{T}_{1,1} \\
& \quad \quad \quad \mathcal{N}_{2,1} := \mathcal{N}_{2,1} \cup \{\mu_{2,1}\} \\
& \quad \quad \quad \dots \\
& \text{für } l_1 := 1, |\mathcal{N}_{id(|\mathcal{R})-1,1}| \\
& \quad \mathcal{T}_{|\mathcal{R}|-1,1} := \text{SuchbereichPartner}(\mu_{id(|\mathcal{R})-1,1,l_1}) \\
& \quad \text{für } l_2 := 1, |\mathcal{T}_{|\mathcal{R}|-1,1}| \\
& \quad \quad \text{falls } (\{\mu_{id(|\mathcal{R})-1,1,l_1}, \mu_{id(|\mathcal{R})-1,2,l_2}\}, \{\sigma\}) \in \mathbf{A} \\
& \quad \quad \quad \text{mit } \mu_{id(|\mathcal{R})-1,1,l_1} \in \mathcal{N}_{id(|\mathcal{R})-1,1} \text{ und } \mu_{id(|\mathcal{R})-1,2,l_2} \in \mathcal{T}_{|\mathcal{R}|,1} \\
& \quad \quad \quad \text{ein Analyseergebnis } \sigma \text{ wurde gefunden}
\end{aligned} \tag{3-22}$$

Unter der Annahme, dass in den Suchbereichen  $\mathcal{T}_{0,2}$  und  $\mathcal{T}_{i,j}; i > 0$  näherungsweise konstant  $s$  Instanzen enthalten sind und in der Startkonfiguration  $\mathcal{T}_{0,1}$  die Anzahl der terminalen Instanzen  $t_0$  ist, erhält man analog Gleichung 3-20

$$n_{tbms}(s) := s \cdot \left( \frac{t_0 \cdot |\mathcal{R}|}{2} + \sum_{i=2}^{ld(|\mathcal{R}|)} \left( \sum_{j=1}^{|\mathcal{R}|/2^i} t_{i-1,2j-1} \right) \right) \tag{3-23}$$

Überprüfungen.

### Vergleich der Algorithmen

**Ableitungsketten** Unabhängig davon, ob die Zwischenergebnisse gespeichert werden oder nicht, erhält man mit den Algorithmen 3-4 und 3-7 bei Ableitungsketten  $n_{tk}$  Überprüfungen (Gl. 3-6):

$$n_{tk} = t_0 \cdot \left( \frac{t_0}{2} + \sum_{i=1}^{|\mathcal{R}|-2} t_i \right)$$

**Ableitungsbäume** Bei rbb-Ableitungsbäumen hängt der Aufwand davon ab, ob Zwischenergebnisse gespeichert werden oder nicht.

**Vollständige Suche ohne Speicherung der Zwischenergebnisse** Im einfachsten Fall werden vor der Suche alle terminalen Instanzen von  $\mathcal{T}$  aus den Sen-

sordaten extrahiert. Bei dieser vollständigen Suche werden ohne Speicherung der Zwischenergebnisse (Algorithmus 3-10) mit  $t_0 = t_{0,j} := |\mathcal{T}|, j \in \mathbb{N}$  und  $t_{i,j}^m := m_{i,j} \cdot |\mathcal{N}_{i,j}|$  insgesamt  $n_{tbo}$  Überprüfungen durchgeführt (Gl. 3-12):

$$n_{tbo} = \sum_{i=1}^{ld(|\mathcal{R}|-1)} \left( \sum_{j=1}^{|\mathcal{R}|/2^{(i+1)}} t_{i,2j}^m \right) + \frac{t_0^2}{2} \cdot \left[ 1 + \sum_{i=1}^{ld(|\mathcal{R}|-1)} \left( \sum_{j=1}^{|\mathcal{R}|/2^{(i+1)}} t_{i,2j-1}^m \right) \right]$$

Algorithmus B-161 ersetzt die nur für Algorithmus 3-10 verwendeten Größen  $t_{i,j}^m = m_{i,j} \cdot t_{i,j}$  (Gl. 3-13), die von Mehrfachgenerierungen  $m_{i,j}$  abhängen, durch die Größen  $t_{i,j}$ , die unabhängig von Mehrfachgenerierungen sind und auch für Algorithmus 3-19, der Zwischenergebnisse speichert, verwendet werden. Für  $|\mathcal{R}| = 4$ ,  $|\mathcal{R}| = 8$  und  $|\mathcal{R}| = 16$  zusammenzufassende terminale Instanzen erhält man für  $n_{tbo}$  die Größen  $n_{tbo4}$  bis  $n_{tbo16}$ :

$$\begin{aligned} n_{tbo4} &:= t_{1,2}^m + \frac{t_0^2}{2} \cdot [1 + t_{1,1}^m] = \\ &= t_{1,1} \cdot t_{1,2} + \frac{t_0^2}{2} \cdot [1 + t_{1,1}] \end{aligned} \quad (3-24)$$

$$\begin{aligned} n_{tbo8} &:= t_{1,2}^m + t_{1,4}^m + t_{2,2}^m + \frac{t_0^2}{2} \cdot [1 + t_{1,1}^m + t_{1,3}^m + t_{2,1}^m] = \\ &= t_{1,1} \cdot t_{1,2} + t_{1,3} \cdot t_{1,4} \cdot t_{2,1} + t_{2,1} \cdot t_{2,2} + \frac{t_0^2}{2} \cdot [1 + t_{1,1} + t_{1,3} \cdot t_{2,1} + t_{2,1}] \end{aligned} \quad (3-25)$$

$$\begin{aligned} n_{tbo16} &:= t_{1,2}^m + t_{1,4}^m + t_{1,6}^m + t_{1,8}^m + t_{2,2}^m + t_{2,4}^m + t_{3,2}^m + \\ &+ \frac{t_0^2}{2} \cdot [1 + t_{1,1}^m + t_{1,3}^m + t_{1,5}^m + t_{1,7}^m + t_{2,1}^m + t_{2,3}^m + t_{3,1}^m] = \\ &= t_{1,1} \cdot t_{1,2} + t_{1,3} \cdot t_{1,4} \cdot t_{2,1} + t_{1,5} \cdot t_{1,6} \cdot t_{3,1} + t_{1,7} \cdot t_{1,8} \cdot t_{2,3} \cdot t_{3,1} + \\ &\quad + t_{2,1} \cdot t_{2,2} + t_{2,3} \cdot t_{2,4} \cdot t_{3,1} + \\ &\quad + t_{3,1} \cdot t_{3,2} + \\ &+ \frac{t_0^2}{2} \cdot [1 + t_{1,1} + t_{1,3} \cdot t_{2,1} + t_{1,5} \cdot t_{3,1} + t_{1,7} \cdot t_{2,3} \cdot t_{3,1} + t_{2,1} + t_{2,3} \cdot t_{3,1} + t_{3,1}] \end{aligned} \quad (3-26)$$

**Vollständige Suche mit Speicherung der Zwischenergebnisse** Bei Speicherung der Zwischenergebnisse mit Algorithmus 3-19 werden Mehrfachgenerierungen vermieden. Mit  $t_{i,j} := |\mathcal{N}_{i,j}|$  erhält man bei Speicherung der Zwischenergebnisse  $n_{tbm}$  Überprüfungen (Gl. 3-20):

$$n_{tbm} = \frac{|\mathcal{R}| \cdot t_0^2}{4} + \sum_{i=2}^{ld(|\mathcal{R}|)} \left[ \sum_{j=1}^{|\mathcal{R}|/2^i} (t_{i-1,2j-1} \cdot t_{i-1,2j}) \right]$$

Für  $|\mathcal{R}| = 4$ ,  $|\mathcal{R}| = 8$  und  $|\mathcal{R}| = 16$  zusammenzufassende terminale Instanzen erhält man  $n_{tbm4}$  bis  $n_{tbm16}$ :

$$\begin{aligned} n_{tbm4} &:= t_0^2 + \\ &+ t_{1,1} \cdot t_{1,2} \end{aligned} \quad (3-27)$$

$$\begin{aligned} n_{tbm8} &:= 2t_0^2 + \\ &+ t_{1,1} \cdot t_{1,2} + t_{1,3} \cdot t_{1,4} + t_{2,1} \cdot t_{2,2} \end{aligned} \quad (3-28)$$

$$\begin{aligned} n_{tbm16} &:= 4t_0^2 + \\ &+ t_{1,1} \cdot t_{1,2} + t_{1,3} \cdot t_{1,4} + t_{1,5} \cdot t_{1,6} + t_{1,7} \cdot t_{1,8} + \\ &+ t_{2,1} \cdot t_{2,2} + t_{2,3} \cdot t_{2,4} + \\ &+ t_{3,1} \cdot t_{3,2} \end{aligned} \quad (3-29)$$

**Vergleich der vollständigen Suche mit und ohne Speicherung der Zwischenergebnisse** In Tabelle 3-2 wird für einige Beispiele ohne (Algorithmus 3-10) und mit Speicherung der Zwischenergebnisse (Algorithmus 3-19) die Anzahl der Überprüfungen angegeben.  $t_0, t_1, t_2, t_3$  und  $t_4$  sind die Eingangsgrößen, die die sieben aufgeführten Fälle definieren.  $t_0$  ist die Anzahl der terminalen Instanzen der Menge  $\mathcal{T}$ .  $t_1, t_2, t_3$  und  $t_4$  sind die Anzahl der Zwischenergebnisse  $\mu_0, \mu_{1,j}, \mu_{2,j}, \mu_{3,j}$  und  $\mu_{4,j}$ . Dabei wird angenommen, dass die Anzahl  $t_{i,j}; j = 1, 2, \dots$  der Zwischenergebnisse  $\mu_{i,j}$  der Ebene  $i; i = 1, 2, \dots$  näherungsweise gleich ist, so dass mit  $t_i := t_{i,1}$  die Näherung  $t_i \approx t_{i,j}; j = 2, 3, \dots$  gilt. Aus der Anzahl der Zwischenergebnisse werden Näherungen für die Anzahl der Überprüfungen mit Speicherung ( $n_{t_{bm}4}, n_{t_{bm}8}$  und  $n_{t_{bm}16}$ ) und ohne Speicherung ( $n_{t_{bo}4}, n_{t_{bo}8}$  und  $n_{t_{bo}16}$ ) bei vier ( $n_{t_{bm}4}, n_{t_{bo}4}$ ), acht ( $n_{t_{bm}8}, n_{t_{bo}8}$ ) und 16 ( $n_{t_{bm}16}, n_{t_{bo}16}$ ) zu einem Endergebnis zusammenzusetzenden terminalen Instanzen berechnet. Für  $n_{t_{bm}4}$  bis  $n_{t_{bm}16}$  erhält man in diesem Fall die Größen  $n_{t_{bm}4a}$  bis  $n_{t_{bm}16a}$ :

$$n_{t_{bm}4a} := t_0^2 + t_1^2 \quad (3 - 30)$$

$$n_{t_{bm}8a} := 2t_0^2 + 2t_1^2 + t_2^2 \quad (3 - 31)$$

$$n_{t_{bm}16a} := 4t_0^2 + 4t_1^2 + 2t_2^2 + t_3^2 \quad (3 - 32)$$

Für  $n_{t_{bo}4}$  bis  $n_{t_{bo}16}$  erhält man die Größen  $n_{t_{bo}4a}$  bis  $n_{t_{bo}16a}$ :

$$n_{t_{bo}4a} := \frac{1}{2}(1 + t_1)t_0^2 + t_1^2 \quad (3 - 33)$$

$$n_{t_{bo}8a} := \frac{1}{2}(1 + t_1)(1 + t_2)t_0^2 + (1 + t_2)t_1^2 + t_2^2 \quad (3 - 34)$$

$$n_{t_{bo}16a} := \frac{1}{2}(1 + t_1)(1 + t_2)(1 + t_3)t_0^2 + (1 + t_2)(1 + t_3)t_1^2 + (1 + t_3)t_2^2 + t_3^2 \quad (3 - 35)$$

Für die Faktoren  $v_{t_{br}4} = n_{t_{bo}4}/n_{t_{bm}4}$ ,  $v_{t_{br}8} = n_{t_{bo}8}/n_{t_{bm}8}$  und  $v_{t_{br}16} = n_{t_{bo}16}/n_{t_{bm}16}$  erhält man als Näherungen  $v_{t_{br}4a}$  bis  $v_{t_{br}16a}$ :

$$v_{t_{br}4a} := \left[ \frac{1}{2}(1 + t_1) + (t_1/t_0)^2 \right] / \left[ 1 + (t_1/t_0)^2 \right] \quad (3 - 36)$$

$$v_{t_{br}8a} := \frac{\left[ \frac{1}{2}(1 + t_1)(1 + t_2) + (1 + t_2)(t_1/t_0)^2 + (t_2/t_0)^2 \right]}{\left[ 2 + 2(t_1/t_0)^2 + (t_2/t_0)^2 \right]} \quad (3 - 37)$$

$$v_{t_{br}16a} := \frac{\left[ \frac{1}{2}(1 + t_1)(1 + t_2)(1 + t_3) + (1 + t_2)(1 + t_3)(t_1/t_0)^2 + (1 + t_3)(t_2/t_0)^2 + (t_3/t_0)^2 \right]}{\left[ 4 + 4(t_1/t_0)^2 + 2(t_2/t_0)^2 + (t_3/t_0)^2 \right]} \quad (3 - 38)$$

Diese Größen  $v_{t_{br}4a}$  bis  $v_{t_{br}16a}$  nehmen für  $t_i = 1, i \geq 1$  ( $t_0$  beliebig) den Wert 1 an. Wenn es also auf allen Stufen jeweils nur ein Zwischenergebnis gibt, reduziert die Speicherung der Zwischenergebnisse den Aufwand nicht.

Die in Tabelle 3-2 aufgeführten Größen sind gerundet. Die Anzahl der zu einem Muster zusammenzufassenden terminalen Instanzen der Menge  $\mathcal{R}$  soll wesentlich kleiner sein als die Anzahl der insgesamt analysierten terminalen Instanzen der

$t_0$	30	100	$10^3$	$10^5$	$t_0$	$t_0$	$t_0$
$t_1$	10	30	100	$10^5$	$\frac{t_0}{10}$	$t_0$	$t_0^2$
$t_2$	3	10	10	$10^5$	$\frac{t_0}{100}$	$t_0$	$t_0^4$
$t_3$	1	3	1	1000	$\frac{t_0}{1000}$	$t_0$	$t_0^8$
$t_4$	–	1	–	10	$\frac{t_0}{10000}$	$t_0$	$t_0^{16}$
Fall	1.	2.	3.	4.	5.	6.	7.
$n_{tbm4a}$	$10^3$	$1,1 \cdot 10^4$	$10^6$	$2 \cdot 10^{10}$	$t_0^2$	$(1,4t_0)^2$	$t_0^4$
$n_{tbo4a}$	$5,1 \cdot 10^3$	$1,6 \cdot 10^5$	$5 \cdot 10^7$	$5 \cdot 10^{14}$	$(\frac{t_0}{1,4})^2 + (\frac{t_0}{2,7})^3$	$(1,2t_0)^2 + (\frac{t_0}{1,3})^3$	$\frac{3t_0^4}{2}$
$v_{tbr4a}$	5,1	14,3	50	$2,5 \cdot 10^4$	$\frac{t_0}{20}$	$\frac{t_0}{4}$	$\frac{3}{2}$
$n_{tbm8a}$	$2 \cdot 10^3$	$2,2 \cdot 10^4$	$2 \cdot 10^6$	$5 \cdot 10^{10}$	$(1,4t_0)^2$	$(2,2t_0)^2$	$t_0^8$
$n_{tbo8a}$	$2 \cdot 10^4$	$1,7 \cdot 10^6$	$5,6 \cdot 10^8$	$5 \cdot 10^{19}$	$(1,6t_0)^2 + (\frac{t_0}{2,2})^3 + (\frac{t_0}{5,6})^4$	$(1,6t_0)^2 + (1,3t_0)^3 + (\frac{t_0}{1,2})^4$	$\frac{5t_0^8}{2}$
$v_{tbr8a}$	10,1	78,3	275	$10^9$	$\frac{t_0}{36} + (\frac{t_0}{63})^2$	$\frac{t_0}{2,5} + (\frac{t_0}{3,2})^2$	$\frac{5}{2}$
$n_{tbm16a}$	–	$4,4 \cdot 10^4$	–	$10^{11}$	$(2,8t_0)^2$	$(3,3t_0)^2$	$t_0^{16}$
$n_{tbo16a}$	–	$6,9 \cdot 10^6$	–	$5 \cdot 10^{22}$	$t_0^2 + (\frac{t_0}{2,2})^3 + (\frac{t_0}{5,6})^4 + (\frac{t_0}{15,8})^5$	$(1,9t_0)^2 + (1,7t_0)^3 + (1,3t_0)^4 + (1,4t_0)^5$	$\frac{7t_0^{16}}{2}$
$v_{tbr16a}$	–	157	–	$5 \cdot 10^{11}$	$\frac{t_0}{80} + (\frac{t_0}{89})^2 + (\frac{t_0}{200})^3$	$\frac{t_0}{2,4} + (\frac{t_0}{2,1})^2 + (\frac{t_0}{2,8})^3$	$\frac{7}{2}$

Ohne bzw. mit Speicherung notwendige Anzahl der Überprüfungen mit dem ersetzenden Algorithmus 3-10 ( $n_{tbo\dots}$ ) bzw. dem zuordnenden Algorithmus 3-19 ( $n_{tbm\dots}$ ) und rbb-Ableitungsbäumen  
**Tabelle 3-2:** mit  $v_{tbr\dots} = n_{tbo\dots}/n_{tbm\dots}$

Menge  $\mathcal{T}$  ( $t_0 = |\mathcal{T}| \gg |\mathcal{R}|$ ). Außerdem sollen alle relevanten Zwischenergebnisse definiert sein, so dass sich die Anzahl der von der Analyse berücksichtigten Instanzen der Menge  $\mathcal{I}_a(\mathbf{A})$  nicht auswirkt (vgl. Seite 69).

Die in dieser Arbeit betrachteten Anwendungen liegen in der Regel zwischen dem 5. und 6. Fall. Der 7. Fall arbeitet mit der maximalen Anzahl von Zwischenergebnissen und ist maschinell in der Regel nicht mehr zu bewältigen. Die Fälle 1. bis 4. geben Zahlenbeispiele und zeigen z. B., dass bei wenigen Instanzen (1. Fall) der Unterschied der Anzahl der Überprüfungen relativ gering ist, während bei umfangreichen Analysen (4. Fall) nur bei Speicherung der Zwischenergebnisse mit Algorithmus 3-19 Ergebnisse berechnet werden können; Algorithmus 3-10 mit den Größen  $n_{tbo4a}$  bis  $n_{tbo16a}$  kann jetzt nicht mehr eingesetzt werden.

Die mit Algorithmus 3-10 benötigte Anzahl der Überprüfungen  $n_{tbo}$  ist für  $t_0 \gg |\mathcal{R}|$  und  $t_i > 1, i \geq 0$  immer größer als die mit Algorithmus 3-19 benötigte Anzahl  $n_{tbn}$  (Satz B.99), so dass bei umfangreichen zu analysierenden Datensätzen und rbb-Ableitungsbäumen durch Speicherung der Zwischenergebnisse die Anzahl der Überprüfungen reduziert wird.

**Suche mit Suchbereichen** Die Anzahl der Überprüfungen hängt u. a. von der Anzahl der Instanzen in den Suchbereichen ab.

Der eine Extremfall ist, dass die Suchbereiche alle Elemente übergeben ( $n_{tbo}$  Gl. 3-12 und  $n_{tbn}$  Gl. 3-20).

$$n_{tbo} = \sum_{i=1}^{ld(|\mathcal{R}|-1)} \binom{|\mathcal{R}|/2^{(i+1)}}{\sum_{j=1}^m t_{i,2j}^m} + \frac{t_0^2}{2} \cdot \left[ 1 + \sum_{i=1}^{ld(|\mathcal{R}|-1)} \binom{|\mathcal{R}|/2^{(i+1)}}{\sum_{j=1}^m t_{i,2j-1}^m} \right]$$

$$n_{tbn} = \frac{|\mathcal{R}|}{2} \cdot \frac{t_0^2}{2} + \sum_{i=2}^{ld(|\mathcal{R}|)} \binom{|\mathcal{R}|/2^i}{\sum_{j=1} t_{i-1,2j-1} \cdot t_{i-1,2j}}$$

Der andere Extremfall ist, dass in einem Suchbereich nur eine Instanz vorliegt (ohne Speicherung:  $n_{tbos}(1)$  Gl. 3-18; mit Speicherung  $n_{tbms}(1)$  Gl. 3-23).  $n_{tbos}(s)$  beschreibt die Situation mit  $s$  Instanzen je Suchbereich (Gl. 3-18), wenn keine Zwischenergebnisse gespeichert werden. Mit Speicherung der Zwischenergebnisse werden  $n_{tbms}(s)$  Überprüfungen benötigt (Gl. 3-23).

$$n_{tbos}(s) = \sum_{i=1}^{ld(|\mathcal{R}|-1)} \binom{|\mathcal{R}|/2^{(i+1)}}{\sum_{j=1}^m t_{i,2j}^m} + t_0 \cdot s + s^2 \cdot \sum_{i=1}^{ld(|\mathcal{R}|-1)} \binom{|\mathcal{R}|/2^{(i+1)}}{\sum_{j=1}^m t_{i,2j-1}^m}$$

$$n_{tbms}(s) = \frac{|\mathcal{R}|}{2} \cdot t_0 \cdot s + \sum_{i=2}^{ld(|\mathcal{R}|)} \binom{|\mathcal{R}|/2^i}{\sum_{j=1} t_{i-1,2j-1} \cdot s}$$

In Tabelle 3-3 wird für sechs Analyseanwendungen mit unterschiedlicher Anzahl von Instanzen die Anzahl der Überprüfungen angegeben.  $|\mathcal{R}|$  ist die Anzahl der terminalen Instanzen eines Endergebnisses.  $t_0$  ist die Anzahl der zu analysierenden terminalen Instanzen und  $t_i$  ist die Anzahl der Zwischenergebnisse der Stufe  $i$ .

$t_0$	30	100	$10^3$	$10^5$	$10^5$	$10^5$
$t_1$	10	30	100	$10^5$	$10^4$	$10^6$
$t_2$	3	10	10	$10^5$	$10^3$	$10^7$
$t_3$	1	3	1	1000	$10^2$	$10^6$
$t_4$	–	1	–	10	100	$10^5$
$t_5$	–	–	–	–	10	$10^4$
$t_6$	–	–	–	–	1	$10^3$
$t_7$	–	–	–	–	–	100
$t_8$	–	–	–	–	–	10
$t_9$	–	–	–	–	–	1
Fall	1.	2.	3.	4.	5.	6.
$ \mathcal{R} $	8	16	8	16	64	512
$n_{t_{bm}}$	$2 \cdot 10^3$	$4,4 \cdot 10^4$	$2 \cdot 10^6$	$10^{11}$	$1,6 \cdot 10^{13}$	$6,5 \cdot 10^{15}$
$n_{t_{bo}}$	$2 \cdot 10^4$	$6,9 \cdot 10^6$	$5,6 \cdot 10^8$	$5 \cdot 10^{22}$	$5,6 \cdot 10^{26}$	$5,6 \cdot 10^{44}$
$n_{t_{bms}}(1)$	143	943	$4,2 \cdot 10^3$	$1,4 \cdot 10^6$	$3,3 \cdot 10^7$	$8,2 \cdot 10^8$
$n_{t_{bos}}(1)$	482	$4,1 \cdot 10^4$	$1,1 \cdot 10^5$	$10^{18}$	$1,1 \cdot 10^{20}$	$1,1 \cdot 10^{40}$
$n_{t_{bos}}(1)/n_{t_{bm}}$	0,23	0,94	0,06	$10^7$	$6,8 \cdot 10^6$	$1,7 \cdot 10^{24}$

**Tabelle 3-3:** Anzahl der Überprüfungen mit Suchbereichen

Das Verhältnis  $n_{t_{bos}}(1)/n_{t_{bm}}$  verwendet für die Analyse ohne Speicherung der Zwischenergebnisse Suchbereiche mit nur einer Instanz. Für die Analyse mit Speicherung der Zwischenergebnisse werden Suchbereiche mit allen Instanzen angesetzt. Unter dieser für die Suche mit Speicherung der Zwischenergebnisse ungünstigsten Annahme benötigt im 4. Fall die Suche ohne Speicherung immer noch die  $10^7$ -fache Anzahl an Überprüfungen. Dies ist ein weiterer Hinweis darauf, dass bei Analysen mit sehr vielen Instanzen die Speicherung der Zwischenergebnisse die Anzahl der Überprüfungen entscheidend reduziert.

### **Einfluss der Mächtigkeit des Merkmalraumes auf den Aufwand**

Im ungünstigsten Fall entsteht bei den hier aufgeführten Algorithmen (z. B. Algorithmus 3-19) für jede Teilmenge  $\mathcal{R}$  der zu analysierenden Menge  $\mathcal{T}$  ein Ergebnis. Für  $|\mathcal{T}| \gg |\mathcal{R}|$  gibt es  $|\mathcal{T}|^{|\mathcal{R}|}$  Teilmengen, so dass bei der Analyse mit Produktionsnetzen exponentieller Aufwand entstehen kann. Dabei wird allerdings die

Anzahl der bei einer speziellen Analyse berücksichtigten Instanzen  $\mathcal{I}_a(\mathbf{A})$  (Gl. B-2), zu der auch alle Ergebnisse, die Teilmengen bezeichnen, gehören, vernachlässigt.

Wenn der Merkmalraum  $\mathcal{I}_a(\mathbf{A})$  reduziert ist, so dass z. B.  $|\mathcal{T}|^{|\mathcal{R}|} > |\mathcal{I}_a(\mathbf{A})|$  ist, können maximal noch  $|\mathcal{I}_a(\mathbf{A})|$  Ergebnisse entstehen. In diesem Fall werden bei der ersetzenden Erkennung (Algorithmus 3-4) Iterationen ausgelassen. Bei der zuordnenden Erkennung (Algorithmus 3-7) fehlen die nicht definierten Zwischenergebnisse.

### 3.3.2 Algorithmen mit bewerteten Analysezuständen

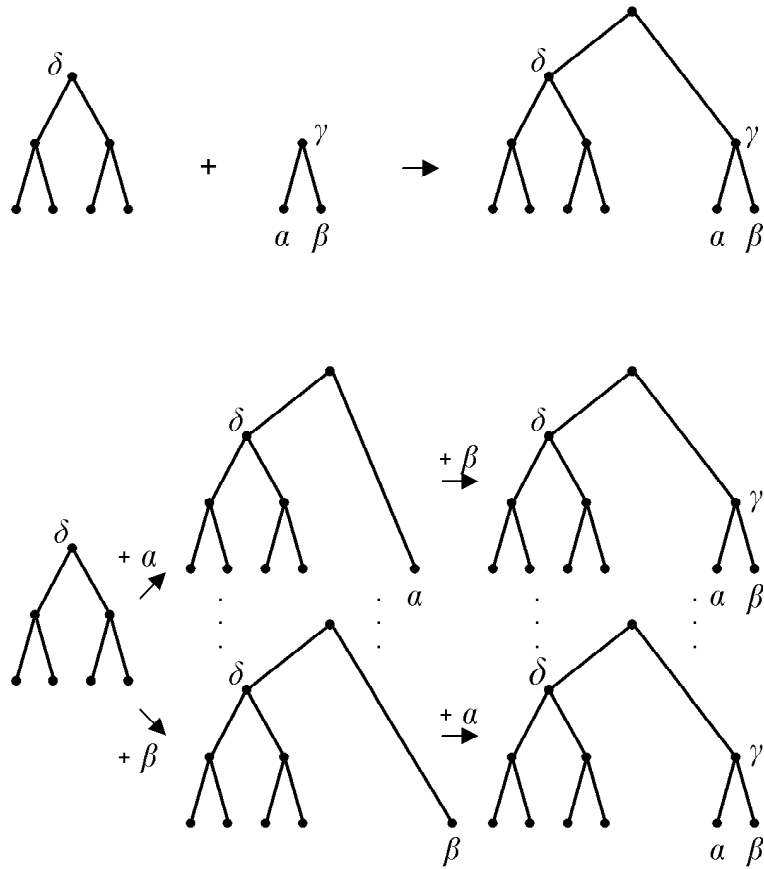
In den vorherigen Abschnitten werden Algorithmen erläutert, die Mengen von Instanzen vollständig durchsuchen. Dabei wird jeder mit den zu analysierenden terminalen Instanzen mögliche Ableitungsgraph berücksichtigt. Diese Ableitungsgraphen können bewertet und von Optimierungsverfahren zur weiteren Analyse gezielt ausgewählt werden. Werden Zwischenergebnisse als Ableitungen (Gl. B-54) beschrieben, so liegt der Ableitungsgraph (Gl. B-59 u. B-60) über die Verzweigung der Ableitungen direkt vor.

Wenn weder Zwischenergebnisse noch Analysezustände gespeichert werden, werden die Zwischenergebnisse mit ihren Ableitungsgraphen nacheinander aufgebaut. Bei Speicherung der Zwischenergebnisse (Algorithmus 3-19) werden bei der Generierung eines Zwischenergebnisses jeweils mehrere gespeicherte elementare Ableitungsgraphen (d. h. Instanzen) zu einem komplexen Ableitungsgraphen zusammengefasst. Bei Speicherung der Analysezustände (Algorithmus 3-39), die ebenfalls durch Ableitungsgraphen repräsentiert sind, entstehen aus jeweils einem Analysezustand ein oder mehrere neue Analysezustände. Abbildung 3-4 verdeutlicht diesen Sachverhalt für Analysen mit Ableitungsbäumen. Bei Ableitungsketten entsprechen die Analysezustände der ersetzenden Verfahren den Zwischenergebnissen der zuordnenden Verfahren.

Für Bewertungs- und Auswahlverfahren kann ausgehend von Algorithmus 3-10 in jeder Suchschleife jeder Durchlauf explizit als Analysezustand gespeichert werden, so dass eine Breitensuche /Poo98, Ben96/ organisiert wird, bei der Analysezustände bewertet und für eine aufwandsminimale Suche geeignet ausgewählt werden können. Jeder Analysezustand ist ein bewerteter Ableitungsgraph, der die bis zum aktuellen Stand abgelaufene Teilanalyse repräsentiert. Wird von einem Analysezustand zum nächsten übergegangen, so wird ein Ableitungsgraph um eine neue Instanz ergänzt (Abbildung 3-4 unten). Aus einer Analyseketten können so mehrere neue Ketten entstehen, so dass sich ein Baum von Analysezuständen ergibt (Abb. 2.4.2.2 in /Bun85/) /Qui97, Sag85/.

Werden beim Übergang von einem Analysezustand zum nächsten nur terminale Instanzen hinzugefügt und keine nichtterminale Instanzen, die in sich abgeschlossene Zwischenergebnisse repräsentieren, dann kann zur Generierung der Analysezustände z. B. entsprechend Algorithmus 3-39, der auf Algorithmus 3-10 aufbaut, vorgegangen werden.





**Abb. 3-4:** Generierung von Ableitungsgraphen  
 oben: zusammenfassend bei Speicherung nicht terminaler Instanzen  
 unten: aufspaltend bei Speicherung von Analysezuständen

Funktion Suche-bewertet-Stufe-2 ( $\alpha$ )  
 für  $l_2 := 1, |\mathcal{T}|$   
 falls  $(\{\alpha, \gamma_{l_2}\}, \{\nu_{1,1}\}) \in \mathbf{A}$   
 speichere (Bewertung( $\nu_{1,1}$ ), Suche-bewertet-Stufe-3,  $\nu_{1,1}$ )

Funktion Suche-bewertet-Stufe-3 ( $\alpha$ )  
 für  $l_3 := 1, |\mathcal{T}|$   
 speichere (Bewertung( $\gamma_{l_3}$ ), Suche-bewertet-Stufe-4,  $\gamma_{l_3}, \alpha$ )

Funktion Suche-bewertet-Stufe-4 ( $\alpha_1, \alpha_2$ )  
 für  $l_4 := 1, |\mathcal{T}|$   
 falls  $(\{\alpha_1, \gamma_{l_4}\}, \{\nu_{1,2}\}) \in \mathbf{A}$   
 falls  $(\{\alpha_2, \nu_{1,2}\}, \{\nu_{2,1}\}) \in \mathbf{A}$   
 speichere (Bewertung( $\nu_{2,1}$ ), Suche-bewertet-Stufe-5,  $\nu_{2,1}$ )

...

Funktion Suche-bewertet-Stufe- $[|\mathcal{R}| - 1]$  ( $\alpha_1, \alpha_2, \dots, \alpha_{ld(|\mathcal{R}|-1)}$ )  
 für  $l_{|\mathcal{R}|-1} := 1, |\mathcal{T}|$   
 speichere (Bewertung( $\nu_{1,3}$ ), Suche-bewertet-Stufe- $|\mathcal{R}|$ ,  $\gamma_{l_{|\mathcal{R}|-1}}, \alpha_1, \alpha_2, \dots, \alpha_{ld(|\mathcal{R}|-1)}$ )

Funktion Suche-bewertet-Stufe- $|\mathcal{R}|$  ( $\alpha_1, \alpha_2, \dots, \alpha_{ld(|\mathcal{R}|)}$ )  
 für  $l_{|\mathcal{R}|} := 1, |\mathcal{T}|$   
 falls  $(\{\alpha_1, \gamma_{l_{|\mathcal{R}|}\}, \{\nu_{1,ld(|\mathcal{R}|)/2}\}) \in \mathbf{A}$   
 falls  $(\alpha_2, \nu_{1,ld(|\mathcal{R}|)/2}, \{\nu_{2,ld(|\mathcal{R}|)/4}\}) \in \mathbf{A}$   
 ...  
 falls  $(\{\alpha_{ld(|\mathcal{R}|-1)}, \nu_{ld(|\mathcal{R}|),1}, \{\sigma\}) \in \mathbf{A}$   
 ein Analyseergebnis  $\sigma$  wurde gefunden  
 bewerte  $\sigma$

Beginn des Algorithmus  
 für  $l_1 := 1, |\mathcal{T}|$   
   speichere (Bewertung( $\gamma_{l_1}$ ), Suche-bewertet-Stufe-2,  $\gamma_{l_1}$ )  
 solange Speicher nicht leer:  
   führe für die bestbewertete Menge gespeicherter Instanzen die zugehörige Funktion aus

(3 – 39)

Aus der Menge aller gespeicherten offenen Analysezustände (Bewertung, Suchfunktion, Instanzen) kann derjenige zur weiteren Expansion bzw. Analyse ausgewählt werden, der am besten bewertet ist.

In Algorithmus 3-39 werden in jeder Suchschleife alle terminalen Instanzen berücksichtigt. Hier laufen die Indices von 1 bis  $|\mathcal{T}|$ , so dass die Bewertung jeder terminaler Instanz berücksichtigt werden kann. Im Gegensatz dazu laufen in Algorithmus 3-10 die Indices  $l_{2i-1}, i = 1, 2, \dots$  von 1 bis  $|\mathcal{T}| - 1$  und die Indices  $l_{2i}$  von  $l_{2i-1} + 1$  bis  $|\mathcal{T}|$ , so dass jedes Paar von Instanzen nur ein Mal überprüft wird.

Im Abschnitt 4.2 (Seite 92) werden bewertete Analysezustände im Zusammenhang mit Systemen, die auf semantischen Netzen aufbauen, diskutiert.

### 3.3.3 Algorithmen mit bewerteten Zwischenergebnissen

Bei Speicherung der Zwischenergebnisse, die während Analysen entstehen, können diese bewertet und dann bewertungsgesteuert für weitere Analysen ausgewählt werden.

Der Analyseaufwand hängt von der Anzahl der während der Analyse gespeicherten Zwischenergebnisse ab (Gl. 3-20). Diese Anzahl soll durch die Optimierung minimiert werden.

Ein möglicher Algorithmus zur Optimierung mit bewerteten Zwischenergebnissen ist der folgende, aus Algorithmus 3-19 abgeleitete:

$\mathcal{N}_{i,j} := \emptyset; j = 1, 2, \dots, \frac{|\mathcal{R}|}{2^i}; i = 1, 2, \dots, \text{ld}(|\mathcal{R}|) - 1$   
 Funktion Suche-bewertet-T-Stufe-1( $\alpha$ )  
 für  $l_1 := 1, |\mathcal{T}|$   
   mit  $\alpha, \gamma_{l_1} \in \mathcal{T}$   
   falls  $(\{\alpha, \gamma_{l_1}\}, \{\mu_{1,1}\}) \in \mathbf{A}$   
      $\mathcal{N}_{1,1} := \mathcal{N}_{1,1} \cup \{\mu_{1,1}\}$   
     speichere (Bewertung( $\mu_{1,1}$ ), Suche-bewertet-T-Stufe-2-1,  $\mu_{1,1}$ )  
     ...  
   falls  $(\{\alpha, \gamma_{l_1}\}, \{\mu_{1,|\mathcal{R}|/2}\}) \in \mathbf{A}$   
      $\mathcal{N}_{1,|\mathcal{R}|/2} := \mathcal{N}_{1,|\mathcal{R}|/2} \cup \{\mu_{1,|\mathcal{R}|/2}\}$   
     speichere (Bewertung( $\mu_{1,|\mathcal{R}|/2}$ ), Suche-bewertet-T-Stufe-2- $|\mathcal{R}|/2$ ,  $\mu_{1,|\mathcal{R}|/2}$ )

Funktion Suche-bewertet-T-Stufe-2-1( $\alpha$ )  
für  $l_1 := 1, |\mathcal{N}_{1,2}|$   
falls  $(\{\alpha, \mu_{1,2,l_1}\}, \{\mu_{2,1}\}) \in \mathbf{A}$  mit  $\alpha \in \mathcal{N}_{1,1}$  und  $\mu_{1,2,l_1} \in \mathcal{N}_{1,2}$   
 $\mathcal{N}_{2,1} := \mathcal{N}_{2,1} \cup \{\mu_{2,1}\}$   
speichere (Bewertung( $\mu_{2,1}$ ), Suche-bewertet-T-Stufe-3-1,  $\mu_{2,1}$ )  
...  
Funktion Suche-bewertet-T-Stufe-2- $\lfloor |\mathcal{R}|/2 \rfloor$ ( $\alpha$ )  
für  $l_1 := 1, |\mathcal{N}_{1, \lfloor |\mathcal{R}|/2 \rfloor}|$   
falls  $(\{\alpha, \mu_{1, \lfloor |\mathcal{R}|/2 \rfloor, l_1}\}, \{\mu_{2, \lfloor |\mathcal{R}|/4 \rfloor}\}) \in \mathbf{A}$   
mit  $\alpha \in \mathcal{N}_{1, \lfloor |\mathcal{R}|/2 \rfloor - 1}$  und  $\mu_{1, \lfloor |\mathcal{R}|/2 \rfloor, l_1} \in \mathcal{N}_{1, \lfloor |\mathcal{R}|/2 \rfloor}$   
 $\mathcal{N}_{2, \lfloor |\mathcal{R}|/4 \rfloor} := \mathcal{N}_{2, \lfloor |\mathcal{R}|/4 \rfloor} \cup \{\mu_{2, \lfloor |\mathcal{R}|/4 \rfloor}\}$   
speichere (Bewertung( $\mu_{2, \lfloor |\mathcal{R}|/4 \rfloor}$ ), Suche-bewertet-T-Stufe-3- $\lfloor |\mathcal{R}|/4 \rfloor$ ,  $\mu_{2, \lfloor |\mathcal{R}|/4 \rfloor}$ )  
...  
Funktion Suche-bewertet-T-Stufe- $\lfloor \text{id}(|\mathcal{R}|) - 1 \rfloor$ ( $\alpha$ )  
für  $l_1 := 1, |\mathcal{N}_{\text{id}(|\mathcal{R}|) - 1, 2}|$   
falls  $(\{\alpha, \mu_{\text{id}(|\mathcal{R}|) - 1, 2, l_1}\}, \{\sigma\}) \in \mathbf{A}$   
mit  $\alpha \in \mathcal{N}_{\text{id}(|\mathcal{R}|) - 1, 1}$  und  $\mu_{\text{id}(|\mathcal{R}|) - 1, 2, l_1} \in \mathcal{N}_{\text{id}(|\mathcal{R}|) - 1, 2}$   
ein Analyseergebnis  $\sigma$  wurde gefunden

Beginn des Algorithmus

für  $l_1 := 1, |\mathcal{T}|$   
speichere (Bewertung( $\gamma_{l_1}$ ), Suche-bewertet-T-Stufe-1,  $\gamma_{l_1}$ ) (3 – 40)

solange Speicher nicht leer:

führe für die bestbewertete gespeicherte Instanz die zugehörige Funktion aus

Bei Ausfall der Bewertung entartet die Analyse in eine Suche entsprechend Algorithmus 3-19. Dabei wird durch die Speicherung der Zwischenergebnisse erreicht, dass der zu erwartende Aufwand kleiner ist als der bei Ausfall der Bewertung mit Algorithmus 3-39 zu erwartende Aufwand bei der Suche mit bewerteten Analysezuständen (Satz B.99). In diesem Fall entartet die Analyse mit Algorithmus 3-39 in eine Suche entsprechend Algorithmus 3-10.

Im Abschnitt 3.7 wird die Minimierung des Analyseaufwandes bei Einsatz eines Zuordnungsautomaten, der mit bewerteten Zwischenergebnissen arbeitet, erläutert.

### 3.4 Mehrfachgenerierungen

Bei strukturellen Analysen werden Analyseergebnisse in vielen Fällen mehrfach generiert. Zuordnungsautomaten generieren prinzipbedingt mehrfach. Da die Ergebnisse aber nur einfach gespeichert werden, verursacht diese Art der Mehrfachgenerierung keine Mehrfachüberprüfungen. Mehrfach generierte Primitivobjekte und nicht deterministisch generierte Zwischenergebnisse führen dagegen auch zu Mehrfachüberprüfungen und erhöhtem Analyseaufwand.

#### 3.4.1 Mehrfachgenerierungen bei der Analyse mit Zuordnungsautomaten

Bei Zuordnungsautomaten werden komplexe Instanzen  $\beta$  aus ihren Teilen  $\alpha_1, \alpha_2, \dots$  zusammengesetzt. Für jedes Teil  $\alpha_i$  wird ein Überprüfungsauftrag

(Gl. B-83, B-127) generiert, so dass jede komplexe Instanz  $\beta$  mehrfach überprüft wird. Bei den hier betrachteten Produktionen  $(\mathcal{A}, \mathcal{B})$  bzw. Assoziationen  $(\mathcal{X}, \mathcal{Y})$  mit  $|\mathcal{A}| = |\{\alpha_1, \alpha_2\}| = 2$  bzw.  $|\mathcal{X}| = 2$  werden komplexe Instanzen aus zwei Teilen zusammengesetzt, so dass jede Überprüfung doppelt erfolgt. Um zu verhindern, dass deswegen identische Zwischenergebnisse doppelt gespeichert werden und nachfolgende Überprüfungen mehrfach erfolgen, muss vor der Abspeicherung jedes Analyseergebnisses überprüft werden, ob das neue Ergebnis bereits vorliegt. Abgespeichert wird nur, wenn das Ergebnis nicht vorliegt, also genau ein Mal. Im Abschnitt 3.4.3 wird dies weiter ausgeführt.

### 3.4.2 Mehrfachgenerierung von Primitivobjekten

Aus der Statistik der Szene bzw. der Eingangsdaten kann nur eine untere Grenze für die Anzahl der generierten Zwischenergebnisse abgeleitet werden. Dass die Anzahl selber nicht abgeleitet werden kann, liegt (i) daran, dass die Extraktionsprogramme für ein zu ermittelndes Ereignis aus den zu analysierenden Sensordaten mehrere redundante Primitivobjekte extrahieren können. Ein Beispiel hierfür sind Objektgrenzen in Grauwertbildern, die durch mehrere Geradenstücke beschrieben werden (vgl. Abbildung 2-12a und f). Wenn eine Objektfläche aus mehreren begrenzenden Abschnitten, die jeweils mehrfach vorliegen, aufgebaut wird, so entsteht für jede umschließende Kombination der Geradenstücke eine Objektfläche. Bei z. B. vier Randabschnitten, die jeweils dreifach vorliegen, gibt es  $3^4 = 81$  leicht unterschiedlich beschriebene Objektflächen, die alle als Zwischenergebnisse generiert werden. Wenn daraus weitere Objekte aufgebaut werden, pflanzt sich diese Mehrfachgenerierung entsprechend fort.

Dieses Problem tritt (ii) in verschärfter Form auf, wenn in der Szene vorliegende Objekte mehrfach in unterschiedlichen Sensordatensätzen vorliegen und die Analyse alle Datensätze parallel auswertet. Für jeden Datensatz werden Vorverarbeitungs- und Extraktionsprogramme ausgeführt, die in der Praxis Primitivobjekte von identischen Ereignissen der Szene in unterschiedlichen Datensätzen mit leicht variierenden Merkmalen generieren. Dieses Problem entsteht selbst dann, wenn die Beschreibung der Datensätze (z. B. die Sensororientierung) fehlerfrei ist.

Wenn (iii) die Beschreibung der parallel analysierten Datensätze mit Fehlern behaftet ist, werden die Unterschiede der Ergebnisse, die aus verschiedenen Datensätzen stammen, noch größer.

Deshalb können bei der strukturellen Analyse für identische Ereignisse mehrere Ergebnisse mit variierenden Merkmalen entstehen, deren Anzahl z. B. mit zusätzlichen *Ballungsanalysen* reduziert werden kann. Im Ausblick (Abschnitt 5.3) wird die Ballungsanalyse aufgegriffen.

### 3.4.3 Mehrfachgenerierung von Zwischenergebnissen

Zusätzlich zur Statistik der Eingabedaten, die darüber entscheidet, wieviele Zwischenergebnisse mindestens generiert werden, ist bei der Analyse mit Produktionsnetzen zu betrachten, wie oft ein Zwischenergebnis redundant generiert wird.

#### Mehrfachgenerierungen mit Ableitungsketten

Analysen, die Ableitungsketten erzeugen, beginnen ohne zusätzliche Maßnahmen mit jeder Instanz einer Ableitungskette, die sowohl Anfang als auch Mittelteil einer Ableitungskette sein kann, auch eine neue Ableitungskette. Deshalb wird in diesem Fall massiv mehrfach analysiert.

#### Mehrfachgenerierungen mit Ableitungsbäumen

Bei Ableitungsbäumen mit Speicherung der Zwischenergebnisse können ebenfalls Mehrfachgenerierungen entstehen. Abbildung 3-5 zeigt zwei Vorgehensweisen. Nicht deterministisch formuliert werden z. B. nur lageunabhängige Zusammenhänge modelliert. Dabei entstehen Mehrfachergebnisse. Um deterministisch zu analysieren, müssen weitere einschränkende Bedingungen hinzugenommen werden, die über die notwendigen Zusammenhänge hinausgehen.

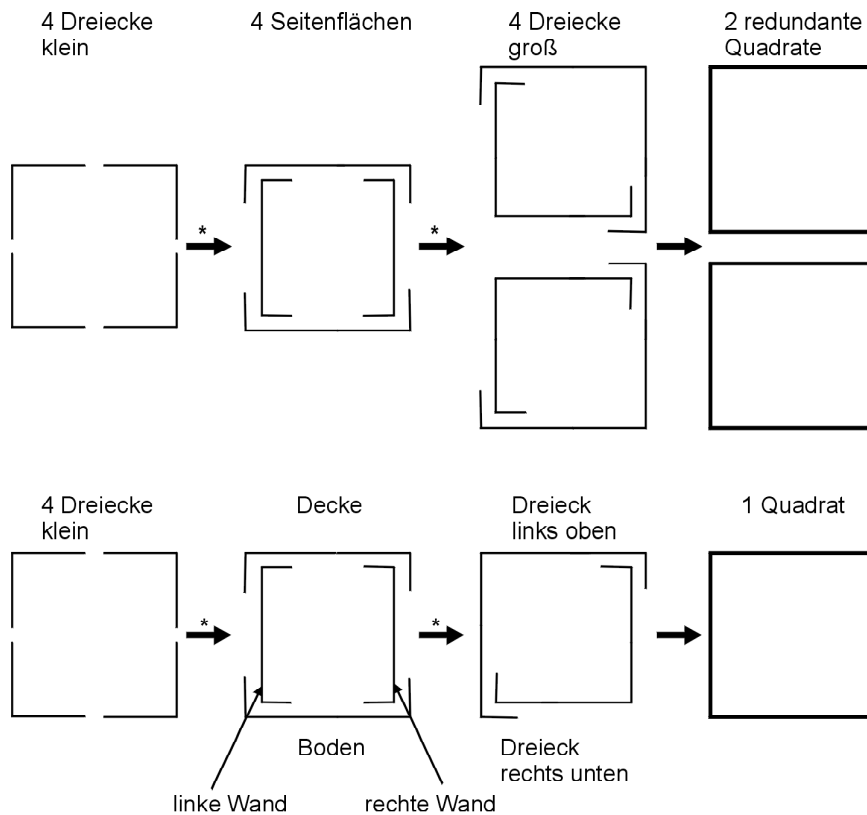
Das Beispiel zeigt den Nachteil der deterministischen Produktionen, bei denen aus den relativ einfachen Zusammenhängen der nicht deterministischen Produktionen umfangreiche Sonderfälle abgeleitet werden müssen, die dann als Produktionen zu formulieren sind. Aufgabe des Entwicklers eines Analyseprogramms ist, einen Kompromiss zwischen der Komplexität der Produktionsnetze und dem zusätzlichen durch Mehrfachgenerierungen verursachten Analyseaufwand zu finden.

Die bei Zuordnungsautomaten nicht vermeidbare Mehrfachgenerierung aller Ergebnisse (vgl. Seite 73) ist dagegen unkritisch, da die  $n$  Überprüfungen immer die gleiche Konfiguration aus  $n$  Instanzen bearbeiten und bei korrekter Berechnung  $n$  Mal identische Ergebnisse entstehen, die als identisch erkannt und deshalb nur ein Mal abgespeichert werden. Nicht deterministische Produktionen generieren dagegen Ergebnisse mehrfach über verschiedene Zwischenergebnisse und mehrere Analysestufen. Da die Merkmale dieser konkurrierenden Zwischenergebnisse bei gestörten Daten in der Regel unterschiedlich approximiert werden, entstehen unterschiedliche Merkmale für identische Ereignisse, die aufgrund der abweichenden Merkmale mehrfach abgespeichert werden.

## 3.5 Begrenzung des Analyseaufwandes

Die Erfahrung zeigt, dass Sensordaten Qualitätsmerkmale zugeordnet werden können, von denen der Analyseaufwand abhängt.

Zu diesen Qualitätsmerkmalen gehören beispielsweise:



**Abb. 3-5:** Rechteck aus orientierten (unten) bzw. nicht orientierten (oben) Teilobjekten

1. *Aufbau der Szene:* Die Auswahl und Anordnung der Szenenobjekte und der Sensoren bestimmen u. a. Objektansichten sowie -verdeckungen und verhindern in ungünstigen Fällen die Analyse einzelner Objekte.
2. *Geometrische Auflösung:* Die geometrische Auflösung ist durch die verfügbare Sensorik und die Aufnahmeparameter (z. B. Entfernung zwischen Sensor und Objekt) definiert und in ungünstigen Fällen nicht für die Analyseaufgabe geeignet.
3. *Die in den Sensordaten feststellbaren Unterschiede an Objektgrenzen, mit denen die Grenzen von Objekten oder Objektteilen bestimmt werden (radiometrische Auflösung):* In nicht kontrollierter Umgebung können Schattenbereiche, schlechte Beleuchtungsverhältnisse oder Einflüsse der Atmosphäre (z. B. Regen, Schneefall, geringer Kontrast bei großer Entfernung) und der Beleuchtung (z. B. Blendung durch direkten Sonneneinfall) dazu führen, dass Objekte mit dem Hintergrund verschmelzen und vom Menschen wahrgenommene scheinbare Objektgrenzen in den Sensordaten objektiv nicht vorliegen. Abbildung 3-6 /Sti91/ zeigt als Beispiel im Schatten liegende, auf den ersten Blick vom Menschen "erkannte", messtechnisch aber nicht erfassbare

Dachbegrenzungen<sup>8</sup>.

4. *A priori Eigenschaften*: Künstliche Objekte haben in vielen Fällen charakteristische Eigenschaften und stehen z. B. oft senkrecht auf der Erdoberfläche. Mit geringem Schwierigkeitsgrad zu analysieren hieße in diesem Beispiel, zuerst nur die senkrechten Objektteile zu analysieren. Danach würden dann zunehmend auch Objekte mit größeren Abweichungen von der Senkrechten analysiert.
5. *Eigenschaften, die mit vorausgehenden Analysen bestimmt wurden*: Die Analyseergebnisse nacheinander ausgeführter Analysen korrelierter Bildinhalte sind in der Regel ebenfalls korreliert. Beispielsweise kann in Bildfolgen aus Ergebnissen von Vorgängerbildern auf Ergebnisse in Nachfolgebildern geschlossen werden.
6. *Die Güte der aus den Sensordaten extrahierten Primitivobjekte und daraus aufgebauter Zwischenergebnisse*: Gut ausgeleuchtete nicht verdeckte Objektteile, die sich deutlich vom Hintergrund abheben, werden in der Regel von den Programmen zur Extraktion der Primitivobjekte aus den Sensordaten präzise beschrieben. Diese terminalen Instanzen passen dann genau zu den Konzepten der Produktionen. Ihre Analyse sollte mit relativ geringem Aufwand möglich sein, so dass den "ungestörten" Instanzen, die gut mit den Konzepten des Produktionsnetzes übereinstimmen, geringer Schwierigkeitsgrad zuzuweisen ist und sie zuerst analysiert werden sollten.

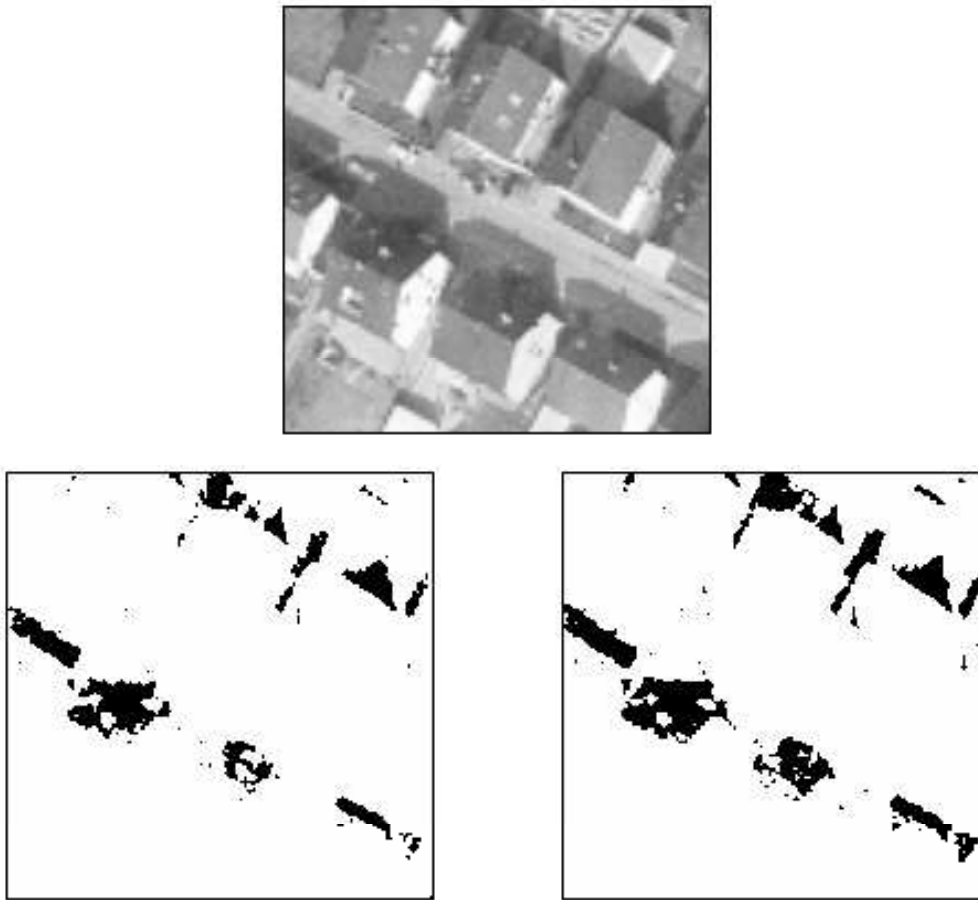
In kontrollierter Umgebung können diese Größen in vielen Fällen z. B. durch entsprechende Beleuchtung sowie Auslegung der zu analysierenden Objekte und der Sensorik optimiert werden, so dass mit einer relativ geringen Anzahl von Zwischenergebnissen analysiert werden kann. In unkontrollierter Umgebung treten regelmäßig die oben angeführten Probleme auf. Diese führen dazu, dass entsprechend des Ausmaßes der qualitätsmindernden Einflüsse für eine erfolgreiche Gesamtanalyse sehr viele Instanzen bzw. Zwischenergebnisse generiert werden müssen oder sogar keine Ergebnisse gefunden werden können.

Um den Analyseaufwand, der in der Regel proportional zur Anzahl der Zwischenergebnisse ist, abzuschätzen und Obergrenzen für die Anzahl der notwendigen Zwischenergebnisse anzugeben, kann versucht werden, die Anzahl der zur Analyse benötigten Zwischenergebnisse als Funktion eines Schwierigkeitsgrades, der u. a. von den oben aufgeführten Qualitätsmerkmalen abhängt, zu lernen. Während der Analyse kann dann z. B. bei Algorithmus 3-19 bzw. Algorithmus 3-40 auf jeder Analysestufe die Anzahl der weiter zu analysierenden Zwischenergebnisse entsprechend der gelernten Obergrenzen begrenzt werden, so dass ein maximaler Gesamtaufwand nicht überschritten wird. Dabei wird gegebenenfalls ein Teil der gespeicherten, relativ schlecht bewerteten Zwischenergebnisse nicht weiter bearbeitet. In der Signalanalyse ist dieser CFAR-Ansatz<sup>9</sup> /Hol95/ üblich. Im nächsten Abschnitt wird der CFAR-Ansatz, der die Zwischenergebnisse in die Menge der weiter zu be-

---

<sup>8</sup>In der unteren Hausreihe ist die obere Dachbegrenzung des mittleren Hauses messtechnisch nicht zu erfassen.

<sup>9</sup>CFAR: Constant False Alarm Rate



**Abb. 3-6:** Durch Schatten nicht mehr detektierbare Dachbegrenzung (/Sti91/  
 Mitte: Originalbild  
 links: Binärbild mit einer Helligkeits-Schwelle oberhalb des Dachgrauwertes  
 rechts: Binärbild mit einer Helligkeits-Schwelle unterhalb des Dachgrauwertes

arbeitenden und die Menge der zu ignorierenden Zwischenergebnisse aufteilt, verfeinert.

### 3.6 Minimierung des Analyseaufwandes bei Zuordnungssystemen

Bewertungen sind Grundlage für Verfahren zur Minimierung des Analyseaufwandes. Zur Definition des Optimalitätskriteriums wird angenommen, dass zu analysierenden Daten (d. h. Mengen terminaler Instanzen) zusammen mit den zu bestimmenden Ergebnissen (d. h. Startsymbolen) *Schwierigkeitsgrade* zugeordnet werden können und dass die Anzahl der Zwischenergebnisse, die bei der Analyse gespeichert und weiter analysiert werden muss, monoton mit dem Schwierigkeitsgrad ansteigt.



Entsprechend den zu analysierenden Daten und den zu bestimmenden Analyseergebnissen müssen konkrete Größen für den Schwierigkeitsgrad gefunden und durch entsprechende Versuche die Anzahl der jeweils zu analysierenden Zwischenergebnisse bestimmt werden. Beispiele für solche den Schwierigkeitsgrad beeinflussende bzw. definierende Größen sind auf Seite 75 aufgelistet.

Die Analyse soll mit geringstem Schwierigkeitsgrad beginnen, und der Schwierigkeitsgrad soll erhöht werden, sobald alle Instanzen für den aktuellen Schwierigkeitsgrad analysiert wurden. Dazu werden den Schwierigkeitsgraden  $i$  jeweils Teilmengen  $\mathbf{A}_r^i$  der Menge der Assoziationen  $\mathbf{A}$  zugeordnet, mit  $\mathbf{A}_r^i \subseteq \mathbf{A}_r^j; i, j \in \mathbb{N}$  für  $i < j$ . Im ersten Schritt werden für den geringsten Schwierigkeitsgrad 1 mit  $\mathbf{A}_r^1$  alle Ergebnisse bestimmt, so dass der zu analysierenden Konfiguration  $\mathcal{T}$  das erste Ergebnis  $AF_1(\mathcal{T})$  abschließend zugeordnet wird [ $\mathcal{T} \xrightarrow{\dagger(\mathcal{I}, \mathbf{A}_r^1)} AF_1(\mathcal{T})$ , Gl. B-5, B-15]. Danach wird der Schwierigkeitsgrad erhöht. Mit dem erhöhten Schwierigkeitsgrad werden dann zusätzliche terminale und nichtterminale Instanzen analysiert, und es wird ein weiteres Mal abschließend zugeordnet [ $AF_1(\mathcal{T}) \xrightarrow{\dagger(\mathcal{I}, \mathbf{A}_r^2)} AF_2(\mathcal{T})$  bzw. mit  $\mathbf{A}_r^1 \subseteq \mathbf{A}_r^2$ , Lemma B.13 und Satz B.12  $\mathcal{T} \xrightarrow{\dagger(\mathcal{I}, \mathbf{A}_r^2)} AF_2(\mathcal{T})$ ]. Wird für eine Analyse ein zu geringer Schwierigkeitsgrad  $i$  angenommen, dann entstehen keine Endergebnisse (d. h. es werden keine Startsymbole zugeordnet:  $\forall \sigma \in \Sigma : \sigma \notin AF_i(\mathcal{T})$ ). Trotzdem liegen die für diesen Schwierigkeitsgrad zugeordneten Zwischenergebnisse  $AF_i(\mathcal{T}) \setminus \mathcal{T}$  vor und können für weitere Analysen verwendet werden. Wird nach einer erfolglosen Analyse der Schwierigkeitsgrad erhöht, dann werden alle Zwischenergebnisse, die für den vorher bearbeiteten niedrigeren Schwierigkeitsgrad erzeugt wurden, auch für den erhöhten Schwierigkeitsgrad benötigt (Satz B.14), so dass keine unnötigen Zwischenergebnisse entstehen. Da zusätzliche Überprüfungen nur für zusätzliche Zwischenergebnisse durchgeführt werden, werden auch keine unnötigen Überprüfungen ausgeführt.

Diese Auswahlstrategie bricht eine Analyse endgültig erst ab, wenn alle möglichen Ergebnisse (Zwischen- und Endergebnisse) bestimmt wurden. Mit dieser Strategie werden schnell erste Ergebnisse generiert und danach sukzessiv immer aufwändiger zu analysierende Ergebnisse, mit denen die ersten Ergebnisse ergänzt, präzisiert oder auch korrigiert werden. Der Ansatz gehört damit zur Klasse der *jederzeit* oder *anytime* Ansätze /Ben96/.

Die Teilmengen  $\mathbf{A}_r^i \subseteq \mathbf{A}$  können mit bewerteten Teilmengen  $\mathcal{I}_r^i$  der Menge aller Instanzen  $\mathcal{I}$  mit  $\mathcal{I}_r^i \subseteq \mathcal{I}_r^j; i, j \in \mathbb{N}$  für  $i < j$  definiert werden. Mit  $\mathbf{A} \subseteq \wp(\mathcal{I}) \times \wp(\mathcal{I})$  können z. B. die anschließend aufgeführten reduzierten Assoziationenmengen  $\mathbf{A}_{r_1}^i, i = 1, 2, 3$  definiert werden.

$\mathbf{A}_{r_1}^i$  überprüft und generiert nur Instanzen der entsprechenden Bewertung:

$$\mathbf{A}_{r_1}^i \subseteq \wp(\mathcal{I}_r^i) \times \wp(\mathcal{I}_r^i); i \in \mathbb{N} \quad (3 - 41)$$

$\mathbf{A}_{r_2}^i$  schränkt die Überprüfung auf entsprechend bewertete Instanzen ein:

$$\mathbf{A}_{r_2}^i \subseteq \wp(\mathcal{I}_r^i) \times \wp(\mathcal{I}); i \in \mathbb{N} \quad (3 - 42)$$

Assoziationen von  $\mathbf{A}_{r_3}^i$  überprüfen, sobald eine zu überprüfende Instanz entsprechend bewertet ist:

$$\mathbf{A}_{r_3}^i \subseteq \left\{ (\mathcal{X}, \mathcal{Y}) \left| \begin{array}{l} \mathcal{X}, \mathcal{Y} \subseteq \mathcal{I} \\ \exists \gamma \in \mathcal{I}_r^i : \\ \gamma \in \mathcal{X} \end{array} \right. \right\}; i \in \mathbb{N} \quad (3-43)$$

$\mathbf{A}_{r_1}^i$  und  $\mathbf{A}_{r_2}^i$  werden nicht weiter betrachtet.  $\mathbf{A}_{r_3}^i$  kann zur Minimierung des Analyseaufwandes mit Zuordnungsautomaten verwendet werden.

### 3.7 Minimierung des Analyseaufwandes bei Zuordnungsautomaten

Die Auswahlfunktion  $a$  (Gl. B-84) eines Zuordnungsautomaten  $((\mathcal{I}, \mathbf{P}, i), a)$  wählt die bei einem Zustandsübergang zu überprüfenden Überprüfungsaufträge aus und steuert so die Folge der Zustandsübergänge während der Analyse (Gl. B-86 bis Gl. B-88). Optimierung der Analyse heißt, die Reihenfolge der Analyseschritte so festzulegen, dass die Endergebnisse der Analyse (d.h. Startsymbole) mit möglichst wenigen Zustandsübergängen zugeordnet werden und danach nur noch zur Absicherung der Ergebnisse Zwischenergebnisse entstehen, aus denen keine Startsymbole mehr aufgebaut werden können.

Der Zustand, in dem das letzte Startsymbol  $\sigma \in \Sigma$  zur Analyse einer Konfiguration  $\mathcal{T}$  zugeordnet wurde, sei  $(Q, U_q)$ , d.h.,  $\Sigma \cap AF(\mathcal{T}) \subseteq Q$ . In diesen Zustand wurde ausgehend von  $(\mathcal{T}, u(\mathcal{T}))$  übergegangen:  $(\mathcal{T}, u(\mathcal{T})) \stackrel{*A}{=} (Q, U_q)$ . Der Automat hält an, wenn abschließend in einen Zustand ohne Überprüfungsaufträge übergegangen wird:  $(\mathcal{T}, u(\mathcal{T})) \stackrel{\dagger A}{=} (AF(\mathcal{T}), \emptyset)$ . Dabei wird die Analysefunktion  $AF(\mathcal{T})$  abschließend zugeordnet (Korollar B.63). Da die Zustandsübergänge konvergent durchgeführt werden (Korollar B.64), erhält man  $(\mathcal{T}, u(\mathcal{T})) \stackrel{*A}{=} (Q, U_q) \stackrel{\dagger A}{=} (AF(\mathcal{T}), \emptyset)$ . Die Konfiguration  $Q$  (nicht aber  $AF(\mathcal{T})$ ) hängt von der Auswahlfunktion  $a$  ab. Zur Optimierung des Zustandsüberganges  $(\mathcal{T}, u(\mathcal{T})) \stackrel{*A}{=} (Q, U_q)$  ist die Auswahlfunktion  $a$  so einzurichten, dass die Anzahl der direkten Zuordnungen  $k$  der Zuordnung  $\mathcal{T} \stackrel{(k)Z}{=} Q$ , die zum Zustandsübergang  $(\mathcal{T}, u(\mathcal{T})) \stackrel{*A}{=} (Q, U_q)$  gehört (Satz B.49), minimiert wird<sup>10</sup>. Die danach folgenden Zustandsübergänge mit der abschließenden Zuordnung  $Q \stackrel{\dagger Z}{=} AF(\mathcal{T})$  wirken sich nach Voraussetzung auf die erkannten Startsymbole nicht mehr aus.

Für diesen Ablauf kann die Auswahlfunktion  $a$  so definiert werden, dass zuerst nur Überprüfungsaufträge  $(\gamma, (\mathcal{A}, \mathcal{B}))$  (Gl. B-83) ausgewählt werden, bei denen die triggernde Instanz  $\gamma$  höchste Bewertung hat ( $\gamma \in \mathcal{I}_r^1$ ). Danach werden Überprüfungsaufträge mit schlechter bewerteten triggernden Instanzen bearbeitet. Das heißt, dass der Zuordnungsautomat zuerst nur Assoziationen  $(\mathcal{A}, \mathcal{B}) \in \mathbf{A}_{r_3}^1$  (Gl. 3-43) ausführt und dann zu Überprüfungsaufträgen bzw. Assoziationen übergeht, die

<sup>10</sup>Dies gilt bei Einsatz eines Einzelprozessors (Gl. B-86). Bei Einsatz mehrerer Prozessoren (Gl. B-87, Gl. B-88) sind zusätzliche analoge Kriterien zu definieren.

auch schlechter bewertete triggernde Instanzen berücksichtigen. In der zweiten Phase werden Assoziationen  $(\mathcal{A}, \mathcal{B}) \in \mathbf{A}_{r_3}^1 \cup \mathbf{A}_{r_3}^2$ , in der  $j$ -ten Phase Assoziationen  $(\mathcal{A}, \mathcal{B}) \in \bigcup_{i=1}^j \mathbf{A}_{r_3}^i, j \in \mathbb{N}$  überprüft. Wie in Abschnitt 3.6 erläutert, entstehen dabei keine unnötigen Zwischenergebnisse.

Es ist also möglich, mit Zuordnungsautomaten Bewertungen (Schwierigkeitsgrade) für eine Reduzierung des Analyseaufwandes einzusetzen. Die Untersuchungen hierzu sind noch nicht abgeschlossen. Arbeitshypothese ist, dass mit entsprechend definierten bewerteten Instanzenmengen  $\mathcal{I}_r^i, i \in \mathbb{N}$  sowohl a priori Information (z. B. erwartete Eigenschaften wie *Objekte stehen senkrecht*) als auch während der Analyse erkannte Zusammenhänge (z. B. Objektpositionen in bereits analysierten Bildern einer Bildfolge für Erwartungen in nachfolgenden Bildern) für eine Optimierung der Verarbeitungsreihenfolge genutzt werden können (siehe auch Abschnitt 5.3).

## 4 Vergleich mit anderen Ansätzen und Systemen

Die Bildanalyzesysteme der späten 80-er Jahre wie Sigma /Mat90/ und andere sind in vieler Hinsicht vergleichbar mit dem hier vorgestellten. Sie werden z. B. in der Einleitung von /Cae97/ (Seite 2 ff.) diskutiert. Objekte werden durch Teile und ihre Relationen definiert. Dabei werden die Teile in vielen Fällen durch n-Tupel beschrieben.

Im Unterschied zu den Systemen der 80-er Jahre ist der in dieser Arbeit vorgestellte Ansatz detailliert formal als Automat beschrieben und für den Einsatz mit Parallelrechnern sowie für explorative Analysen erweitert. Außerdem werden Aussagen zur Analyse mit Zuordnungsautomaten bewiesen, so dass die Analyseergebnisse nachvollziehbar und reproduzierbar sind.

Ausgangspunkte des in dieser Arbeit entwickelten Ansatzes sind der PANDEMONIUM-Ansatz und tafelbasierte Systeme zur strukturellen Bildanalyse (vgl. Abschnitt 1.3). Die entwickelte formale Darstellung orientiert sich an linguistischen bzw. syntaktischen Ansätzen.

### 4.1 Syntaktische Ansätze

Die Theorie formaler Sprachen ist Grundlage der linguistischen bzw. syntaktischen Bildanalyse und war Motivation für die Formulierung der formalen Beschreibung der strukturellen Musteranalyse mit Produktionsnetzen.

#### 4.1.1 Vergleich mit syntaktischen Bildanalyseverfahren

Syntaktische Bildanalyse-Verfahren bauen unmittelbar auf Verfahren zur Analyse von Zeichenketten und deren abgesicherten formalen Basis auf. Dazu werden Zeichenketten mit  $\xrightarrow[STS]{(V,P)}$  manipuliert (Gl. 2-6), was bei entsprechender Interpretation der Zeichen von  $V$  zu Mustergenerierungen und -analysen führt. Transformationssysteme  $\left( W^*(V), \xrightarrow[STS]{(V,P)} \right)$  für Semi-Thue-Systeme (Gl. 2-6) definieren die syntaktischen Verfahren. Beispielsweise sind die Mengen der Produktionen  $P$  der anschließend erläuterten Chromosomen-Grammatiken  $(V_T, V_N, \Sigma, P)$  Semi-Thue-Systeme (Gl. 2-5), mit denen Bildanalyse-Schritte über Veränderungen von Zeichenketten spezifiziert werden. Einen Überblick über dieses Gebiet geben z. B. die Bücher *Syntactic Pattern Recognition* /Gon78/, *Syntactic Pattern Recognition, Applications* /Fu77/, *Syntactic and Structural Pattern Recognition - Theory and Applications* - /Bun90/ und *Visual Language Theory* /Mar98/.

Die formal abgesicherte Basis der Verfahren ist ihr großer Vorteil und gestattet, eingeführte Verfahren zur Analyse von Zeichenketten für die Bildanalyse zu nutzen.

Nachteil der auf Semi-Thue-Systemen aufbauenden syntaktischen Bildanalyseverfahren ist, dass sich die zweidimensionale Bildanalyse nur unvollkommen auf

eindimensionale Manipulationen von Zeichenketten abbilden lässt und deshalb syntaktische Einschränkungen bei der Formulierung der Analysevorschriften, die für komplexe Probleme nicht zu akzeptieren sind, hingenommen werden müssen. Dies gilt im verstärkten Maße, wenn die Grammatik zur Zeichenkettenveränderung aus Effizienzgründen kontextfrei /Sal78/ formuliert wird.

Die anschließend erläuterte Chromosomen-Grammatik gibt ein Beispiel für syntaktische Bildanalyseverfahren und die damit verbundene Einschränkung der Analysefähigkeit.

### Chromosomen-Grammatiken als Beispiel für syntaktische Verfahren

Die Chromosomen-Grammatik baut unmittelbar auf der Zeichenkettenverarbeitung auf und wird mit folgendem Beispiel, das sich an ein entsprechendes Beispiel in /Gon78/ anlehnt, erläutert. Dazu werden terminale Symbole  $V_T$ , nichtterminale Symbole  $V_N$  und eine Menge der Produktionen  $P$  angegeben. In den Produktionen sind nur die in Klammern angegebenen Abkürzungen der Symbole von  $V_T$  und  $V_N$  eingetragen.

$$\begin{aligned} V_T := \{ & \text{Halbbogen mit kleinem Radius im Uhrzeigersinn (a),} \\ & \text{näherungsweise Geradenstück (b),} \\ & \text{Halbbogen mit kleinem Radius im Gegenuhrzeigersinn (c),} \\ & \text{Ausbuchtung (d), Teilbogen mit großem Radius im Uhrzeigersinn (e)} \} \end{aligned} \quad (4-1)$$

$$\begin{aligned} V_N := \{ & \text{Submedian (S1), Telecentric (S2),} \\ & \text{Armpaar (A), Unterteil (B), Seitenteil (C), Arm (D),} \\ & \text{rechtes Teil (E), linkes Teil (F)} \} \end{aligned} \quad (4-2)$$

$$\begin{aligned} P := \{ & (S1, AA), (A, CA), (A, DE), (B, bB), (B, e), (C, Cb), (D, Db), (E, cD), \\ & (S2, BA), (A, AC), (A, FD), (B, Bb), (C, bC), (C, b), (D, bD), (D, a), (F, Dc) \} \end{aligned} \quad (4-3)$$

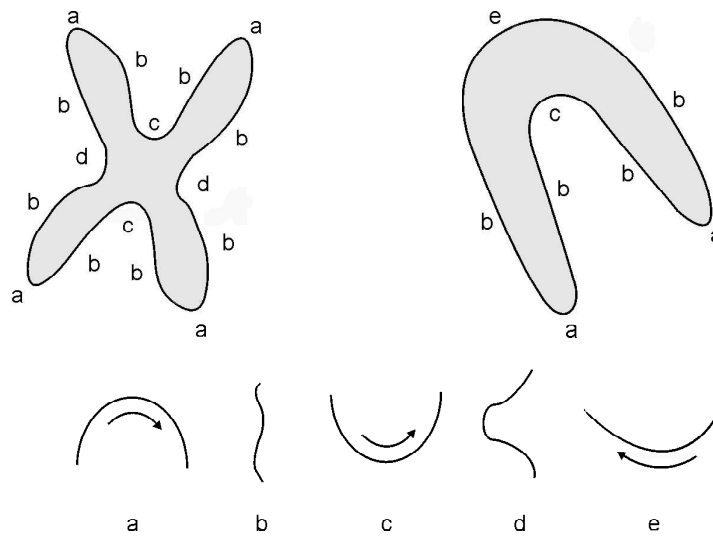
Die Produktionen sind ein Semi-Thue-System. Das dazugehörige Transformationssystem  $\left( W^*(V_T \cup V_N), \frac{(V,P)}{STS} \right)$  definiert die Generierung bzw. mit invertierten Produktionen die Analyse der Chromosomen.

Nimmt man noch die Menge der Startsymbole  $\Sigma = \{S1, S2\}$  hinzu, so erhält man eine Grammatik  $(V_T, V_N, \Sigma, P)$  zur Generierung bzw. Analyse von Chromosomen (vgl. Abschnitt 2.3.1).

Abbildung 4-1, die sich an Abbildung 1.14 bzw. Abbildung 2.5 und Abbildung 2.6 aus /Gon78/ anlehnt, zeigt mögliche Ergebnisse sowie die terminalen Symbole. Weitere Beispiele zu Chromosomen-Grammatiken gibt z. B. /Fu77/.

#### 4.1.2 Vergleich mit erweiterten syntaktischen Bildanalyseverfahren

Syntaktische Verfahren definieren die Mustergenerierung und -analyse über Semi-Thue-Systeme mit ihrer Zeichenkettenersetzung. Dieser Ansatz wird



**Abb. 4-1:** Ergebnisse der Analyse mit einer Chromosomen-Grammatik  
 links: Submedian Chromosom definiert durch  
 die Zeichenkette *abcbabdbabcabdb*  
 rechts: Telocentric Chromosom definiert durch  
 die Zeichenkette *ebabcab*  
 unten: terminale Symbole

mit mehrfachverketteten Zeichenketten, Bäumen, Graphen und NAPE-Einheiten /Gon78/ erweitert. Koordinatengrammatiken /Mil72, Sha69, Mich96, Mich98a/ sind eine weitere Verallgemeinerung syntaktischer Verfahren.

Vorteil der erweiterten syntaktischen Verfahren ist, dass durch den Einsatz mehrdimensionaler Strukturen eine größere Flexibilität erreicht wird als bei Ansätzen, die unmittelbar auf der Zeichenkettenanalyse aufbauen.

Nachteil ist, dass keines dieser Verfahren mit Mengen von Produktionen arbeitet, die Semi-Thue-Systeme sind, so dass die abgesicherte Basis der formalen Sprachen verloren geht. Dies wird mit den anschließend erläuterten PDL-Verfahren gezeigt, die die Transformationsrelation  $\xrightarrow[STS]{(V,P)}$  (Gl. 2-6) nicht mehr verwenden, so dass eine neue Transformationsrelation  $\xrightarrow[PDL]{\implies}$  definiert werden müsste und für die Zeichenkettenverarbeitung entwickelte Verfahren nicht mehr eingesetzt werden können. Ein Beispiel hierzu ist die PDL-Zeichenkette  $((\sim(b+a)) + c)$ , die der Zeichenkette *abc* entspricht. Zur Manipulation von *abc* geeignete Produktionen können offensichtlich  $((\sim(b+a)) + c)$  nicht analysieren. Gleichzeitig sind auch bei mehrdimensionalen Grundmustern nach wie vor starke syntaktische Einschränkungen bei der Formulierung der Bildanalyse hinzunehmen.

### Picture description language (PDL) als Beispiel für erweiterte syntaktische Verfahren

Ein Beispiel für erweiterte syntaktische Verfahren, das sich an entsprechende Beispiele in /Gon78/ und /Fu77/ anlehnt, ist die zweidimensionale Bildbeschreibungssprache (PDL) /Sha70/, bei der die Konkatenation verallgemeinert wird /Bun90, Mar98/. Statt wie bei Zeichenketten die einzelnen Zeichen einfach hintereinander zu schreiben, werden Zeichen mit den Symbolen  $+$ ,  $\times$ ,  $-$ ,  $*$  verkettet. Durch Klammerung wird die Reihenfolge der Verkettungen eindeutig spezifiziert. Das Zeichen  $\sim$  dreht ein Symbol um, und mit  $/$ , das vor ein Zeichen geschrieben werden kann, wird angezeigt, dass das Zeichen in einem Ausdruck mehrfach vorkommt und mit mehreren anderen Zeichen verkettet wird. Um mehrfachverkettete Zeichen in einem Ausdruck eindeutig zu identifizieren, werden sie mit hochgestellten Indizes markiert. Diese Indizes sind Elemente der Menge  $V_T^l$  (label designator),  $V_T^a$  sind die anwendungsspezifischen terminalen Zeichen, und  $V_N^a$  sind die anwendungsspezifischen nichtterminalen Zeichen.

Für die Menge der terminalen Zeichen  $V_T$  gilt mit  $V_T^x$ :

$$\begin{aligned} V_T^x &:= \left\{ t^l \mid \begin{array}{l} t \in V_T^a \\ l \in V_T^l \end{array} \right\} \\ V_T &:= V_T^a \cup V_T^x \cup \{ /s \mid s \in V_T^x \} \end{aligned} \quad (4-4)$$

Für die Menge der nichtterminalen Zeichen  $V_N$  gilt analog:

$$\begin{aligned} V_N^x &:= \left\{ n^l \mid \begin{array}{l} n \in V_N^a \\ l \in V_T^l \end{array} \right\} \\ V_N &:= V_N^a \cup V_N^x \cup \{ /m \mid m \in V_N^x \} \end{aligned} \quad (4-5)$$

Die Syntax der Produktionen der Menge  $P_{PDL}$  muss der oben angegebenen Form entsprechen. Mit der Menge der zulässigen Ausdrücke  $V_A$  wird  $P_{PDL}$  definiert.

$$\begin{aligned} V_A^0 &:= V_T \cup V_N \\ V_U^i &:= \bigcup_{j \in \{0,1,\dots,i\}} V_A^j \\ V_A^{i+1} &:= V_U^i \cup \left\{ (x \bullet y) \mid \begin{array}{l} x, y \in V_U^i \\ \bullet \in \{+, \times, -, *\} \end{array} \right\} \cup \{ (\sim x) \mid x \in V_A^i \} \\ V_A &:= \bigcup_{i \in \mathbb{N}_0} V_A^i \end{aligned} \quad (4-6)$$

$$P_{PDL} \subseteq \left\{ (n, a) \mid \begin{array}{l} n \in V_N \\ a \in V_A \end{array} \right\} \quad (4-7)$$

Jedes terminale und nichtterminale Zeichen sowie jeder Ausdruck  $(a \bullet b)$ ,  $\bullet \in \{+, \times, -, *\}$  bzw.  $(\sim a)$  wird mit einem Kopf ( $head(x)$ ) und einem Ende ( $tail(x)$ ) versehen. Deshalb werden die graphischen Repräsentanten von PDL-Zeichen an geeigneter Stelle mit einer den Kopf kennzeichnenden Pfeilspitze markiert.  $a \times b$  hat

zwei heads und  $a - b$  hat zwei tails als Verbindungspunkte (Abbildung 4-2). Dabei gilt die Regel, dass der tail der Struktur der tail des ersten Teils ist und dass der head der Struktur der head des zweiten Teils ist /Gon78/. Abbildung 4-2, die sich an Abbildung 1.6 in /Gon78/ anlehnt, verdeutlicht diese Zusammenhänge.

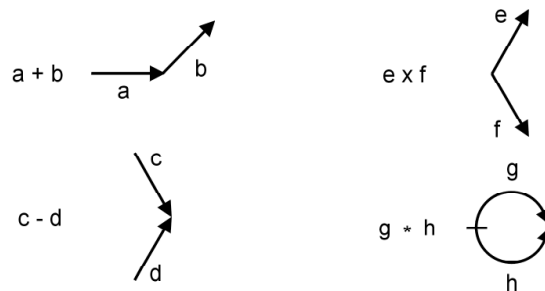


Abb. 4-2: Verbindung von PDL-Zeichenketten

Der Buchstabe "A" kann dann z.B., wie in Abbildung 4-3 gezeigt, zusammengesetzt werden: Die Menge der terminalen Symbole ist  $V_T$ . Im 1. Schritt wird die Spitze von "A" mit  $(a + b)$  erzeugt. Anschließend wird mit  $((\dots) * c)$  der waagerechte Mittelstrich eingefügt. Schließlich werden mit  $((\dots) + b)$  der rechte Abstrich und mit  $(a + (\dots))$  der linke Aufstrich hinzugefügt.

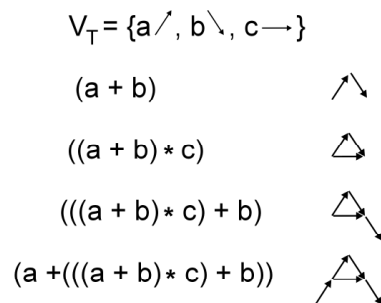


Abb. 4-3: Das Zeichen "A" als PDL-Zeichenkette

Eine Anwendung von PDLs zur Auswertung von Blasenkammeraufnahmen erläutert /Gon78/. Andere Beispiele für erweiterte syntaktische Verfahren sind Generierungs- und Analyseverfahren, die auf Baum-, Graph-, PLEX- und Shape-Grammatiken aufbauen und z. B. in /Gon78/, /Bun90/ und /Mar98/ dargestellt sowie in /Mich98a/ diskutiert werden.

#### 4.1.3 Vergleich mit Koordinatengrammatiken

Für Musteranalysen, die mit zuordnenden Erkennungsverfahren beschrieben werden, können auch Koordinatengrammatiken /Mil72, Sha69/ eingesetzt werden. Von Michaelsen /Mich98a/ wurden beispielsweise Koordinatengrammatiken für die



3-dimensionale Szenenanalyse mit Produktionsnetzen präzisiert und erweitert sowie Objekte 3-dimensional in Bildfolgen analysiert.

Die von Michaelsen verwendeten erweiterten Koordinatengrammatiken unterscheiden sich von den in der vorliegenden Arbeit definierten Systemen und Automaten im Wesentlichen in den anschließend aufgeführten Punkten:

1. Grundlagen:
  - (a) Michaelsen baut auf Koordinatengrammatiken  $(T, N, D, n, P, g)$  auf mit  $T$ : Menge von Terminalen,  $N$ : Menge von Nichtterminalen,  $V = T \cup N$ : Alphabet,  $D = (D_1, D_2, \dots, D_n)$ : Attributbereich,  $P = (\Lambda, \Sigma, \pi, \phi)$ : Produktionen mit  $\Lambda, \Sigma$ : Tupel von Symbolen,  $\pi \subseteq D^k$ : Prädikat,  $\phi$ :  $j$ -Tupel von  $k$ -stelligen Funktionen  $\phi : D^k \rightarrow D^j$ ,  $g$ : Startsymbol (Abschnitt 2.1.1 in /Mich98a/).
  - (b) Die vorliegende Arbeit baut auf Transformations- bzw. Reduktionssystemen auf.
2. Instanzen:
  - (a) Koordinatengrammatiken definieren Instanzen als Elemente des Produktes  $V \times D_1 \times D_2 \times \dots \times D_n$ . Dabei sind  $V$  eine Menge von Symbolen und  $D_i$  Mengen von Merkmalen.
  - (b) In der vorliegenden Arbeit gilt:
    - i. Zuordnungssysteme geben die Menge der Instanzen  $\mathcal{I}$  vor. Deren Eigenschaften sind unabhängig von einer speziellen Repräsentation.
    - ii. Zuordnungssysteme mit Konzepten definieren Instanzen mit Produktionen und einer Instanzierungsfunktion.
    - iii. Zuordnungssysteme mit Merkmalvektoren  $\gamma \in \mathcal{I}_M = T \times M_2 \times M_3 \times \dots$  definieren Instanzen als Elemente Kartesischer Produkte. Diese Instanzen entsprechen weitgehend den von Michaelsen definierten. Der Menge der Symbole  $V$  entspricht das Typ-Merkmal  $T$ . Die Merkmale  $M_i, i = 2, 3, \dots$  können als Koordinaten  $D_i$  interpretiert werden.
    - iv. Zuordnungssysteme mit Ableitungen verwenden Instanzen mit einem Merkmalvektor und einem Index sowie einer Indexmenge, mit der vermerkt wird, aus welchen Instanzen eine Instanz entstand.
3. Konzepte:
  - (a) Michaelsen arbeitet ohne Konzepte und berechnet die Merkmale neuer Instanzen mit einer Generierungsfunktion aus den Merkmalen derjenigen Instanzen, die durch die neuen Instanzen beschrieben werden.
  - (b) Die vorliegende Arbeit definiert mit Konzepten Produktionen.
4. Assoziationen:
  - (a) Die vorliegende Arbeit bezeichnet alle Paare von Konfigurationen, die während einer Analyse oder Generierung eine Rolle spielen können, als Assoziationen.
  - (b) Michaelsen verzichtet auf die Definition aller Paare von Konfigurationen, die mögliche Analyseschritte angeben. Deshalb kann die Analyse mit Koordinatengrammatiken nicht mit Transformationssystemen beschrieben werden.

5. Produktionen:
  - (a) Michaelsen listet alle Paare von Zeichenketten aus den Symbolen einer Menge  $V$  auf, mit denen überprüft und generiert werden kann. Eine Zeichenkette entspricht einer Konfiguration, so dass Paare von Zeichenketten in vieler Hinsicht den in der vorliegenden Arbeit definierten Produktionen entsprechen. Dabei wird die Zeichenkette ausschließlich dazu verwendet anzugeben, von welchem "Typ" die betrachteten Instanzen sind. Die Reihenfolge der Symbole, die bei Zeichenkettengrammatiken entscheidend ist, wird nicht berücksichtigt, so dass die Zeichenkette eine Menge definiert.
  - (b) Die vorliegende Arbeit listet alle Paare von Konfigurationen mit Konzepten auf, aus denen Assoziationen abgeleitet werden können, und nennt diese Paare Produktionen.
6. Instanziierungsfunktion:
  - (a) Michaelsen überprüft für den Test zu analysierender Instanzen ein Prädikat  $\pi$ . Zur Berechnung der Merkmale neu generierter Instanzen wird eine davon unabhängige Funktion  $\phi$  verwendet.
  - (b) In der vorliegenden Arbeit wird für den Test zu analysierender Instanzen und zur Berechnung der Merkmale der neu generierten Instanzen, die die zu analysierenden Instanzen beschreiben, eine Instanziierungsfunktion verwendet.
7. Produktionsnetze und Ableitungsgraphen werden von Michaelsen und in der vorliegenden Arbeit vergleichbar definiert.
8. Beschreibungsgraphen werden nur in der vorliegenden Arbeit definiert.
9. Parser:
  - (a) Michaelsen definiert Parser über Algorithmen, die in einer PASCAL-ähnlichen Syntax formuliert sind.
  - (b) Die vorliegende Arbeit definiert Zuordnungsautomaten, die Parser realisieren, über Relationen.
10. Erzeugungs- und Erkennungsverfahren:
  - (a) Michaelsen definiert mit Parsern Erkennungsverfahren.
  - (b) Die vorliegende Arbeit definiert Erzeugungs- und Erkennungsverfahren für Konfigurationen.
11. Beweise zur Konfluenz und Konvergenz, zu Aussagen über die Analyse mit Parallelprozessoren und -rechnern sowie zur explorativen Analyse mit der Datenerfassung in der Schleife werden nur in der vorliegenden Arbeit angegeben.
12. Aussagen über Hierarchien von Koordinaten-Grammatiken sowie Produktionsnetzen und die damit verbundene Komplexität werden nur von Michaelsen gemacht.

Vorteil des von /Mich98a/ entwickelten Ansatzes ist, dass er auf etablierten Verfahren sowie Veröffentlichungen aufbaut und Hierarchien von Grammatiken sowie die damit verbundene Analysekomplexität betrachtet.

Nachteil ist, dass keine Erzeugungsverfahren definiert wurden, so dass gewisse Fragestellungen nur unzureichend diskutiert werden können. Zum Beispiel ist in

/Mich98a/ der Vergleich zwischen ersetzenden und zuordnenden (kumulierenden) Verfahren insoweit unvollständig, als davon ausgegangen wird, dass ersetzende Verfahren keinen Einschränkungen unterliegen und zuordnende Verfahren bis auf Vorteile bei der Rechenzeit nur Nachteile bringen. In Wirklichkeit können mit mehrfach assoziierenden generierenden Assoziationen entstandene Konfigurationen nur zuordnend, nicht aber ersetzend erkannt werden (Satz B.84), so dass zuordnende Verfahren neben geringfügigen Nachteilen auch einen grundlegenden Vorteil gegenüber ersetzenden Verfahren haben. Ein weiterer Nachteil von Koordinatengrammatiken ist die Repräsentation von Mengen durch Zeichenketten. Was bedeutet es z. B., wenn in einer Zeichenkette ein Symbol mehrfach erscheint? Soll damit eine Multimenge /Cae97/ repräsentiert werden, in der Elemente mehrfach enthalten sein können?

## 4.2 Weitere Ansätze

Im Abschnitt 1.3 werden der PANDEMONIUM-Ansatz sowie tafelbasierte Systeme als Ausgangspunkte des hier entwickelten Ansatzes vorgestellt. Anschließend werden diese Ansätze sowie semantische Netze und regelbasierte Systeme mit dem in dieser Arbeit entwickelten Ansatz verglichen.

### 4.2.1 Vergleich mit PANDEMONIUM-Ansatz

Der 1958 vorgestellte PANDEMONIUM-Ansatz /Sel58/ beeinflusste viele Bereiche /And88/. Die Idee, eine Analyse mit mehreren voneinander unabhängigen Modulen durchzuführen, wird von Newell /New62/ und Simon /Sim66/ mit ihren Tafel- und Produktionssystemen übernommen (vgl. auch Seite 6). Künstliche Neuronale Netze greifen zusätzlich das Lernen der Module und ihrer Parametrisierung auf /And88/.

Die in dieser Arbeit vorgestellte Musteranalyse mit Produktionsnetzen ist ein weiterer Ansatz, der mit Assoziationen und Produktionen den PANDEMONIUM-Gedanken aufnimmt und weiterführt. Das Lernen, das ein wesentlicher Aspekt in der Veröffentlichung /Sel58/ mit dem Titel *PANDEMONIUM: a paradigm for learning* war, spielt bei der Musteranalyse mit Produktionsnetzen allerdings wie bei tafelbasierten Systemen keine Rolle. Die Produktionen und ihr Zusammenspiel müssen vom Entwickler formuliert und programmiert werden.

Bemerkenswert am PANDEMONIUM ist, dass es der erste Ansatz zur maschinellen Durchführung von Wahrnehmungsaufgaben ist und viele Ansätze von ihm inspiriert wurden.

Nachteil des PANDEMONIUMs ist, dass es in den 50-er Jahren noch keine ausreichend leistungsfähigen Rechner gab, so dass das PANDEMONIUM nicht zum praktischen Einsatz kam.

### 4.2.2 Vergleich mit tafelbasierten Systemen

Das im Rahmen des DARPA-Sprach-Analyse-Programms entwickelte HEARSAY II-System /Erm80/ verhalf der Blackboard-Idee<sup>1</sup> zu großer Popularität (vgl. auch Seite 6). HEARSAY II war in der Lage, gesprochene Anfragen an eine Datenbank mit ausgewählten Veröffentlichungen maschinell zu verstehen, in denen 1000 unterschiedliche Wörter zugelassen wurden. Im VISION-System /HRi78/ wird der Tafel-Ansatz das erste Mal zur automatischen Bildanalyse eingesetzt. Im von Ohta /Oht85/ vorgestellten Segmentationssystem wird der Tafel-Ansatz mit Produktionsregeln und aktiven semantischen Netzen /Bra77/ verbunden. Levine schlägt in /Lev78/ ein System vor, dessen erste zwei Ebenen von Shaheen /Sha79/ und Nazif /Naz83/ implementiert wurden. Shaheen stellte mit einer relationalen Datenbank die unterste Ebene des von Levine vorgeschlagenen Systems bereit, Nazif entwickelte die Produktionsregelebene. Für die dritte Ebene war zur Optimierung der Auswertung der A\*-Algorithmus vorgesehen.

Vorteil tafelbasierter Systeme ist ihre konsequente Modularisierung mit einfachen einheitlichen Schnittstellen. Problemlösungsansätze und die Implementierung werden dadurch vereinheitlicht und nachvollziehbar.

Nachteil einiger tafelbasierter Systeme ist, dass der Ansatz nicht immer konsequent verfolgt und z. B. dadurch verwässert wird, dass gemeinsame Dateien als Tafel bezeichnet werden, ohne dass präzise Modulschnittstellen vorliegen. Außerdem lassen sich aus der Tafelorganisation keine Aussagen über die Qualität der Analyseergebnisse (wie z. B. Konvergenz) ableiten.

### 4.2.3 Vergleich mit regelbasierten Systemen

Regelbasierte Systeme, die auch Produktionssysteme /Win84/ genannt werden, sind ein weiterer Ansatz zur strukturellen Musteranalyse.

Unter einem Produktionssystem wird in diesem Zusammenhang<sup>2</sup> ein mit sogenannten *Produktionsregeln* /Bar81, Leh85, Hrf85, Win84, Nie89/ arbeitendes System einschließlich Regelspeicher und (Arbeits-)Speicher der aktiven Einträge verstanden. Jede Regel wird aus einem Bedingungsteil und einem Aktionsteil aufgebaut. Über den Bedingungsteil wird die Aktion einem gewissen Zustand des Arbeitsspeichers zugeordnet und festgelegt, in welcher Situation die Aktionen der Regel auszuführen sind.

Steht wie bei Nazif /Naz83/ und Ohta /Oht85/ in Bildauswertungssystemen die Formulierung mit Produktionsregeln im Vordergrund, so ist mit der Syntax der Wenn-Dann-Produktionsregeln zu programmieren. Ob die Sprachsyntax der Produktionsregeln entscheidende Vorteile gegenüber der Syntax klassischer Programmiersprachen (z. B. C) bietet, ist umstritten. Der Ablaufzyklus der Produktionssysteme vereinfacht zwar bei geeigneten Anwendungen die Entwicklung komplexer

---

<sup>1</sup>Blackboard-System = Tafel-System

<sup>2</sup>Semi-Thue-Systeme (vgl. Gl. 2-6) werden ebenfalls als Produktions- oder Regelsysteme bezeichnet /Ste88/.

Systeme mit Fallunterscheidungen; für die automatische Bildauswertung sind Ansätze, die Assoziationsabläufe simulieren, aber sicher problemgerechter als der übliche Produktionsregelzyklus, der z. B. schon bei einfachen Anwendungen die Gefahr beinhaltet, Endlosschleifen zu programmieren /For81/. Um dies zu verhindern, sind spezielle Regeln und Hilfskonstruktionen einzuführen, die Regelsätze schwer verständlich machen können.

Außerdem muss bei klassischen regelbasierten Systemen wie z. B. OPS5 /For81/ in jedem der  $z$  Ausführungszyklen jeder der  $n$  Einträge im Arbeitsspeicher von jeder der  $r$  Regeln überprüft werden, so dass die Bearbeitung umfangreicher Datensätze zu extremen Rechenzeiten führen kann.

Vorteil dieser regelbasierten Systeme ist das einfache Ablauf- und Programmierschema für Fallunterscheidungen.

Nachteil ist, dass das Ablauf- und Programmierschema für viele Anwendungen zu starr ist und sich aus diesem Schema keine Aussagen über die Qualität (z. B. Konvergenz) der Analyseergebnisse ableiten lassen. Außerdem müssen mit großem Rechenzeitaufwand in jedem Zyklus alle Regeln alle Arbeitsspeichereinträge überprüfen.

#### 4.2.4 Vergleich mit wissensbasierten Systemen

In den 80-er Jahren wurden Bildanalysesysteme entwickelt, mit denen komplexe Strukturen über mehrere Abstraktionshierarchien analysiert werden. Forschungsthemen waren z. B. die Repräsentation der Objekteigenschaften und der Zusammenhänge zwischen den Objekten sowie deren Nutzung zur Analyse. In diesem Zusammenhang wurden die Begriffe *Wissen*, *Wissensrepräsentation* sowie *wissensbasierte Systeme* geprägt /Nie89, Bun85, Bal82, Cae97/.

Problem vieler wissensbasierter Systeme zur automatischen Bildauswertung ist, dass sie von der Merkmalextraktion eindeutige Segmentationsergebnisse fordern und Segmentationsfehler in der Regel durch die nachfolgende symbolische Verarbeitung nicht mehr korrigiert werden können /Bin82/. Steht die Segmentationsaufgabe im Vordergrund, wie in der Arbeit von Ohta /Ohta85/, so werden teilweise unrealistische Vereinfachungen vorgenommen, um dadurch Segmentationsflächen eindeutig zu benennen. Ohta behandelt z. B. keine widersprüchlichen Interpretationen: Treten konkurrierende Alternativen auf, so werden sie gestrichen. Auf Seite 94 wird in /Ohta85/ geschrieben: ... *every action-patch pair, which is going to give a different interpretation to the patch is declared inconsistent and deleted from the agenda* ...

Nachteil der in den 80-er Jahren realisierten wissensbasierten Systeme ist, dass sie ohne formale Grundlage eher experimentell entstanden sind. Die mit ihnen produzierten Analyseergebnisse sind daher in der Regel kaum nachzuvollziehen. Ein weiteres Problem ist die oft schlechte Reproduzierbarkeit der Ergebnisse.

Neuer Beitrag der Systeme war, dass alle für eine Anwendung notwendigen Analyseschritte in einem Gesamtsystem durchgeführt werden.

### 4.2.5 Vergleich mit semantischen Netzen

Die Musteranalyse mit semantischen Netzen kann unterschiedlich realisiert werden. Das in /Sag85/, /Sag97/ und /Nie90/ beschriebene ERNEST-System baut mehrere bewertete Folgen von Analyseschritten konkurrierend auf. Sagerer verwendet den A\*-Algorithmus /Nie89/ als *zulässige* Heuristik (admissible heuristic) /Ben96, Poo98/ zur Auswahl erfolgversprechender Analysezustände (bzw. Suchzustände). Quint /Qui97/ arbeitet im MOSES-System mit einem modifizierten A\*-Algorithmus. Diese Systeme speichern und bewerten Analysezustände. Grau /Gra00/ beschreibt und speichert im AIDA-System Analysezustände inkrementell und erreicht damit eine deutliche Reduktion des Speicher- und Rechenzeitbedarfs. Sagerer /Sag97/ und Niemann /Nie89/ geben eine umfassende Beschreibung von Ansätzen mit semantischen Netzen.

Beim A\*-Algorithmus werden für jeden Analysezustand  $\nu$  die zum Erreichen von  $\nu$  bereits entstandenen Kosten  $h(\nu)$  und die maximal zum Erreichen des Endergebnisses noch entstehenden Kosten  $g(\nu)$  angegeben /Bun85, Nie89, Sag85, Sag97/. Bei komplexen Musteranalysen ist allerdings die Bewertungs- bzw. Kostenfunktion eines Analysezustandes oder eines einzelnen Analyseschrittes ein Problem. Was bedeutet es z. B., wenn ein Zustand mit hoher Rechenzeit (d. h. großen Kosten) erreicht wurde? In der Regel heißt dies nicht, dass dieser Zustand schlechter ist als ein Zustand, der mit geringerem Rechenaufwand erreicht wurde. Die verbrauchte Rechenzeit spielt für zukünftige Überprüfungen keine Rolle. Oder was bedeutet eine schwach ausgeprägte, d. h. "schlechte" Linie, der hohe Kosten zugeordnet werden könnten, für die weitere Analyse?

Ein weiteres Problem ergibt sich, wenn man versucht, zur Aufwandsreduzierung Zwischenergebnisse mehrfach zu nutzen. In Algorithmus 3-10 könnte z. B. das Zwischenergebnis  $\nu_{1,2}$ , das in einer Folge von Analysezuständen erkannt wurde, zusätzlich in einer konkurrierenden anderen Folge von Analysezuständen verwendet werden. Diese Mehrfachnutzung von Zwischenergebnissen widerspricht aber den Voraussetzungen<sup>3</sup> für die zu Grunde liegende A\*-Optimierungsstrategie.

Abschnitt 3.3.2 diskutiert die Analyse mit bewerteten Analysezuständen.

In der Dissertation von Michaelsen /Mich98a/ werden u. a. semantische Netze und Koordinatengrammatiken miteinander verglichen.

Vorteil der semantischen Netze ist, dass sie allgemeine Zusammenhänge spezifizieren, ohne dass bei der Auswahl weiter zu analysierender Zustände angegeben werden muss, wie diese Zusammenhänge konkret für eine Analyse genutzt werden müssen. Solange Realisierungsfragen vernachlässigt werden, bietet dieser Ansatz eine große Flexibilität.

Als Nachteil ist Folgendes festzustellen: Konkrete Anwendungen im Bereich der Bildanalyse nutzen bei semantischen Netzen in der Regel nur Teil-von-Relationen zwischen Objektteilen<sup>4</sup>. Damit wird von der möglichen Flexibilität der Netze nur

<sup>3</sup>Die Zusammenfassung zweier Zwischenergebnisse würde zwei Analysezustände unzulässigerweise verknüpfen.

<sup>4</sup>Die oft noch verwendete "Ist-ein"-Relation dient in der Regel nur zur Programmierung mit Objektklassen.

noch eingeschränkt Gebrauch gemacht. Gleichzeitig werden aber oft noch Optimierungsansätze verwendet, die prinzipiell auch uneingeschränkte semantische Netze zu nutzen gestatten. Dabei werden Analysezustände gespeichert, bewertet und der jeweils optimal bewertete zur weiteren Analyse ausgewählt und expandiert /Qui97, Sag85/.

Ohne Bewertung müßten alle möglichen Folgen von Analyseschritten betrachtet werden (siehe z. B. /Gra00/ Seite 61). Dazu könnten z. B. die Algorithmen 3-4 und 3-10 eingesetzt werden. Für diese Algorithmen, die allerdings ohne die für semantische Netze eingesetzten modifizierten Konzepte und ohne die daraus abgeleiteten Suchbereiche formuliert sind, gilt, dass sie bei Teil-von-Hierarchien mit regulär binären balancierten (rbb) Ableitungsbäumen und Ausfall der Bewertung aufwändiger sind als Verfahren, die Teilobjekte speichern, bewerten und optimal bewertete zur weiteren Analyse auswählen (Satz B.99). Bei Ableitungsketten benötigen ersetzende und zuordnende Verfahren vergleichbaren Aufwand.

Tabelle 3-3 zeigt, dass auch bei Einsatz von Suchbereichen, die z. B. aus modifizierten Konzepten abgeleitet sind, die Graphsuche (Algorithmus 3-17), bei der Ableitungsbäume und keine Ableitungsketten entstehen, ohne Bewertung aufwändiger ist als die Suche mit gespeicherten Zwischenergebnissen (Algorithmus 3-22).

Weiterer Nachteil ist, dass das ERNEST-System nicht als formaler Automat, sondern nur über erweiterte Flussdiagramme /Sag97/ definiert ist.

## 5 Ausblick

Aufbauend auf den Ergebnissen der vorliegenden Arbeit werden in den folgenden Bereichen Forschungsarbeiten zur Weiterentwicklung der strukturellen Mustererkennung mit Produktionsnetzen gesehen:

1. Verbesserung der Merkmalschätzung gestörter terminaler Instanzen und Zwischenergebnisse entsprechend der Endergebnisse, zu denen die Instanzen führen.
2. Verbesserung der Modellierung durch Untersuchung und Klassifikation von Assoziationen-Mengen, um Eigenschaften der Analyse, z. B. Aussagen über Ableitungsgraphen, die Eindeutigkeit von Lösungswegen usw., machen zu können und um die speziellen Randbedingungen des hier verfolgten Ansatzes aufzuzeigen.
3. Reduzierung des Analyseaufwandes durch *von oben nach unten* (top down) organisierte Auswahlverfahren. Damit kann z. B. erreicht werden, dass aufwändige Analysen nur durchgeführt werden, wenn weniger aufwändige Analysen Hinweise auf Ergebnisse geliefert haben, für die die aufwändigen Analysen benötigt werden. Außerdem können Primitivobjekte, die aus gering aufgelösten Sensordaten stammen, Erwartungen für die Analyse von Primitivobjekten aus hoch aufgelösten Sensordaten liefern.
4. Untersuchung der Zusammenhänge zwischen der dynamischen Programmierung /Bel57/ und der strukturellen Musteranalyse mit Produktionsnetzen. Dies liegt nahe, da in beiden Fällen der Aufwand durch die Mehrfachnutzung von Zwischenergebnissen reduziert wird. Beispielsweise ist der CYK-Parser, der wie die BPI-Architektur Zwischenergebnisse abspeichert, eine Methode der dynamischen Programmierung /Aho72/<sup>1</sup> (vgl. z. B. Tabelle 8 Spalte *dynamic programming parsing* in /Bun90/).
5. Untersuchung der Zusammenhänge zwischen allgemeinen Suchverfahren und Zuordnungsautomaten, Zuordnungssystemen sowie den im Abschnitt 3.3 definierten Algorithmen.

In den vorhergehenden Kapiteln wird als einziger Algorithmus der Zuordnungsautomat angegeben. Er speichert Zwischenergebnisse und ist nach Definition ein Zuordnungssystem. Für Aufwandsdiskussionen werden im Anhang weitere Algorithmen angegeben. Zum einen wird eine Analyse, die nur Ableitungsketten erzeugt, zum anderen wird eine Analyse, die regulär binäre balancierte (rbb) Ableitungsbäume erzeugt, betrachtet. Diese Algorithmen generieren alle aus den zu analysierenden Daten ableitbaren Zwischenergebnisse mit einer für den jeweiligen Spezialfall geeigneten Ablaufkontrolle.

Bei den speichernden Algorithmen 3-7 und 3-19 ist unmittelbar einzusehen, dass die Algorithmen korrekt zuordnend ablaufen.

Bei den nicht speichernden Algorithmen, mit denen der Aufwand für Ersetzungssysteme abgeschätzt wird, ist nur der Algorithmus 3-4, der Ableitungsketten erzeugt, einfach nachzuvollziehen.

---

<sup>1</sup>1. Satz im Abschnitt 4.2.1



Die nicht speichernden Algorithmen 3-10 und 3-17, die Ableitungsbäume erzeugen, sind nicht ohne weiteres als geeignete Repräsentanten für ersetzende Algorithmen zu erkennen. Arbeitshypothese ist, dass ersetzende Verfahren immer als Graphsuche /Ben96, Poo98/, die höchstens Analysezustände aber keine nichtterminalen Instanzen speichert, realisiert werden können.

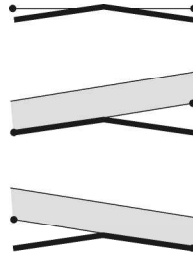
6. Einsatz von Verfahren zur Ballungsanalyse (Clustern), um mehrfach vorliegende, leicht unterschiedlich beschriebene Zwischenergebnisse bei der Analyse durch einen vorrangig zu bearbeitenden Repräsentanten zu ersetzen.
7. Einteilung der Zwischenergebnisse für nur teilweise vorliegende Muster in erforderliche und optionale Teile unter der Randbedingung, dass die Anzahl der dabei auftretenden Mehrfachgenerierungen minimiert wird.
8. Einsatz lernender Verfahren zur Beschleunigung der Analyse durch Optimierung der Auswahlfunktion  $a$  von Zuordnungsautomaten.
9. Verallgemeinerung der Aufwandsabschätzungen im Abschnitt 3.3 für beliebige Ableitungsbäume und nicht nur für Ableitungsketten und rbb-Ableitungsbäume.

Fortschritte im Bereich der Merkmalschätzung (1. Punkt der letzten Aufzählung) erhöhen die Präzision der Analyseergebnisse (Abschnitt 5.1). Die Qualifizierung der Assoziationen (2. Punkt der Aufzählung) verbessert die Modellierung (Abschnitt 5.2). Die anderen Arbeiten dienen in erster Linie zur Reduzierung des Analyseaufwandes (Abschnitt 5.3).

## **5.1 Verbesserung der Beschreibung der Zwischenergebnisse "von oben nach unten"**

Die strukturelle Musteranalyse mit Produktionsnetzen beginnt mit der Extraktion von Primitivobjekten aus den gemessenen Sensordaten (z. B. von Linienstücken und Flächensegmenten aus Bildern). Dabei wird davon ausgegangen, dass zum Zeitpunkt der Extraktion der Primitivobjekte keine Zusammenhänge zwischen den Primitivobjekten bekannt sind, die darüber entscheiden könnten, wie ein aus gestörten Primitivobjekten aufgebautes Zwischenergebnis (z. B. ein Streifen) am besten zu beschreiben wäre.

Sobald ein Zwischenergebnis (oder Primitivobjekt) zur Generierung anderer Zwischenergebnisse dient und damit Teil eines Ableitungsgraphen wird, ergeben sich neue übergeordnete Zusammenhänge. Nach Abschluß der Gesamtanalyse können deshalb die Eigenschaften der gestörten Zwischenergebnisse (und der Primitivobjekte) bezüglich der Endergebnisse, zu deren Generierung sie geführt haben, in der Regel sehr viel genauer geschätzt werden als zum Zeitpunkt ihrer Generierung. Dabei werden für jedes Primitivobjekt und Zwischenergebnis (d. h. jede Instanz) so viele Merkmalvektoren geschätzt, wie es Ableitungsgraphen gibt, in denen die Instanz enthalten ist. Danach können mit den korrigierten Instanzen weitere Analysen durchgeführt werden. Abbildung 5-1 gibt ein Beispiel für diese Zielsetzung.



**Abb. 5-1:** Top-down-Verbesserung von Ergebnissen  
 oben: Näherung für ein Geradenstück ohne Kontext  
 Mitte: Näherung, falls eingebaut in links geneigten Streifen  
 unten: Näherung, falls eingebaut in rechts geneigten Streifen

## 5.2 Verbesserung der Modellierung durch Beschreibung der Eigenschaften von Assoziationen

Sobald Muster durch ungeordnete Mengen beschrieben werden, muss die Einbettung von Teilmustern in übergeordnete Muster durch die Assoziationen erreicht werden. Die sonst z. B. mit Zeichenketten oder Graphen erreichte Einbettung über die Repräsentation der Muster entfällt, so dass mit Assoziationen definierte Transformationsvorschriften sowohl die beabsichtigten Umformungen als auch die einzuhaltende Einbettung beschreiben müssen. Weitere Anforderungen an Assoziationen entstehen durch die zuordnende Analyse nach Gleichung 2-8. Da der Entwickler in der Regel bei der Formulierung seiner Assoziationen ersetzend entsprechend Gleichung 2-7 denkt, muss sichergestellt werden, dass zuordnend keine unsinnigen Muster entstehen (vgl. Seite 50 ff.).

### 5.2.1 Syntaktische Beschreibung der Eigenschaften von Assoziationen

Die am Ende von Abschnitt 2.2.3 (Seite 25 ff.) aufgelisteten Eigenschaften von Assoziationen werden vorgeschlagen, um die Einbettung und die zuordnende Analyse zu präzisieren. Als Einstieg in weiterführende Arbeiten können die folgenden Zusammenhänge dienen (vgl. Abschnitte 2.3 und 3.4.3):

1. Mengen von Assoziationen sind genau dann *lokal kontextfrei* und *hierarchisch*, wenn sie *global kontextfrei* sind (Satz B.42).
2. Analysierende Mengen von Assoziationen  $\mathbf{A}^a = (\mathbf{A}^g)^{-1}$  sind genau dann lokal kontextfrei, wenn die generierenden Assoziationen  $\mathbf{A}^g$  lokal kontextfrei sind (Satz B.43).
3. Die Vereinigung ersetzend generierter Konfigurationen kann nur mit mehrfach assoziierenden Assoziationen generiert (Lemma B.85) und muss zuordnend analysiert werden (Satz B.86).
4. Wenn Instanzen Teil mehrerer Instanzen sind, müssen mehrfach assoziierende Assoziationen eingesetzt werden (Lemma B.87), und es muss zuordnend analysiert werden (Satz B.88).

5. Bei der Analyse von Instanzen, die mehrfach assoziierend generiert wurden, entstehen Ableitungsgraphen mit Zyklen (keine Bäume, Satz B.90).

### 5.3 Reduzierung des Analyseaufwandes

Die anschließend erläuterten Maßnahmen dienen der Reduzierung des Aufwandes bei der strukturellen Analyse mit Produktionsnetzen. Die konzeptionell ausschließlich von *unten nach oben* (bottom up) organisierte Analyse, die mit terminalen Instanzen beginnt und abschließend Startsymbole zuordnet, ignoriert globale Zusammenhänge bei der Auswahl der als nächstes durchzuführenden Analyseschritte. Durch zusätzliche Maßnahmen, insbesondere die "von oben nach unten" (top down) organisierte Berücksichtigung globaler Zusammenhänge, sollte der Aufwand reduziert werden können.

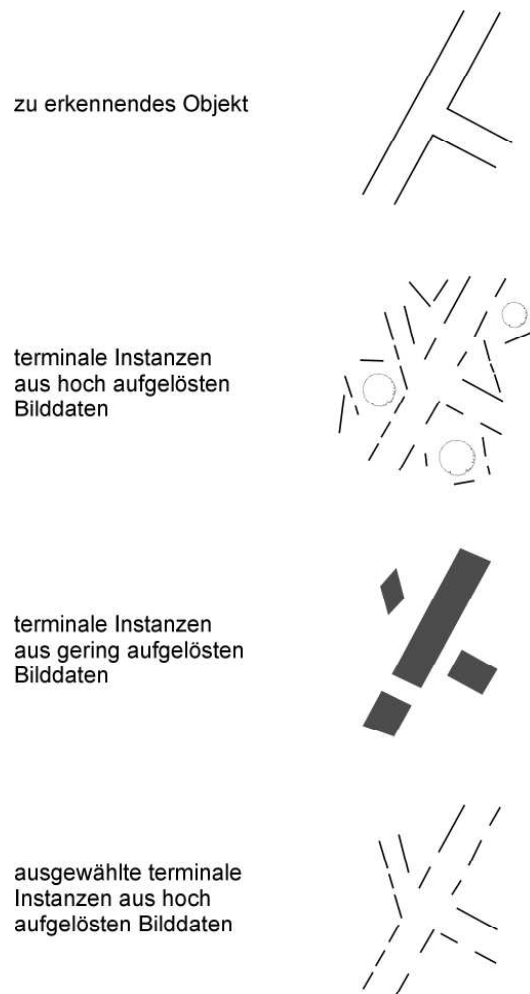
#### 5.3.1 Detailanalyse nach Einweisung

Der Aufwand zur Analyse einer Menge terminaler Instanzen kann u. U. reduziert werden, wenn (nichtterminale) Zwischenergebnisse zusätzlich redundant als (terminale) Primitivobjekte bestimmt werden /Kum97/. Dazu können beispielsweise unterschiedlich aufgelöste Sensordaten aus Auflösungspyramiden /Bal82/ bearbeitet werden. Primitivobjekte aus gering aufgelösten Sensordaten können in hoch aufgelöste Sensordaten eingeblendet werden und dort Erwartungsbereiche für die Analyse hoch aufgelöster Objekte definieren. Bei der Analyse entstehen nichtterminale Instanzen aus höher aufgelösten Primitivobjekten und liefern eine höher aufgelöste detaillierte Beschreibung der aus den gering aufgelösten Sensordaten extrahierten Primitivobjekte (Abb. 5-2).

Ohne die aus den gering aufgelösten Sensordaten extrahierten Primitivobjekte müßte die Analyse ausschließlich mit den Primitivobjekten aus den hoch aufgelösten Sensordaten arbeiten. Da die Gesamtanzahl aller hoch aufgelösten Primitivobjekte wesentlich größer ist als die Anzahl der hoch aufgelösten Primitivobjekte, die sich im Bereich der gering aufgelösten Primitivobjekte befinden, verspricht die redundante Extraktion zusätzlicher gering aufgelöster Primitivobjekte eine signifikante Reduzierung des Aufwandes für die strukturelle Analyse der hoch aufgelösten Primitivobjekte.

#### 5.3.2 Verzögerter Einsatz aufwändiger Analyseprogramme

Der Aufwand zur Analyse einer Menge terminaler Instanzen kann reduziert werden, wenn Zwischenergebnisse einer Analysestufe  $i$  mit unterschiedlichem Aufwand generiert werden. Da für ein Zwischenergebnis auf Stufe  $i + 1$  Zwischenergebnisse beider Typen (der mit hohem Aufwand zu generierenden und der mit wenig Aufwand zu generierenden) auf der Stufe  $i$  vorliegen müssen, genügt es, anfangs nur diejenigen Vorverarbeitungs- und Extraktionsprogramme ablaufen zu



**Abb. 5-2:** Detailanalyse mit Einweisung

lassen und diejenigen Assoziationen einzusetzen, mit denen die aufwandsgünstigen Zwischenergebnisse entstehen. Erst wenn diese aufwandsgünstig bestimmten Ergebnisse wirklich vorliegen, müssen zusätzlich die aufwandsintensive Extraktion der anderen Primitivobjekte sowie Überprüfung von Assoziationen für die noch fehlenden Zwischenergebnisse der Stufe  $i$  ausgeführt werden.

Durch diesen *von oben nach unten* (top down) gesteuerten Einsatz sollte eine deutliche Reduktion des Aufwandes erreicht werden. Dazu muss u. a. die Auswahlfunktion der Zuordnungsautomaten (Gl. B-84) geeignet definiert werden.

### 5.3.3 Vermeidung von Mehrfachgenerierungen mit Ballungsanalysen

Der bei Zuordnungsautomaten notwendige Analyseaufwand hängt u. a. davon ab, ob es gelingt, Zwischenergebnisse, die für identische Ereignisse in einer zu analysierenden Szene stehen, im Analyseablauf tatsächlich nur durch eine Instanz

zu beschreiben und unnötige Instanzen zu vermeiden (vgl. Abschnitte 3.4.2 und 3.4.3).

Wenn die Analyse Zwischenergebnisse mehrfach generiert, könnten Ballungsanalysen (Clustern) /Bal82/ zur Zusammenfassung mehrfach generierter Instanzen den Analyseaufwand mit Zuordnungsautomaten reduzieren.

#### 5.3.4 Analyse mit erforderlichen und optionalen Zwischenergebnissen

Man kann bei der Musteranalyse nicht davon ausgehen, dass alle Primitivobjekte, die in sehr guten Situationen zur Verfügung stehen, auch unter schwierigen Grenzbedingungen ausgewertet werden können. Deshalb sollte zwischen *erforderlichen* sowie *optionalen* Primitivobjekten bzw. Zwischenergebnissen unterschieden werden. Dadurch können alternative Analysepfade in Produktionsnetzen entstehen, zu denen nicht deterministische Assoziationen (vgl. Abschnitt 3.4.3) gehören.

Um den dadurch bedingten erhöhten Analyseaufwand wieder zu reduzieren, kann eine Bewertung sowie eine bewertungsgesteuerte Auswahl und Analysesteuerung implementiert werden. Ziel ist, Ergebnisse mit möglichst vielen optionalen Teilen zuerst zu analysieren. Danach gefundene Zwischenergebnisse, bei denen optionale Teile fehlen, die aber sonst mit vollständigeren Zwischenergebnissen übereinstimmen, müssen als bereits erfasst (d. h. in diesem Sinne als bereits vorliegend) erkannt und dürfen nicht weiter analysiert werden.

## 6 Zusammenfassung

Für die strukturelle Analyse komplexer Muster wird ein Ansatz definiert und detailliert beschrieben.

Ausgehend von der Zielsetzung, tafelbasierte Analysesysteme für zeitkritische Anwendungen auf Parallelrechnern auszuführen, wird die Aufgabe bearbeitet, für strukturelle Bildanalyseverfahren, die Produktionsnetze einsetzen, eine neue formal abgesicherte Basis zu schaffen. Hierfür werden Zuordnungssysteme und -automaten, die Mengen von Instanzen manipulieren, sowie Erzeugungs- und Erkennungsverfahren, die den Bezug zur strukturellen Musteranalyse herstellen, entwickelt.

Zu beachtende Randbedingungen und mögliche Probleme bei der Formulierung von Produktionen werden durch die Definition von Erkennungsverfahren eingegrenzt und verdeutlicht. Aufbauend auf speziellen Relationen für Zuordnungssysteme und -automaten wird nachgewiesen, dass die Verfahren konfluent bzw. bei endlichen Folgen von Analyseschritten konvergent sind und dass sowohl auf Prozessor- als auch auf Rechner Ebene parallel analysiert werden kann. Außerdem ändern sich die endgültigen Analyseergebnisse nicht, wenn Datensätze in Teilmengen aufgeteilt und seriell in beliebiger Reihenfolge teilweise analysiert werden, so dass explorative Analysen, bei denen die Datenerfassung auf vorläufigen Analyseergebnissen aufbaut und parallel zur Analyse erfolgt, durchgeführt werden können.

Die Aufteilung der Analysemodelle in Produktionen unterstützt die Modularisierung realisierter Rechnerprogramme sowie die Portierung einzelner Teilanalysen zur Implementierung neuer Programme. Produktionsnetze verdeutlichen das Zusammenspiel der Produktionen.

Die hier sowie in /Füg92/, /Sti93/, /Füg94/, /Sti94/, /Bäh95/, /Sti95a/, /Sti95b/, /Sti96a/, /Sti96b/, /Sti97a/, /Sti97b/, /Sti97c/, /Mich98a/, /Mich98b/, /Schä98/, /Schw98/ und /Sti98/ dargestellten Ergebnisse demonstrieren die Flexibilität und Robustheit der Musteranalyse mit Produktionsnetzen. Aufwandsabschätzungen zusammen mit den veröffentlichten Analyseergebnissen zeigen, dass mehrere 100.000 Instanzen umfassende Muster analysiert werden können.

Der Ansatz ist eine Weiterführung und Verallgemeinerung regelbasierter und syntaktischer Verfahren zur Analyse komplexer Muster, für die es bis heute keine Formulierungen als Automaten gibt. Der Ansatz unterscheidet sich von vergleichbaren anderen Ansätzen dadurch, dass das erste Mal für komplexe Bildanalysen wesentliche Zusammenhänge und Aussagen über die Formulierung als Automat formal definiert und bewiesen werden, so dass die Analyseergebnisse nachvollziehbar und reproduzierbar sind.

## 7 Literaturverzeichnis

- /Aho72/ Aho, A.V.; Ullmann, J.D.: *The Theory of Parsing, Translation, and Compiling. Vol.1.: Parsing*, Prentice-Hall, Englewood, 1972.
- /Alb78/ Alberts, J.; Schröder, M.: *Berichte zum Symposium Flugzeugmeßprogramm*, BMFT - FBW 78-04, 1978.
- /And88/ Anderson, J.A.; Rosenfeld, E.: *Neurocomputing, Foundations of Research*, MIT Press, Cambridge Mass., 1988.
- /ATH89/ Anderer, C.; Thönnessen, U.: *Ein Bildsegmentierer für die echtzeitnahe Verarbeitung*, in *Mustererkennung 1989, 11. DAGM Symposium* (Ed. H. Burkhardt, K.H. Höhne, B. Neumann), Springer, Berlin, 1989.
- /Ave95/ Avenhaus, J.: *Reduktionssysteme - Rechnen und Schließen in gleichungsdefinierten Strukturen*, Springer, Berlin, 1995.
- /Bäh95/ Bähr, H.P.; Quint, F.; Stilla, U.: *Modellbasierte Verfahren der Luftbildanalyse zur Kartenfortführung*, ZPF Zeitschrift für Photogrammetrie und Fernerkundung, 63(6), 1995, 224-234.
- /Bal82/ Ballard, D.H.; Brown, C.M.: *Computer Vision*, Prentice-Hall, Englewood Cliffs, 1982.
- /Bar81/ Barr, A.; Feigenbaum, E.A. (Ed.): *The Handbook of Artificial Intelligence*, William Kaufmann, Los Altos, 1981.
- /Bel57/ Bellman, R.: *Dynamic Programming*, Princeton University Press, Princeton, N.Y., 1957.
- /Ben96/ Bender, E.A.: *Mathematical Methods in Artificial Intelligence*, IEEE Computer Society Press, Los Alamitos, Cal., 1996.
- /Ber80/ Bers, K.H.; Bohner, M.: *Erhöhung der Navigationsgenauigkeit eines Szenenkorrelators und die Anwendung von Landkarten als Referenzdaten*, ZTL 80, Bericht Nr. VF 3.13-3, 1980.
- /Bin82/ Binford, T.O.: *Survey of Model-Based Image Analysis Systems*, The International Journal of Robotics Research 1 (Spring 1982), 18-64.
- /Boo93/ Book, R.V.; Otto, F.: *String-Rewriting Systems*, Springer, Berlin, 1993.
- /Bra77/ Brachman, R.J.: *What's in a Concept: Structural Foundations for Semantic Networks*, Int.J. Man-Machine Studies 9 (1977), 127-152.
- /Bro81/ Brooks, R.A.: *Model-based Computer Vision*, Computer Science: Artificial Intelligence No. 14, UMI Research Press, Ann Arbor, 1981.
- /Bun85/ Bunke, H.: *Modellgesteuerte Bildanalyse*, Teubner, Stuttgart, 1985.
- /Bun90/ Bunke, H.; Saufeliiu, A.: *Syntactic and Structural Pattern Recognition - Theory and Applications -*, World Scientific, Singapore, 1990.
- /Cae97/ Caelli, T.; Bischof, W.F.: *Machine Learning and Image Interpretation*, Plenum, New York, 1997.

- /Cho65/ Chomsky, N.: *Aspects of the Theory of Syntax*, MIT Press, Cambridge Mass., 1965.
- /Dav86/ Davis, W.A.; Lee, D.: *Fast Search Algorithms for Associative Memories*, IEEE Trans. on Computers, C-36, 1986, 456-461.
- /Dör88/ Dörfler, W.: *Einführung in die Mathematik für Informatiker*, Hanser, München, 1988
- /Erm80/ Erman, L.D.; Hayes-Roth, F.; Lesser V.R.; Reddy, D.R.: *The Hearsay II Speech-Understanding System: Integrating Knowledge to Resolve Uncertainty*, Computing Surveys 12 (June 1980), 213-253.
- /Fen70/ Feng, T.: *Search Algorithms for Associative Memories*, Princeton Conf. on Inform. Sci. and Syst., 1970, 422-426.
- /For81/ Forgy, C.L.: *OPS5 User's Manual*, CMU-CS-81-135, Carnegie-Mellon University, Pittsburgh, 1981.
- /For83/ Forgy, C.L.: *Rete: A Fast Algorithm for the Many Pattern/ Many Object Pattern Match Problem*, Artificial Intelligence 19 (1982), 17-37.
- /Fu77/ Fu, K.S.: *Syntactic Pattern Recognition, Applications*, Springer, Berlin, 1977.
- /Füg92/ Fügler, H.; Jurkiewicz, K.; Lütjen, K.; Stilla, U.: *Ein wissensbasiertes System für die automatische Bildanalyse*, ISPRS, XVIIth Congress, International Archives of Photogrammetry and Remote Sensing, Vol. 29 (1992), Part B3, 167-172.
- /Füg94/ Fügler, H.; Stein, G.; Stilla, U.: *Multi-populations Evolution Strategies for Structural Image Analysis*, IEEE Conference on Evolutionary Computation (ICEC'94), Orlando, Vol I (1994), 229-234.
- /Ger79/ Gerlach, H.: *Digitale Bildfolgenauswertung zum Wiederfinden von Objekten in natürlicher Umgebung* (Diss.), Universität Karlsruhe, 1979.
- /Gib50/ Gibson, J.J.: *The Perception of the Visual World*, Houghton Mifflin, Boston, 1950.
- /Gol72/ Goldstine, H.H.: *The Computer from Pascal to von Neumann*, Princeton University Press, Princeton, 1972.
- /Gon78/ Gonzales, R.C.; Thomason, M.G.: *Syntactic Pattern Recognition*, Addison-Wesley, Reading, 1978.
- /Gra00/ Grau, O.: *Wissensbasierte 3D-Analyse von Gebäudeszenen mit mehreren frei gewählten Stereofotos* (Diss.), Universität Hannover, 2000.
- /Han78/ Hanson, A.R.; Riseman, E.M. (ed.): *Computer Vision Systems*, Academic Press, New York, 1978.
- /Hrb85/ Hayes-Roth, B.: *A Blackboard Architecture for Control*, Artificial Intelligence 26 (1985), 251-321.
- /Hol95/ Holmes, Q.A.; Schwartz, C.R.; Seldin, J.H.; Wright, J.A.; Witter, L.J.: *Adaptive Multispectral CFAR Detection of Land Mines*, Proc. SPIE, Vol. 2496



(1995), 421-432.

/Hrf85/ Hayes-Roth, F.: *Rule-Based Systems*, Communications of the ACM 28 (1985), 921-932.

/HRi78/ Hanson, A.R.; Riseman, E.M.: *VISION: A Computer Vision System for Interpreting Scenes*, in /Han78/, 303-333.

/Jan88/ Janzen, M.: *Confluent String Rewriting*, Springer, Berlin, 1988.

/Kaz80/ Kazmierczak, H. (Hrsg.): *Erfassung und maschinelle Verarbeitung von Bilddaten*, Springer, Wien, New York, 1980.

/Koe69/ Köhler, W.: *The Task of Gestalt Psychology*, Princeton, N.J., Princeton Uni. P., 1969.

/Koh80/ Kohonen, T.: *Content-Addressable Memories*, Springer, Berlin, Heidelberg, New York, 1980.

/Kum97/ Kummert, F.; Fink, G.A.; Sagerer, G.: *Schritthaltende hybride Objektdetektion*, in *Mustererkennung 1997, 19. DAGM Symposium* (Ed. E. Paulus, F.M. Wahl), Springer, Berlin, 1997.

/Leh85/ Lehr, T.F.: *The Implementation of a Production System Machine*, CMU-CS-85-126, Carnegie-Mellon University, Pittsburgh, 1985.

/Lev78/ Levine, M.: *A Knowledge Based Computer Vision System*, in /Han78/, 335-352.

/Lüt86/ Lütjen, K.: *Ein Blackboard-basiertes Produktionssystem für die automatische Bildauswertung*, in *Mustererkennung 1986, 8. DAGM Symposium* (Ed. G. Hartmann), Springer, Berlin, 1986.

/Mar98/ Marriott, K.; Meyer, B.: *Visual Language Theory*, Springer, New York, 1998.

/Mat90/ Matsuyama, T.; Hwang, V.: *Sigma - A Knowledge-Based Aerial Image Understanding System*, Plenum Press, New York, 1990.

/Mich96/ Michaelsen, E.: *3D Coordinate Grammars*, in *3D Image Analysis and Synthesis '96* (Ed. B. Girod, H. Niemann, H.-P. Seidel), Infix, Sankt Augustin, 1996.

/Mich98a/ Michaelsen, E.: *Über Koordinaten Grammatiken zur Bildverarbeitung und Szenenanalyse* (Diss.), Universität Erlangen, 1998.

/Mich98b/ Michaelsen, E.; Wankmüller, U.; Stilla, U.: *Wissenserwerb für Produktionsnetze zur 3D-Erkennung von Fahrzeugen*, in Levi, P. et. al. (eds) *Mustererkennung 1998*, Berlin, Springer, 1998.

/Mich99/ Michaelsen, E.; Lütjen, K.; Stilla, U.: *Associative Access and Special Hardware for Production Nets*, in: Radig, B., et. al. (eds), *Pattern Recognition and Image Understanding*, Infix, Sankt Augustin, 1996, 290-297.

/Mil72/ Milgram, D.L.; Rosenfeld, A.: *A Note on 'Grammars with Coordinates'*, in *Graphic Languages* (Ed. F. Nake, A. Rosenfeld), North Holland, 1972, 187-194.

- /Naz83/ Nazif, A.M.: *A Rule-based Expert System for Image Segmentation*, Thesis No. 83-6T, McGill University, Montreal, 1983.
- /Naz84/ Nazif, A.M.; Levine, M.D.: *Low Level Image Segmentation: An Expert System*, IEEE Trans. PAMI-6 (1984), 555-577.
- /New62/ Newell, A.: *Some Problems of Basic Organization in Problem-Solving Programs*, in Yovits, M.C. et al. (ed.), *Self-Organizing Systems*, Spartan Books, Washington, 1962, 393-423.
- /Nie89/ Niemann, H.: *Pattern Analysis and Understanding*, Springer, Berlin, 1989.
- /Nie90/ Niemann, H.; Sagerer, G.: *ERNEST: A Semantic Network System for Pattern Understanding*, IEEE Trans. PAMI-12 (1990), 883-905.
- /Oht85/ Ohta, Y.: *Knowledge-based Interpretation of Outdoor Natural Color Scenes*, Pitman Advanced Publishing Program, Boston, London, Melbourne, 1985.
- /Pfa69/ Pfaltz, J.L.; Rosenfeld, A.: *Web Grammars*, Technical Report 69-84, University of Maryland, Computer Science Center, 1969.
- /Poo98/ Poole, D.; Mackworth, A.; Goebel, R.: *Computational Intelligence; A Logical Approach*, Oxford University Press, New York, Oxford, 1998.
- /Qui97/ Quint, F.: *Kartengestützte Interpretation monokularer Luftbilder* (Diss.), Universität Karlsruhe, 1997.
- /Ros79/ Rosenfeld, A.: *Picture Languages*, Academic Press, New York, 1979.
- /Rub21/ Rubin, E.: *Visuell wahrgenommene Figuren*, Kopenhagen, 1921.
- /Rud72/ Rudolph, J.A.: *A Production Implementation of an Associative Array Processor: STARAN*, Proc. 1972 Fall Joint Computer Conf., 229-241.
- /Rus95/ Russell, S.J.: *Artificial Intelligence, A Modern Approach*, Prentice Hall, 1995.
- /Sag85/ Sagerer, G.: *Darstellung und Nutzung von Expertenwissen für ein Bild-analysesystem* (Diss.), Informatik-Fachberichte Bd. 104, Springer, Berlin, 1985.
- /Sag97/ Sagerer, G.; Niemann, H.: *Semantic Networks for Understanding Scenes*, Plenum, New York, 1997.
- /Sal78/ Salomaa, A.K.: *Formale Sprachen*, Springer, Berlin, Heidelberg, 1978.
- /Schä98/ Schärf, R.; Schwan, H.; Thönnessen, U.: *Reconnaissance in SAR Images*, Proceedings of the European Conference on Synthetic Aperture Radar (EUSAR), Berlin, VDE, 343-346.
- /Schl83/ Schlageter, G.; Stucky, W.: *Datenbanksysteme: Konzepte und Modelle*, Teubner, Stuttgart, 1983.
- /Schw98/ Schwan, H.; Schärf, R.; Thönnessen, U.: *Reconnaissance of Extended Targets in SAR Image Data*, in: Serpico, S.B. (eds), *Image and Signal Processing for Remote Sensing IV*, Europto, SPIE Proceedings, Vol: 3500, 164-171.
- /Sel58/ Selfridge, O.G.: *PANDEMONIUM: A Paradigm for Learning*, in Proc.

*Symp. on the Mechanization of Thought Processes*, National Physical Laboratory, HMSO London, 1958, 511-529.

/Sha69/ Shaw, A.C.: *A Formal Picture Description Schema as a Basis for Picture Processing Systems*, *Information and Control* 14 (1969), 9-52.

/Sha70/ Shaw, A.C.: *Parsing of Graph-Representable Pictures*, *J. ACM* 17 (1970), 453-481.

/Sha79/ Shaheen, S.I.: *Image Segmentation and Interpretation Using a Knowledge Database* (Diss.), McGill University, Montreal, 1979.

/Sim66/ Simon, H.A.: *Scientific Discovery and the Psychology of Problem Solving*, in Colodny, R.G. (ed.), *Mind and Cosmos: Essays in Contemporary Science and Philosophy*, Pittsburgh, 1966, 22-39.

/Ste88/ Stetter, F.: *Grundbegriffe der Theoretischen Informatik*, Springer, Berlin, 1988.

/Steg98/ Steger, C.: *An Unbiased Detector of Curvilinear Structures*, *IEEE Trans. PAMI-20* (1998), 113-125.

/Sti91/ Stilla, U.; Jurkiewicz, K.: *Objektklassifikation mit einem blackboardorientierten Inferenzmechanismus*, DFG Abschlußbericht Ka414/8-5; FGAN-FIM, Ettlingen, 1991.

/Sti93/ Stilla, U.: *Verfahrensvergleich zur automatischen Erkennung in Metallgeschlagener Zeichen* (Diss.), Universität Karlsruhe, 1993.

/Sti94/ Stilla, U.; Hajdu, A.: *Map-aided Structural Analysis of Aerial Images*, in: H. Ebner, C. Heipke, K. Eder (eds), *Spatial Information from Digital Photogrammetry and Computer Vision*, ISPRS Commission III Symposium, Munich, International Archives of Photogrammetry and Remote Sensing, Vol. 30, Part 3/2 (1994), 769-775.

/Sti95a/ Stilla, U.: *Map-aided Structural Analysis of Aerial Images*, *ISPRS Journal of Photogrammetry and Remote Sensing*, 50(4), 1994, 3-10.

/Sti95b/ Stilla, U.; Michaelsen, E.; Lütjen, K.: *Structural 3D-Analysis of Aerial Images with a Blackboard-based Production System*, in: A. Gruen, O. Kuebler, P. Agouris (eds), *Automatic Extraction of Man-made Objects from Aerial and Space Images*, Birkhäuser, Basel, 1995, 53-62.

/Sti96a/ Stilla, U.; Jurkiewicz, K.: *Structural 3D-Analysis of Urban Scenes from Aerial Images*, ISPRS, XVIIIth Congress, International Archives of Photogrammetry and Remote Sensing, Vol. 31 (1996), Part B3, 832-838.

/Sti96b/ Stilla, U.; Michaelsen, E.; Lütjen, K.: *Automatic Extraction of Buildings from Aerial Images*, in: F. Leberl, R. Kalliany, M. Gruber (eds), *Mapping Buildings, Roads and other Man-made Structures from Images*, Oldenburg, Wien, 1996, 229-244.

/Sti97a/ Stilla, U.; Michaelsen, E.: *Semantic Modelling of Man-made Objects by Production Nets*, in: A. Gruen, E.P. Baltsavias, O. Henricsson (eds) *Automatic Ex-*

*traction of Man-made Objects from Aerial and Space Images (II)*, Birkhäuser, Basel, 1997, 43-52.

/Sti97b/ Stilla, U.; Geibel, R.; Jurkiewicz, K.: *Building Reconstruction Using Different Views and Context Knowledge*, in: Baltsavias E.P. et.al. (eds), *3D Reconstruction and Modelling of Topographic Objects*, International Archives of Photogrammetry and Remote Sensing, Vol. 32, Part 3/4W2 (1997), 129-136.

/Sti97c/ Stilla, U.: *Extraktion von Gebäuden aus Luftbildern mit Produktionsnetzen*, in: List F.K. (eds), *Photogrammetrie im Wandel: Lehre - Forschung - Anwendung*, Publikation der Deutschen Gesellschaft für Photogrammetrie und Fernerkundung DGPF, Band 6, 347-355.

/Sti98/ Stilla, U.; Michaelsen, E.; Jurkiewicz, K.: *Structural Analysis of Right-Angled Building Contours*, in: Schenk, T., Habib, A. (eds), *Object Recognition and Scene Classification from Multi-Spectral and Multi-Sensor Pixels*, International Archives of Photogrammetry and Remote Sensing, Vol. 32, Part 3/1, 397-386.

/Taf71/ Tafel, H.J.: *Einführung in die digitale Datenverarbeitung*, Carl Hanser Verlag, München, 1971.

/Tel79/ TELDIX: *Taschenbuch der Navigation*, TELDIX GmbH, Heidelberg, 1979.

/Ter81/ Therburg, R.-D.; Mühlenfeld, E.: *Bildgestützte Tieffluggnavigation durch einen Kontursensor (Abschlußbericht)*, Bericht Automation 81/5, Technische Universität Clausthal-Zellerfeld, 1981.

/Thu14/ Thue, A.: *Probleme über Veränderungen von Zeichenreihen nach gegebenen Regeln*, Videnskapselskapets Skrifter, J. Math. Naturw. Kl (1914), 3-34.

/Thu79a/ Thurber, K.J.: *Parallel Processor Architectures - Part 1: General Purpose Systems*, Computer Design, Jan. 1979, 89-97.

/Thu79b/ Thurber, K.J.: *Parallel Processor Architectures - Part 2: Special Purpose Systems*, Computer Design, Feb. 1979, 103-114.

/Wat78/ Waterman, D.A.: *Pattern-Directed Inference Systems*, Academic Press, New York, 1978.

/Way86/ Wayne, A.D.; De-Lei Lee: *Fast Search Algorithms for Associative Memories*, IEEE Transactions on Computers, Vol. C-35, No. 5 (May 1986), 456-461.

/Wed76/ Wedekind, H.; Härder, T.: *Datenbanksysteme II*, BI, Mannheim, 1976.

/Win84/ Winston, P.H.: *Artificial Intelligence*, Addison-Wesley, Reading, 1984.

/Yau75/ Yau, S.S.; Fung, H.S.: *Associative Processor Architecture - A Survey -*, Sagamore Computer Conf. Parallel Processing, IEEE Computer Soc., 1975, 1-14.

## **Anhang A Erläuterung der Anwendung bildgestützter Navigation**

Zur Demonstration der der BPI-Architektur zu Grunde gelegten Bildanalyse-Aufgabenstellung sowie zur Diskussion spezieller Aspekte der Analyse mit Produktionsnetzen werden anschließend einige, 1986 in einem internen FIM-Bericht dargestellte Analyseergebnisse für die Anwendung bildgestützter Navigation erläutert. Eine kurze Zusammenfassung gibt /Lüt86/. Weitere Anwendungen, die mit Produktionsnetzen realisiert wurden, sind in /Füg92/, /Sti93/, /Füg94/, /Sti94/, /Bäh95/, /Sti95a/, /Sti95b/, /Sti96a/, /Sti96b/, /Sti97a/, /Sti97b/, /Sti97c/, /Mich98a/, /Mich98b/, /Schä98/, /Schw98/ und /Sti98/ veröffentlicht.

Bei der bildgestützten Navigation /Ter81, Tel79/ von Flugkörpern wird die Flugroute so geplant, dass in vorgegebenen Abständen Objekte überflogen werden, an denen sich der Flugkörper orientiert. Während der Flugmission werden diese Objekte mit einem senkrecht nach unten blickenden Sensor aufgenommen. Die Flugroute und die Aufnahmezeitpunkte sind so gewählt, dass die erwarteten Referenzobjekte bei ungestörtem Flug im Zentrum der jeweiligen Aufnahmen liegen. Kursabweichungen führen zu Positionsabweichungen in der Aufnahme, aus denen Maßnahmen zur Korrektur des Flugkurses abgeleitet werden. Aufgabe der automatischen Bildanalyse ist also, die festgelegten Referenzobjekte in den Missionsbildern wiederzufinden und zu positionieren.

Bei dieser Aufgabenstellung wird davon ausgegangen, dass symbolische Objektbeschreibungen, die z. B. aus Karten extrahiert werden, vorliegen. Bilder mit ihrer Grauwertinformation sollen nicht benötigt werden. Außerdem sollte das eingesetzte Analyseverfahren weitgehend unabhängig von Vegetations- und Beleuchtungseinflüssen sein. Perspektivische Verzerrungen, die durch Orientierungs- und Positionsfehler des Sensors entstehen, sowie fehlerhafte Objektbeschreibungen sollten ebenfalls bis zu einem gewissen Grad toleriert werden. Ein direkter Bildvergleich, wie er z. B. bei Korrelationsverfahren /Ger79, Ber80/ durchgeführt wird, scheidet daher aus.

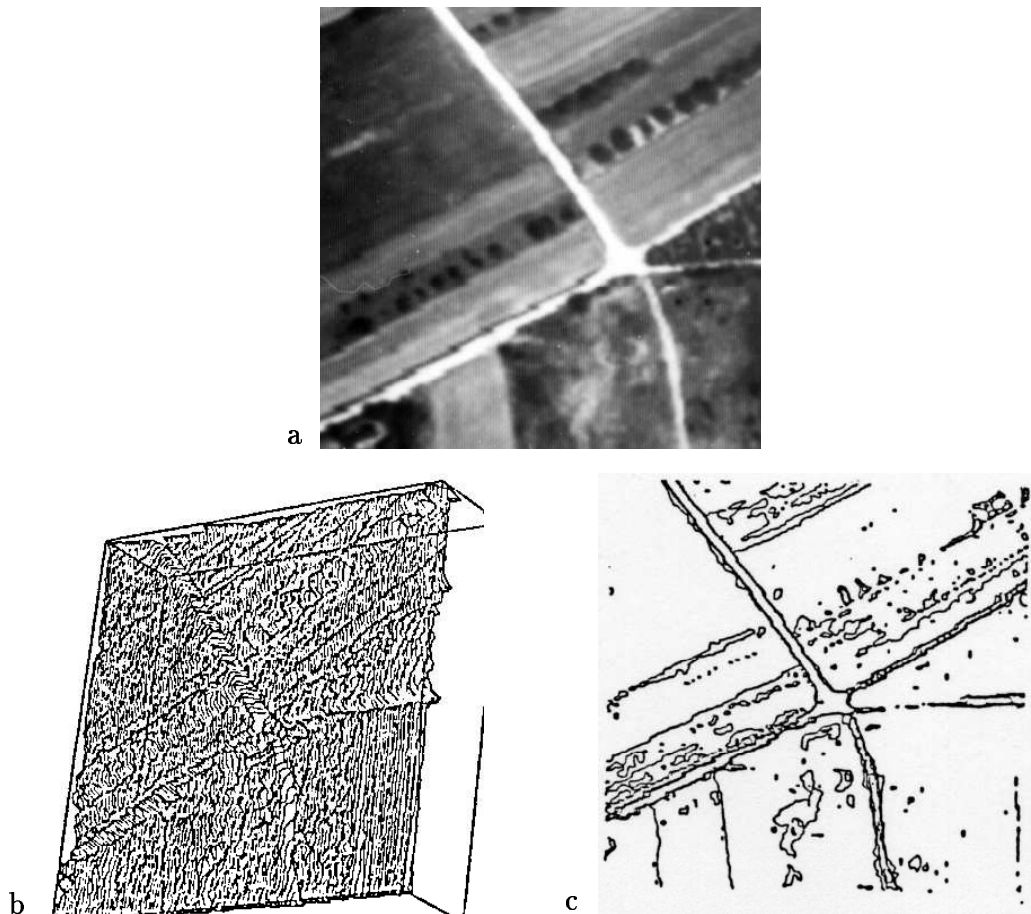
### **A.1 Bildmaterial und Vorverarbeitung der Bilddaten**

Die im Abschnitt A.3 gezeigten Ergebnisse wurden mit Datenmaterial erzielt, das im Rahmen des Flugzeugmessprogramms (FMP) /Alb78/ von der DLR erfohlen wurde.

Die 128 \* 128 Bildpunkte großen Teilbilder mit den Referenzobjekten wurden auf 512 \* 512 Bildpunkte vergrößert und anschließend mit einer 3 \* 3 Untermatrix tiefpassgefiltert (Abbildung A-3). Durch die Bildvergrößerung mit anschließender Tiefpassfilterung sind die kleinsten Objektdetails ungefähr 4 \* 4 Bildpunkte groß, so dass Bildrausereffekte von den Programmen zur Extraktion der Primitivobjekte vernachlässigt werden können.

## A.2 Extraktion der Primitivobjekte

Die für die beschriebene Anwendung eingesetzte Vorverarbeitung /Lüt86/ generiert aus dem zu analysierenden Grauwertbild Primitivobjekte, die von der sich anschließenden symbolischen Verarbeitung zusammengesetzt werden. Das Grauwertgebirge des Bildes (Abbildung A-1b) wird zuerst durch Höhenlinien approximiert (Abbildung A-1c), die die Äquidensiten des Bildes sind. Diese werden dann in einem zweiten Schritt durch Geradenstücke, Kreissegmente und Ecken approximiert (Abbildung A-2). Das Grauwertbild wird so in eine symbolische Beschreibung überführt, aus der eine Approximation des ursprünglichen Grauwertbildes generiert werden kann (Abbildung A-3).



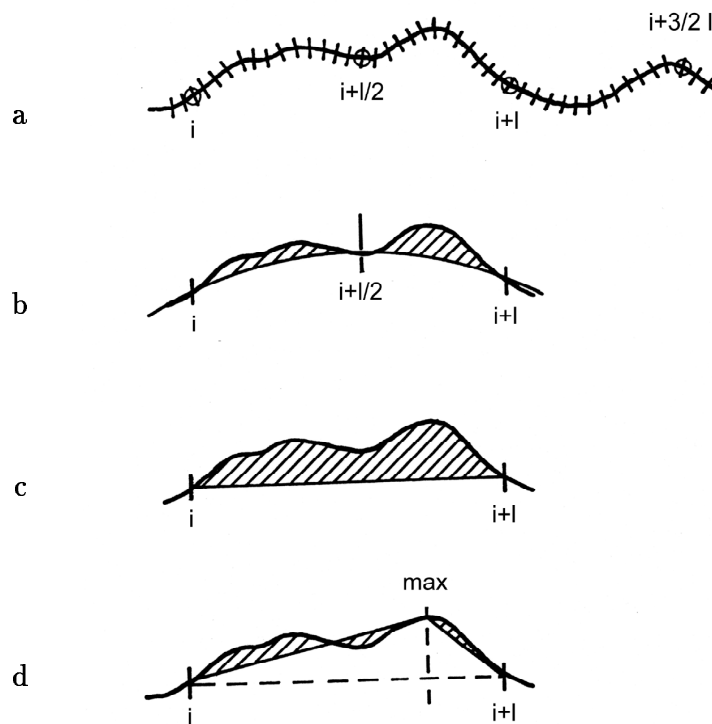
**Abb. A-1:** Darstellung von Grauwertbildern

- a) mit Grauwerten dargestellt
- b) als Grauwertgebirge dargestellt
- c) mit Höhenlinien dargestellt

Das eingesetzte Extraktionsverfahren baut auf dem Mehrschwellenverfahren (level slicing) /ATH89/ auf, das die bei der Binärisierung entstehenden Schnittflä-

chen unmittelbar beschreibt und auswertet.

Zur Generierung der Höhenlinien werden Binärisierungsschwellen vorgegeben. Die Anzahl der Schnitte  $n$  ist ein Verfahrensparameter. Seine Größe kann aufgrund der Darstellung des zu findenden Referenzobjektes im rekonstruierten Bild (Abbildung A-3d) abgeschätzt werden. Üblich sind 4 bis 30 Schnitte.

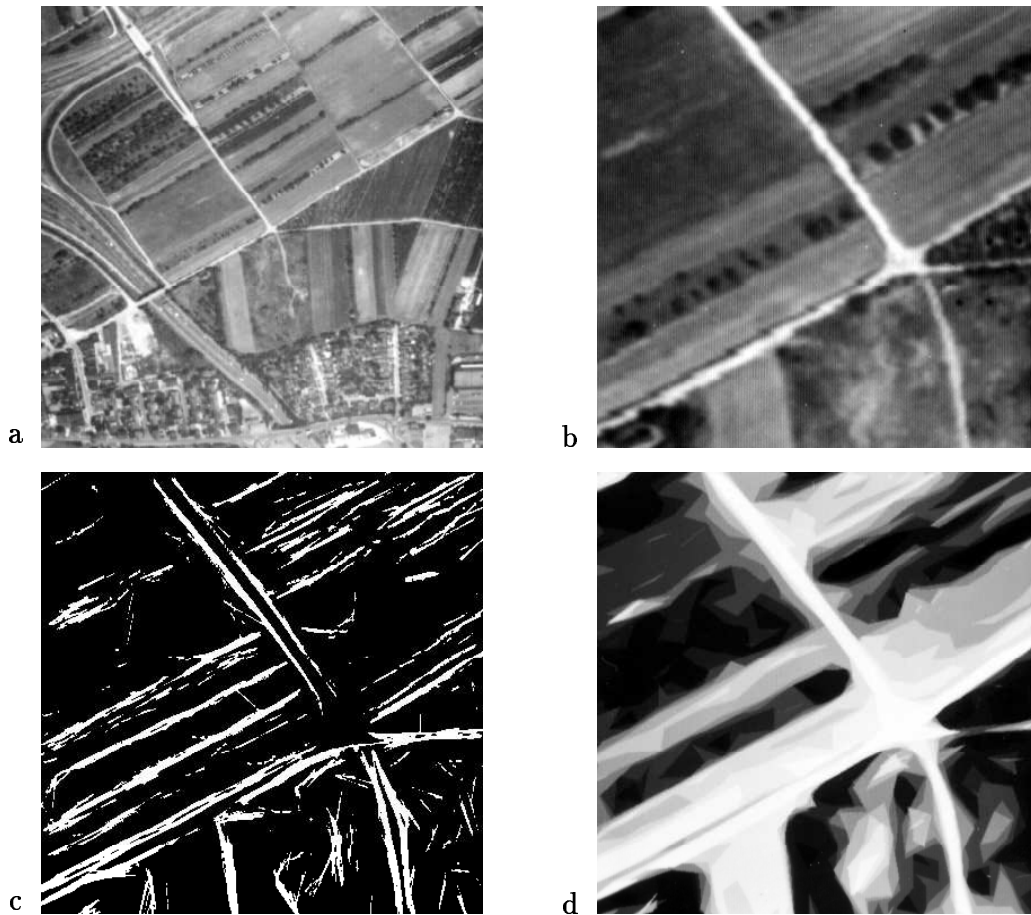


**Abb. A-2:** Generierung der Primitivobjekte  
 a) zu approximierende Höhenlinie  
 b) Approximierung durch Kreissegment  
 c) Approximierung durch Geradenstück  
 d) Approximierung durch Ecke

Die Höhenlinien begrenzen durch Binärisierung entstandene Schnittflächen. Diese Randkurven werden durch Primitivobjekte approximiert. Für die im Abschnitt A.3 dargestellten Ergebnisse werden Geradenstücke und Kreissegmente als Primitivobjekte verwendet. Dazu wird ein Parameter *Approximationslänge*  $l$  definiert, der angibt, wieviele Randelemente von einem Primitivobjekt überdeckt werden sollen. Primitivobjekte überlappen sich auf der Höhenlinie zu 50% (Abbildung A-2).

Das approximierende Geradenstück ist durch den ersten und letzten überdeckten Punkt des zu approximierenden Höhenlinienstückes definiert. Diese Punkte sind die Endpunkte des Geradenstückes. Das approximierende Kreissegment wird

durch den ersten, den mittleren und den letzten Punkt definiert. Die approximierende Ecke ist definiert durch den ersten und letzten Punkt und durch denjenigen Punkt der Höhenlinie, der den größten Abstand zum approximierenden Geradenstück hat. So entstehen für jeden Abschnitt drei Primitivobjekte, die grundsätzlich alle an die symbolische Verarbeitung weitergegeben werden (Abbildung A-2).



**Abb. A-3:** Vorverarbeitetes Grauwertbild mit Wegkreuzung

- a) 512 \* 512 Bildausschnitt
- b) vergrößerter 128 \* 128 Bildausschnitt
- c) generierte Geradenstücke
- d) aus Geradenstücken rekonstruiertes Bild

Die aus den Geradenstücken rekonstruierten Grauwertbilder liefern einen Eindruck von der durch die zweistufige Approximierung entstehenden Verfälschung. In Abbildung A-3c wurden die Höhenlinien durch Geradenstücke approximiert. Anschließend wurden die von den approximierten Höhenlinien begrenzten Flächen mit ihrem jeweiligen Grauwert geflutet und in das Bild eingetragen (Abbildung A-3d). Zuerst wurden die Flächen mit kleinstem Grauwert eingetragen, dann wurden in der Reihenfolge ihres Grauwertes die anderen Flächen darüber geschrieben.



Die Approximationslänge  $l$  legt die kleinsten von den Approximationsprogrammen noch erfassten Bilddetails fest. Sie kann als ein Maß für die *geometrische Auflösung* und die Anzahl  $n$  der Binärisierungsschwellen als ein Maß für die *radiometrische Auflösung* der Analyse interpretiert werden.

Sind die wiederzufindenden Referenzobjekte im rekonstruierten Bild nicht mehr deutlich ausgeprägt, dann ist die Vorverarbeitung zu optimieren. Dazu kann z. B. die Anzahl der Schnitte erhöht, die Länge der Primitivobjekte verkleinert oder vergrößert, oder es können andere Primitivobjekte verwendet werden. Dabei lässt sich das auszuwertende Grauwertbild mit entsprechendem Aufwand grundsätzlich beliebig genau approximieren. Mit z. B. 255 Schnittebenen und einer Approximationslänge  $l$  von vier wird ein Bild mit  $64 * 64$  Abtastwerten und 8 Bit Auflösung, das auf  $512 * 512$  Bildpunkte vergrößert wurde, symbolisch so detailliert beschrieben, dass das rekonstruierte Bild weitgehend mit dem Ursprungsbild übereinstimmt.

Dieses sehr einfache Verfahren zur Extraktion der Primitivobjekte wurde 1986 gewählt, um die Analyse mit Produktionsnetzen umfassend zu untersuchen und den Einfluss der Vorverarbeitung zu minimieren. Für konkrete Anwendungen stehen heute komplexe Extraktionsverfahren zur Verfügung (z. B. /Steg98/), mit denen für viele Anwendungen bessere Ergebnisse erzielt werden als mit dem hier erläuterten Verfahren.

### A.3 Analyseergebnisse für die bildgestützte Navigation

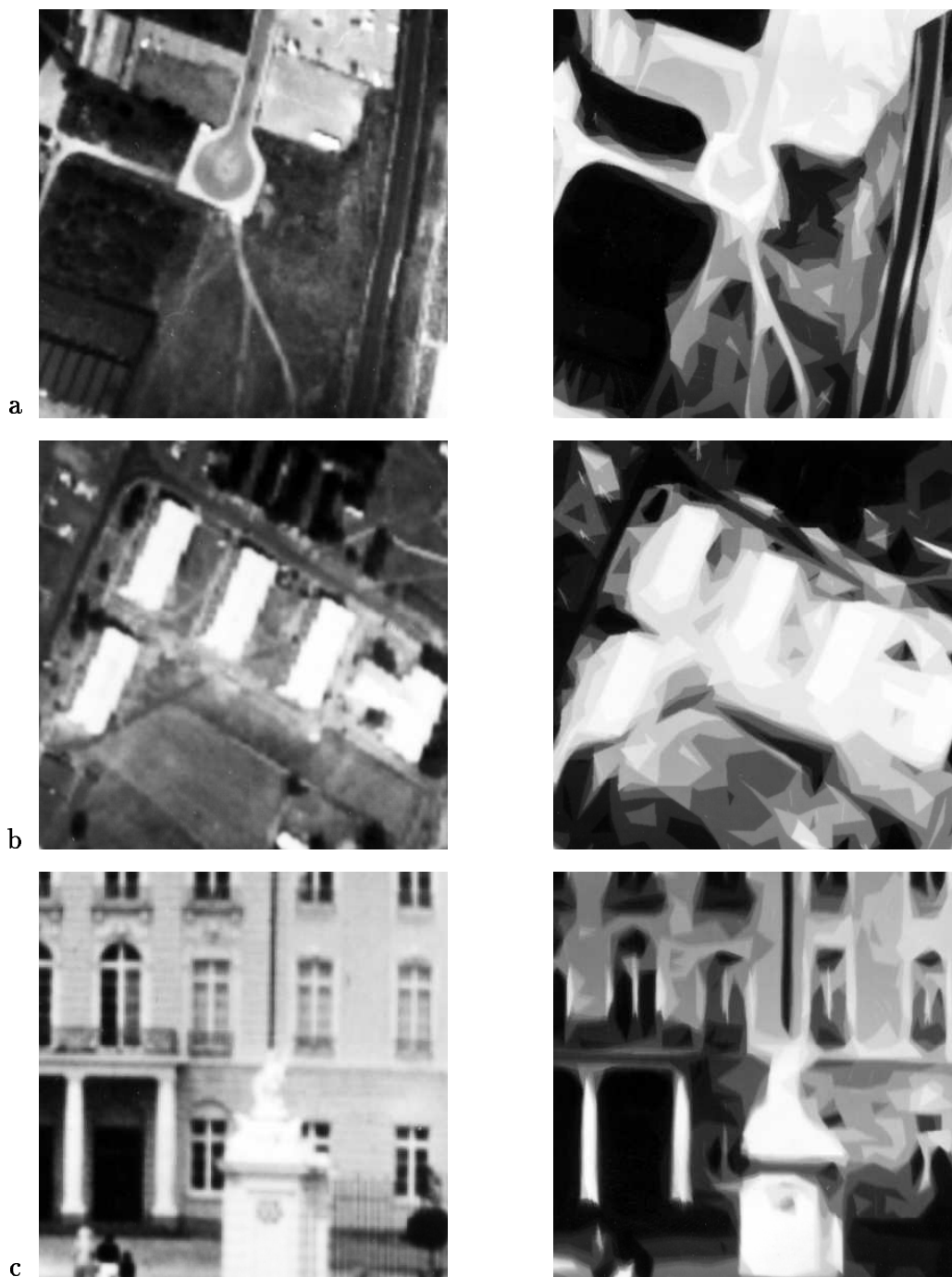
Insgesamt wurden für die Ergebnisse in /Lüt86/ 16 Bilder in 28 Verarbeitungsläufen bearbeitet. Davon sind fünf Bilder ungestört (Abbildung A-4 und A-5).

In den Bildern *Autobahnbrücke*, *Kreuzung*, *Wendehammer* und *Häuserformation* (Abbildung A-4 und A-5) wurde das im jeweiligen Bild enthaltene Referenzobjekt gesucht. Außerdem wurden alle vier Referenzobjekte im Schlossbild (Abbildung A-5c) gesucht. In Abbildung A-4 sind unter den Originalbildern die aus den Bildern extrahierten Geradenstücke, die *sehr gut* oder *gut* approximiert wurden, und die aus allen Geradenstücken rekonstruierten Bilder abgebildet. Die strukturelle Mustererkennung "sieht" nur die Geradenstücke, aus denen *Streifen*, *Straßen*, *Brücken*, *Häuser* und *Kreuzungen* sowie die rekonstruierten Bilder aufgebaut werden. Teilobjekte, die in den rekonstruierten Bildern nicht zu erkennen sind, können daher nicht aufgebaut werden. *Ecken* und *Kreissegmente* wurden bei der Rekonstruktion nicht berücksichtigt, so dass die Rekonstruktion des Wendehammerbildes keine Aussage über die für den Aufbau des *Wendehammers* verwendeten Kreissegmente gestattet. Die extrahierten Geradenstücke sind nur in der Abbildung A-4 dargestellt. In den anschließenden Abbildungen A-5 bis A-9 sind nur noch die Originalbilder und die Rekonstruktionen abgebildet.

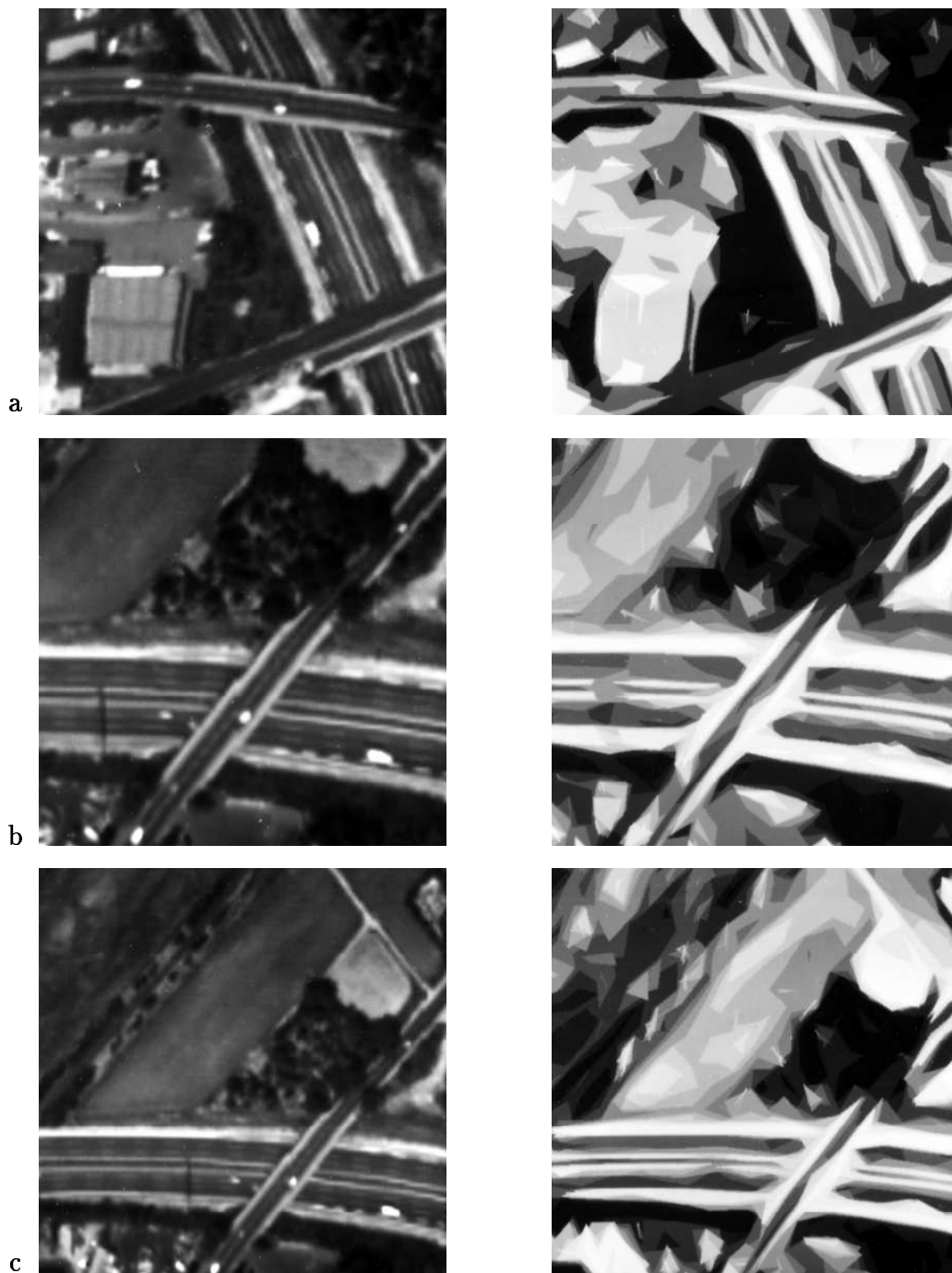
Um die Grenzen des Verfahrens aufzuzeigen, werden Ergebnisse für mehrere gestörte Bilder gezeigt. Das Brückenbild (Abbildung A-4a) wurde gedreht und verschoben, so dass eine zweite Brücke und weitere Objekte im auszuwertenden Bild enthalten sind. Es wurde um 10% vergrößert, um 20% verkleinert und verrauscht. Außerdem wurden Wolkenschatten und Zeilenversatz (Jitter) simuliert.



**Abb. A-4: Ungestörte Bilder**  
a) Autobahnbrücke  
b) Wegkreuzung  
oben: Originalbilder  
Mitte: extrahierte Geradenstücke  
unten: rekonstruierte Bilder



**Abb. A-5:** Ungestörte Bilder  
a) Wendehammer  
b) Häuserformation  
c) Schlossbild  
links: Originalbilder  
rechts: rekonstruierte Bilder



**Abb. A-6:** Gestörte Bilder der Autobahnbrücke

a) gedreht und verschoben

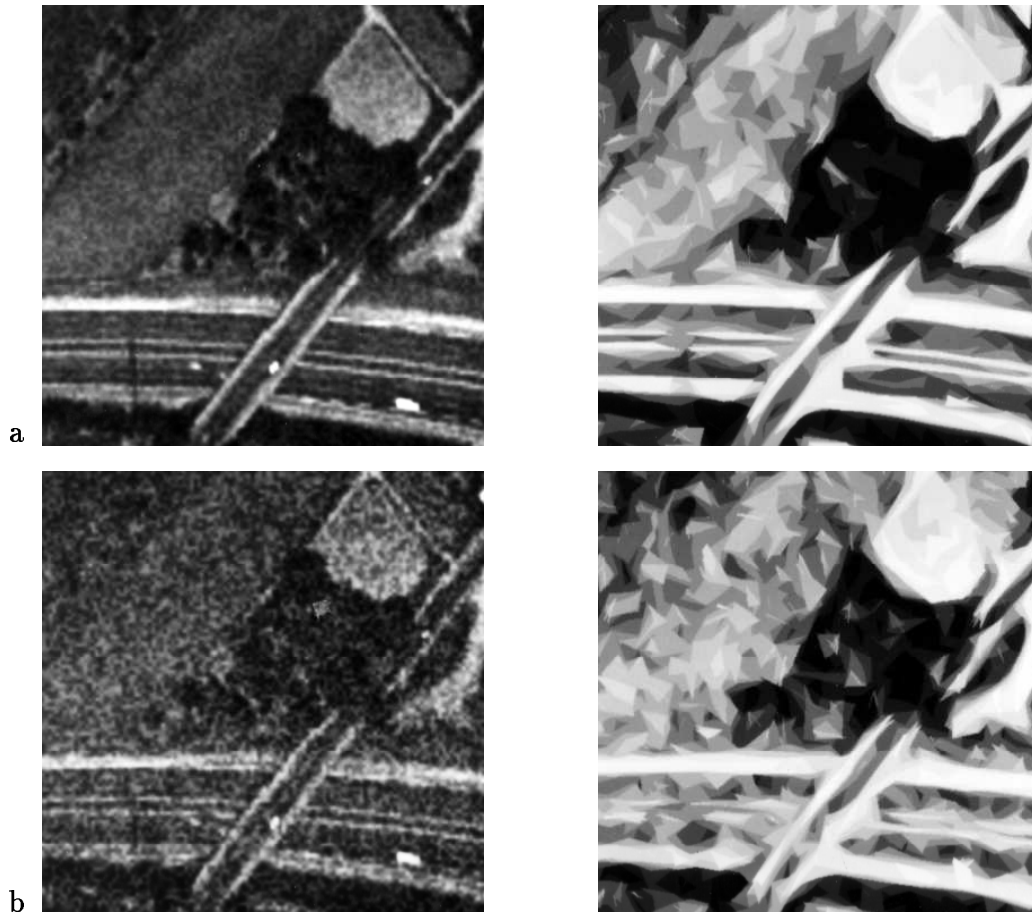
b) um 10% vergrößert

c) um 20% verkleinert

links: Originalbilder

rechts: rekonstruierte Bilder

In Abbildung A-7a betrug die Rauschamplitude des überlagerten gaußverteilten Rauschens 10 Grauwerte, in Abbildung A-7b 25 Grauwerte. Dabei nehmen die Bildpunkte Grauwerte zwischen 0 bis 255 an. In Abbildung A-8a wurde gaußverteilter Zeilenversatz mit einer Amplitude von 5 Bildpunkten simuliert. In Abbildung A-8b betrug die Amplitude 7 Bildpunkte. Eine Zeile überdeckt dabei 128 Bildpunkte.



**Abb. A-7:** Gestörte Bilder der Autobahnbrücke  
(Grauwerte von 0 bis 255)

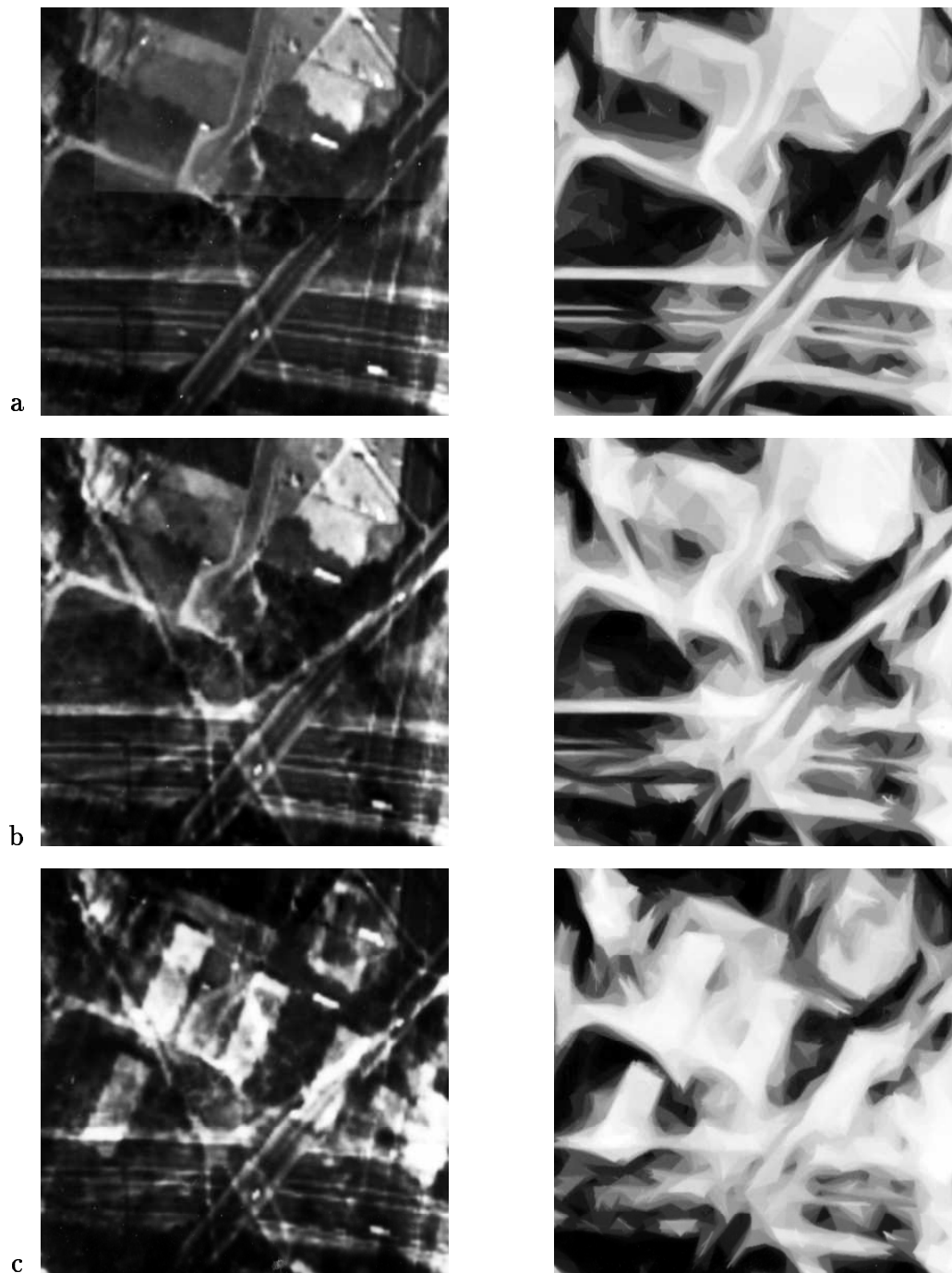
- a) verrauscht mit einer Amplitude von 10 Grauwerten
  - b) verrauscht mit einer Amplitude von 25 Grauwerten
- links: Originalbilder  
rechts: rekonstruierte Bilder

Neben systematisch gestörten Bildern wurden auch überlagerte Bilder bearbeitet. In Abbildung A-9a wurden das Brückenbild und das Wendehammerbild multiplikativ überlagert, so dass die Grauwerte der Punkte des überlagerten Bildes das geometrische Mittel der Grauwerte der entsprechenden Punkte der überlagerten Bilder sind. In Abbildung A-9b wurden das Brückenbild, das Wendehammerbild



**Abb. A-8:** Gestörte Bilder der Autobahnbrücke  
(128 Bildpunkte in einer Zeile)

- a) Zeilenversatz mit einer Amplitude von 5 Bildpunkten
  - b) Zeilenversatz mit einer Amplitude von 7 Bildpunkten
  - c) mit überlagertem simulierten Wolkenschatten
- links: Originalbilder  
rechts: rekonstruierte Bilder



**Abb. A-9:** Überlagerte Bilder  
a) Autobahnbrücke und Wendehammer  
b) Autobahnbrücke, Wendehammer und Kreuzung  
c) Überlagerung aller vier Bilder  
links: Originalbilder  
rechts: rekonstruierte Bilder

und das Kreuzungsbild überlagert. Abbildung A-9c zeigt ein vierfach überlagertes Bild mit allen Referenzobjekten.

### A.3.1 Erläuterung der Verarbeitung

Nach der Vorverarbeitung werden die Bilder analysiert, indem die generierten Primitivobjekte zu komplexen Teilobjekten und den Referenzobjekten zusammengesetzt werden (vgl. Abschnitt 1.4).

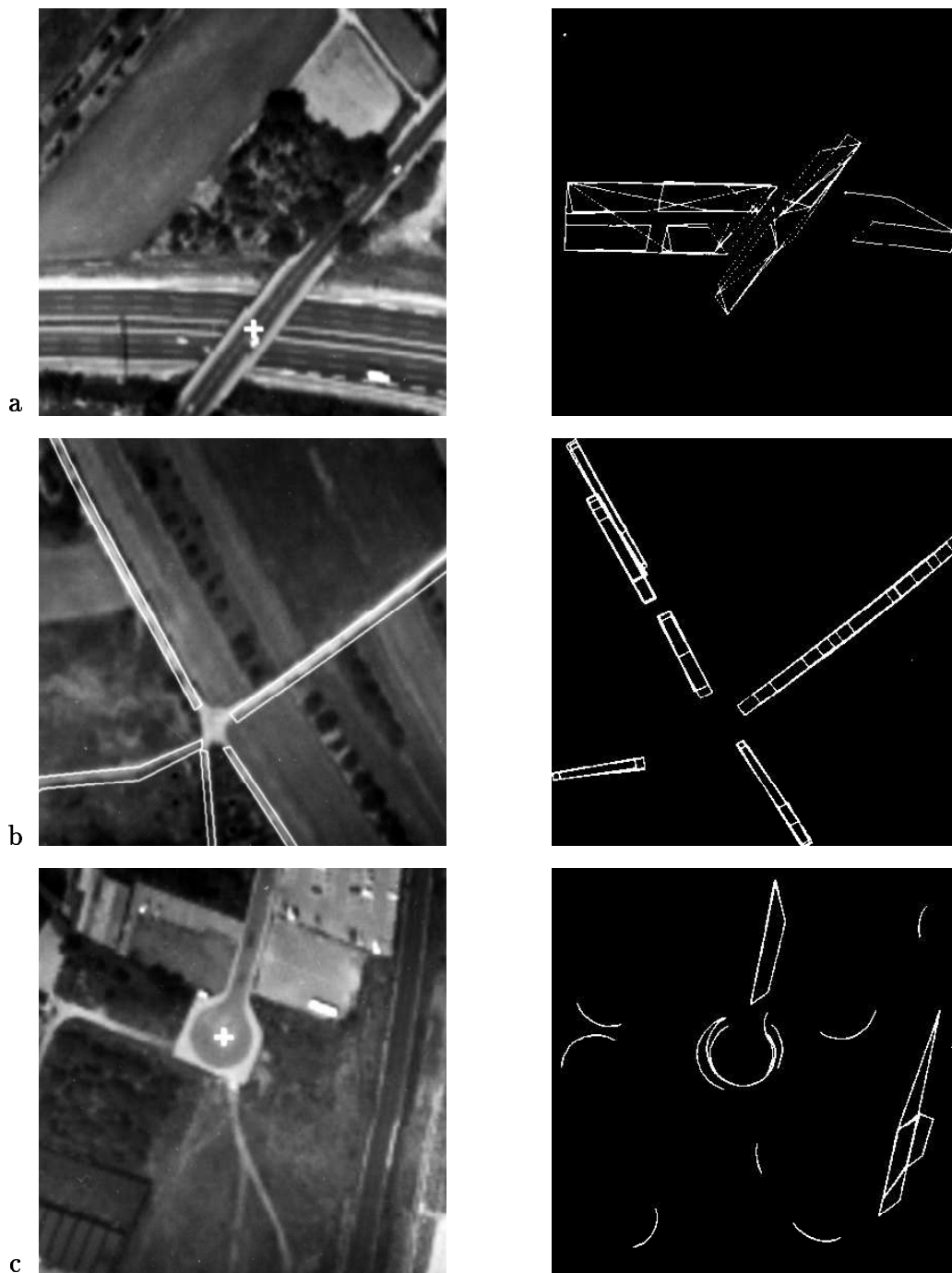
Die hierfür eingesetzten Überprüfungsprogramme werden mit Parametern auf spezielle Teilobjekte eingestellt. So wird beispielsweise das Überprüfungsprogramm für Streifen auf die Breite und die minimale Länge der erwarteten Streifen eingestellt, um z. B. schmale Streifen für den Aufbau des Wendehammers von breiten Streifen für den Aufbau der Autobahn zu unterscheiden. Für jedes wiederzufindende Referenzobjekt wurde eine Systemkonfiguration, die eine Menge von Assoziationen definiert, mit genau einem Parametersatz eingesetzt. Dieser Parametersatz wurde bei allen Bildanalysen mit der zugehörigen Systemkonfiguration unverändert eingesetzt. Variiert wurde nur die Anzahl der Binärisierungsschnitte, mit der die Vorverarbeitungsprogramme das Grauwertgebirge schneiden. Bei den nicht überlagerten Bildern wurde das Grauwertgebirge 10-mal geschnitten, das Bild mit zwei überlagerten Ausgangsbildern wurde 15-mal geschnitten, das dreifach überlagerte 20-mal und das vierfach überlagerte 30-mal. So sollte erreicht werden, dass die von den Ausgangsbildern im überlagerten Bild verursachte Grauwertmodulation mit einer ausreichenden Anzahl von Schnitten erfasst wird.

In den anschließenden Abbildungen sind die Ergebnisse der einzelnen Identifikationsläufe dargestellt, jeder Lauf mit zwei Bildern. Im jeweils oberen Bild ist die dem wiederzufindenden Referenzobjekt zugeordnete Position eingetragen, im unteren die das Objekt aufbauenden Streifen und Kreissegmente. Die anderen Teilobjekte, die ebenfalls zum Aufbau des Referenzobjektes verwendet wurden, sind nicht dargestellt.

In den Bildunterschriften sind die für die strukturelle Analyse benötigte Rechenzeit bei Einsatz eines 1 Mlps Rechners vom Typ VAX 11/780 in m Minuten und s Sekunden ( $\text{Zeit}=\text{m}:\text{s}$ ) sowie eine heuristisch definierte Bewertung ( $\text{Güte}=\text{g}$ ) angegeben, mit  $1 \leq g \leq 0$ . Diese zwei Angaben beschreiben das Verarbeitungsergebnis und erlauben, sichere Zuordnungen von eher spekulativen zu unterscheiden. Die für die Vorverarbeitung benötigte Rechenzeit ist für die Bewertung der Ergebnisse zweitrangig, sie lag je Bild zwischen 30 Sekunden und 3 Minuten.

In den überlagerten Bildern wurden jeweils alle vier Referenzobjekte gesucht, um so Referenzobjekte in durch Bildüberlagerung gestörten Bildern zu suchen und um zu testen, wie groß die Gefahr ist, nicht im Bild vorhandene Referenzobjekte in gestörten Bildern fälschlicherweise zuzuordnen.





**Abb. A-10:** Verarbeitungsergebnisse mit ungestörten Bildern

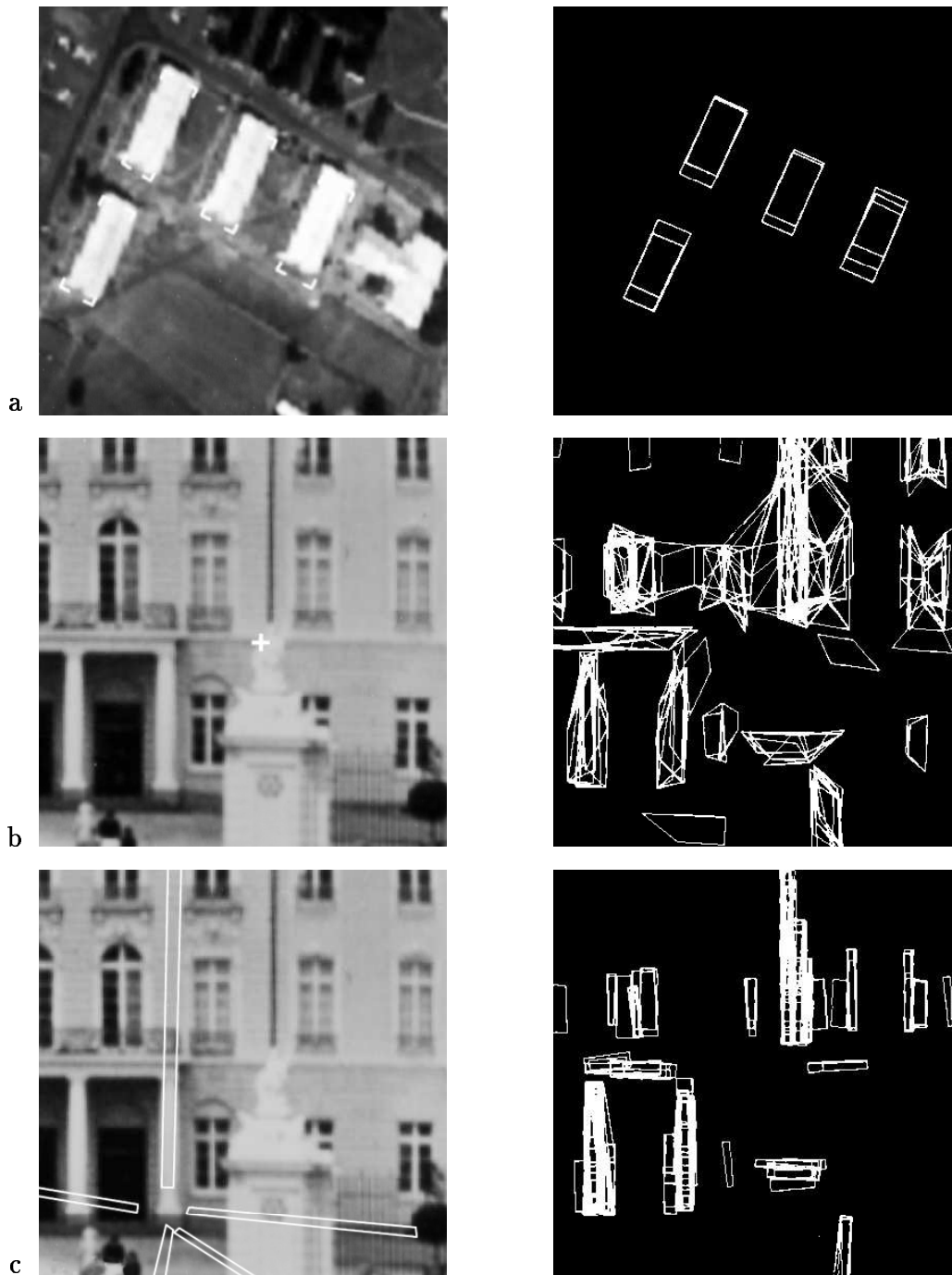
a) Autobahnbrücke (Güte=0.51, Zeit=1:54)

b) Wegkreuzung (Güte=0.51, Zeit=1:35)

c) Wendehammer (Güte=0.78, Zeit=1:46)

links: Originalbilder mit zugeordnetem Referenzobjekt

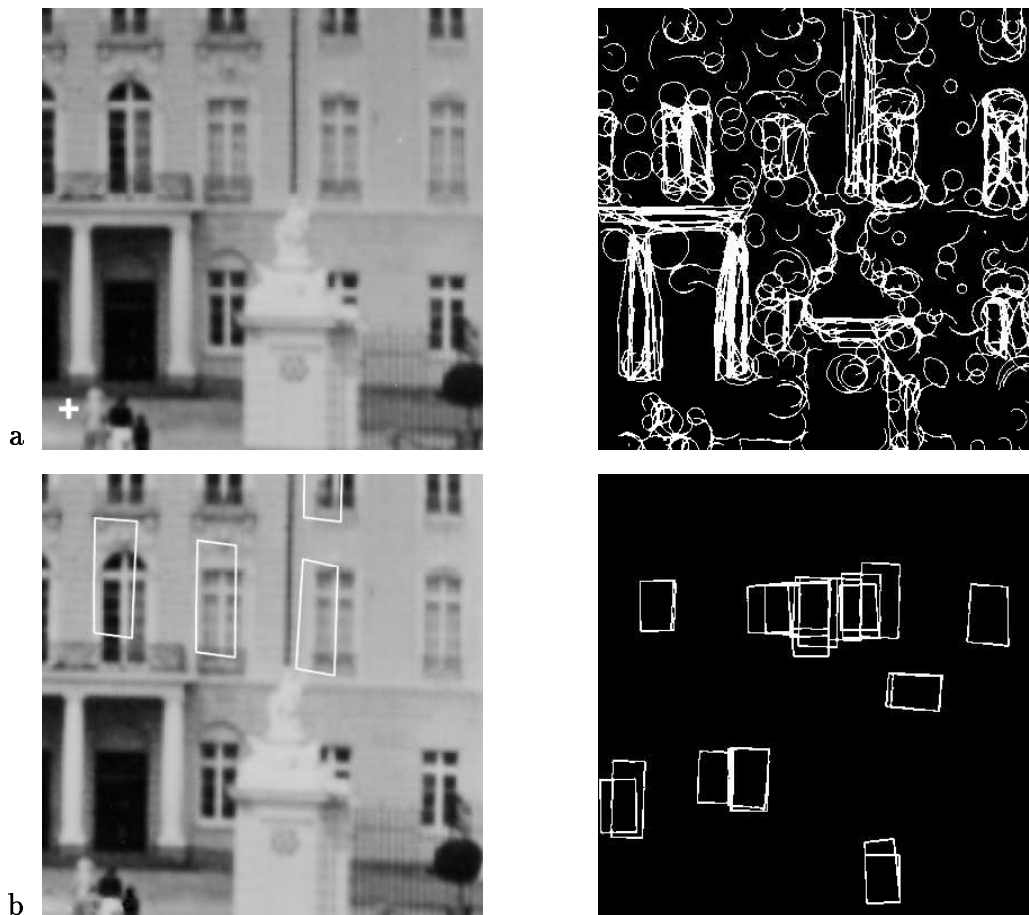
rechts: Zwischenergebnisse



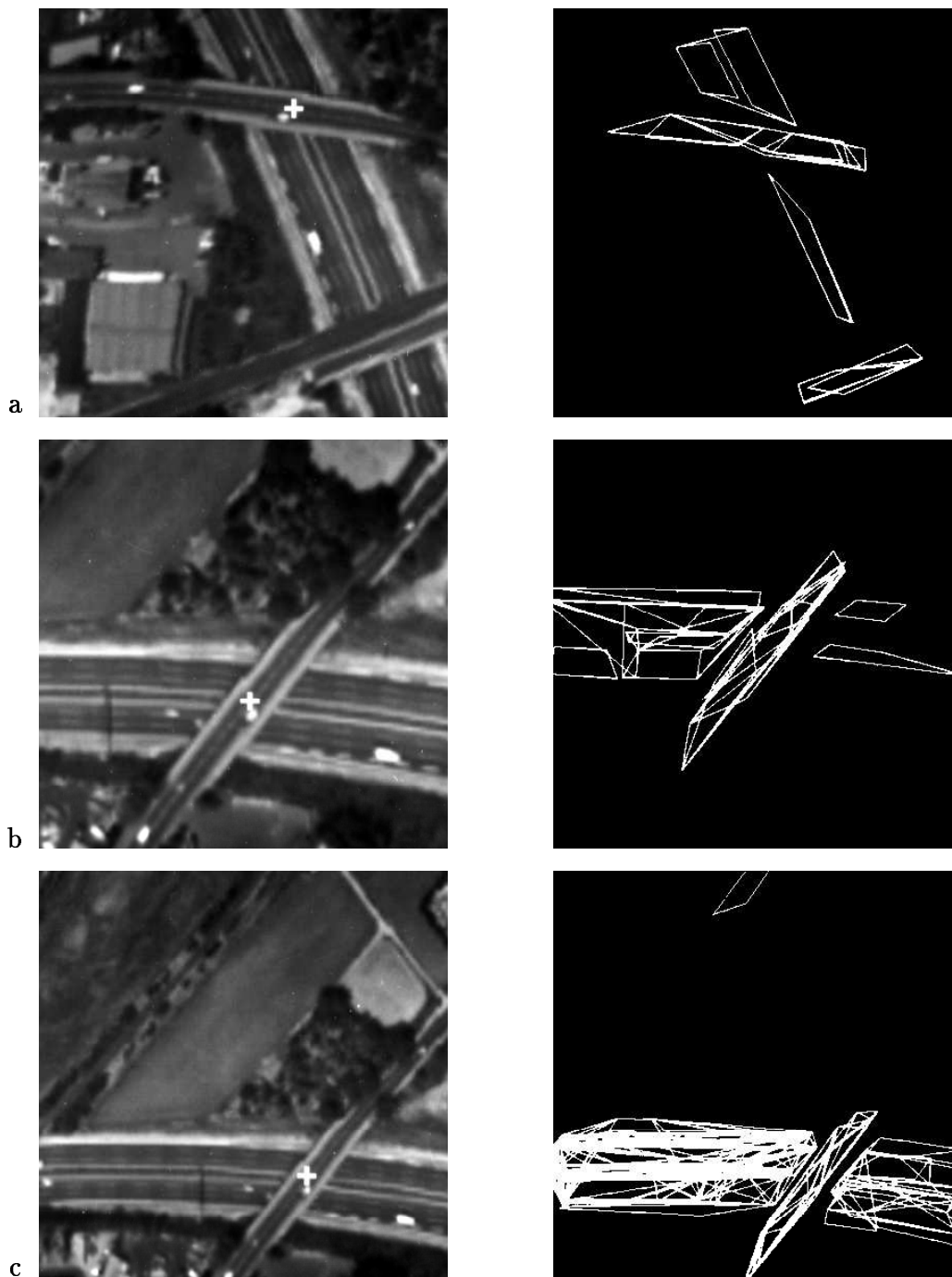
**Abb. A-11:** Verarbeitungsergebnisse mit ungestörten Bildern

- a) Häuserformation (Güte=0.76, Zeit=0:59)
- b) Schlossbild, in dem die Autobahnbrücke gesucht wurde (Güte=0.32, Zeit=28:33)
- c) Schlossbild, in dem die Wegkreuzung gesucht wurde (Güte=0.16, Zeit=13:02)

links: Originalbilder mit zugeordnetem Referenzobjekt  
rechts: Zwischenergebnisse



**Abb. A-12:** Verarbeitungsergebnisse mit ungestörten Bildern  
a) Schlossbild, in dem der Wendehammer gesucht wurde  
(Güte=0.27, Zeit=20:07)  
b) Schlossbild, in dem die Häuserformation gesucht wurde  
(Güte=0.33, Zeit=29:07)  
links: Originalbilder mit zugeordnetem Referenzobjekt  
rechts: Zwischenergebnisse



**Abb. A-13:** Verarbeitungsergebnisse mit gestörten Bildern der Autobahnbrücke

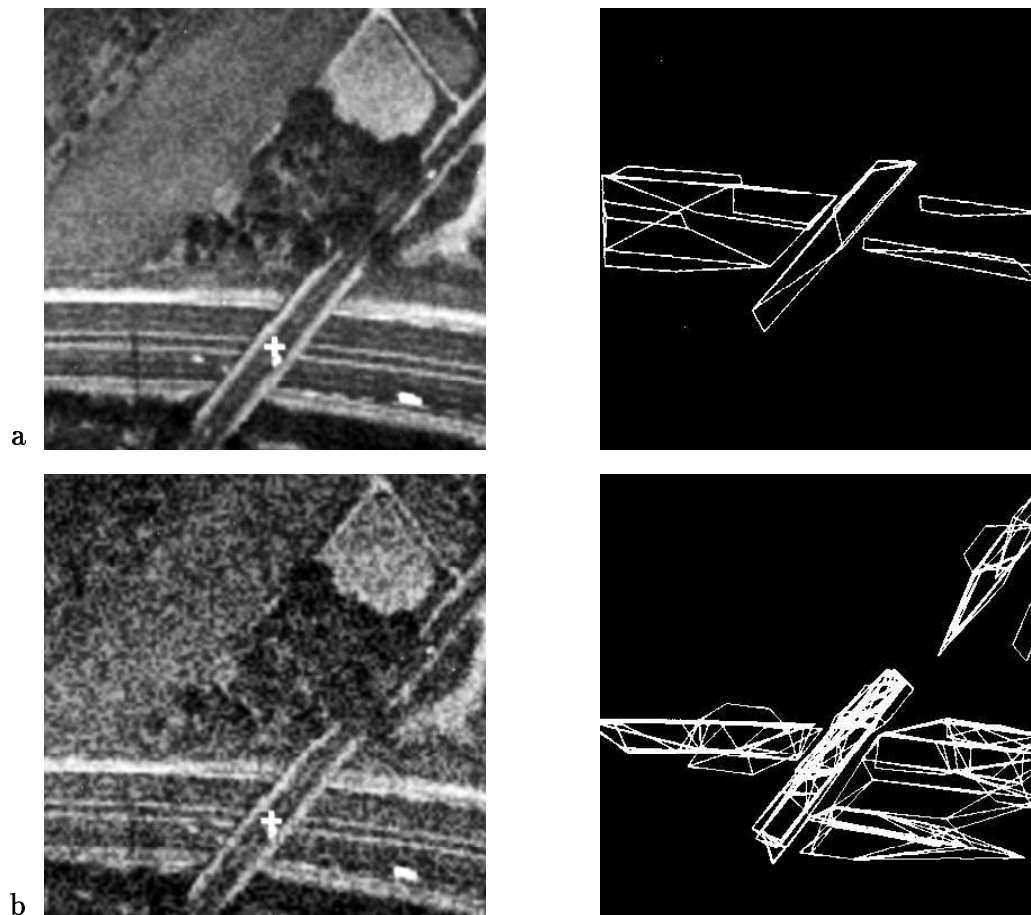
a) gedreht und verschoben (Güte=0.51, Zeit=3:06)

b) um 10% vergrößert (Güte=0.56, Zeit=2:27)

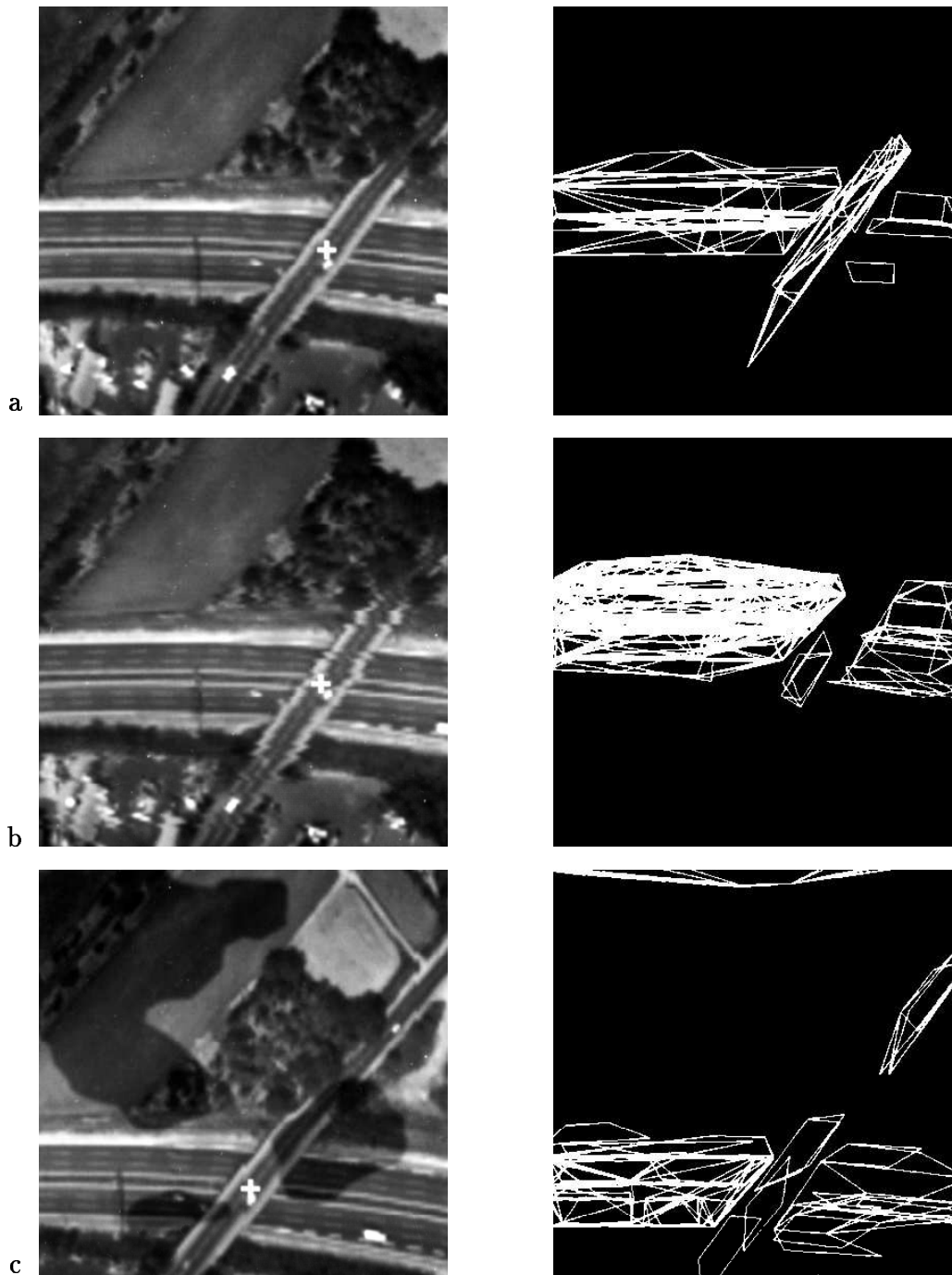
c) um 20% verkleinert (Güte=0.62, Zeit=5:57)

links: Originalbilder mit zugeordnetem Referenzobjekt

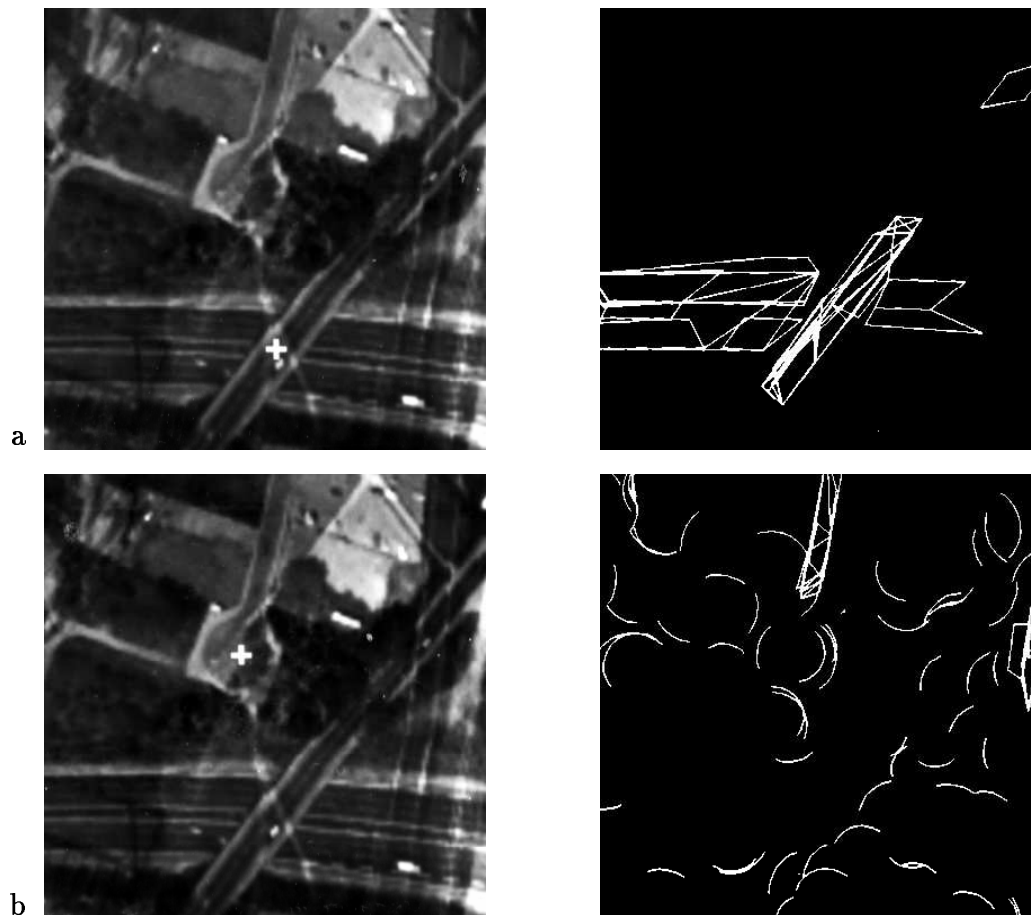
rechts: Zwischenergebnisse



**Abb. A-14:** Verarbeitungsergebnisse mit gestörten Bildern der Autobahnbrücke  
 a) verrauscht mit einer Amplitude von 10 Grauwerten  
 (Güte=0.51, Zeit=1:41)  
 b) verrauscht mit einer Amplitude von 25 Grauwerten  
 (Güte=0.52, Zeit=10:50)  
 links: Originalbilder mit zugeordnetem Referenzobjekt  
 rechts: Zwischenergebnisse



**Abb. A-15:** Verarbeitungsergebnisse mit gestörten Bildern der Autobahnbrücke  
 a) Zeilenversatz mit einer Amplitude von 5 Bildpunkten  
 (Güte=0.53, Zeit=3:41)  
 b) Zeilenversatz mit einer Amplitude von 7 Bildpunkten  
 (Güte=0.60, Zeit=6:26)  
 c) mit überlagertem simulierten Wolkenschatten (Güte=0.59, Zeit=3:40)  
 links: Originalbilder mit zugeordnetem Referenzobjekt  
 rechts: Zwischenergebnisse

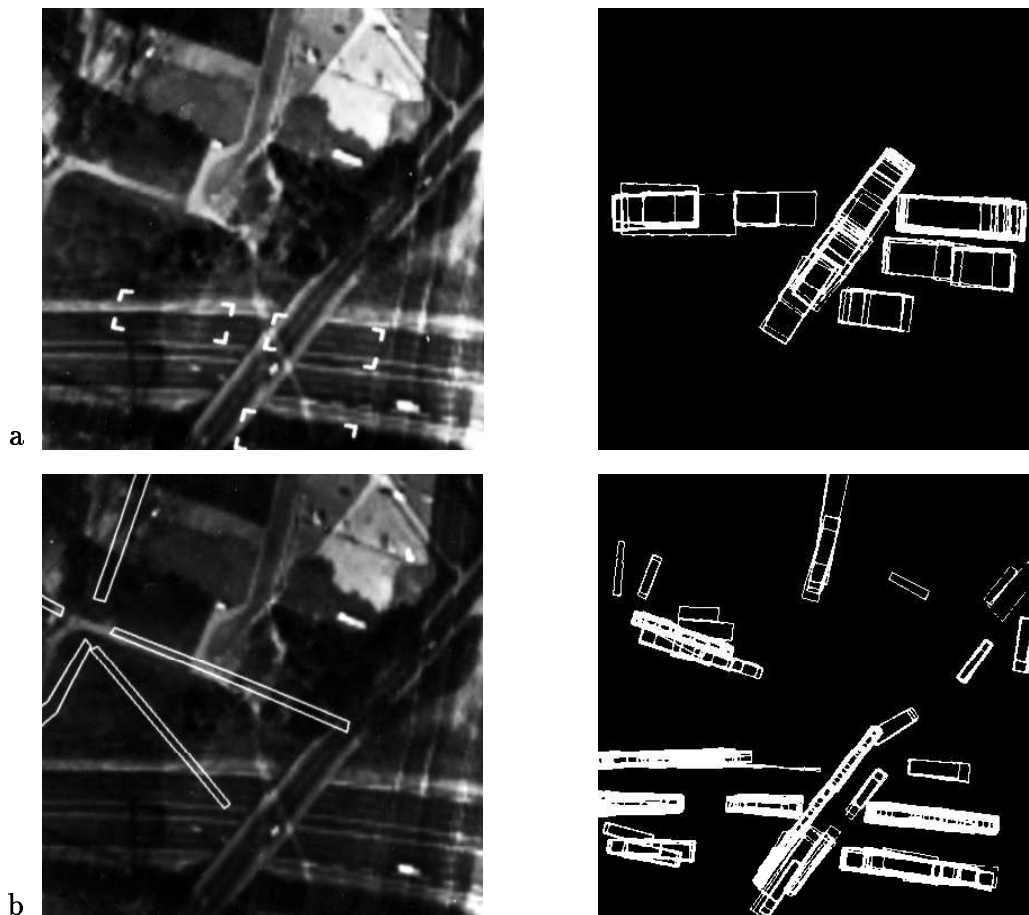


**Abb. A-16:** Verarbeitungsergebnisse mit zweifach überlagerten Bildern

a) Brücke in Überlagerung Brücke-Wendehammer  
(Güte=0.50, Zeit=2:43)

b) Wendehammer in Überlagerung Brücke-Wendehammer  
(Güte=0.57, Zeit=4:30)

links: Originalbilder mit zugeordnetem Referenzobjekt  
rechts: Zwischenergebnisse



**Abb. A-17:** Verarbeitungsergebnisse mit zweifach überlagerten Bildern

a) Häuserformation in Überlagerung Brücke-Wendehammer

(Güte=0.52, Zeit=15:04)

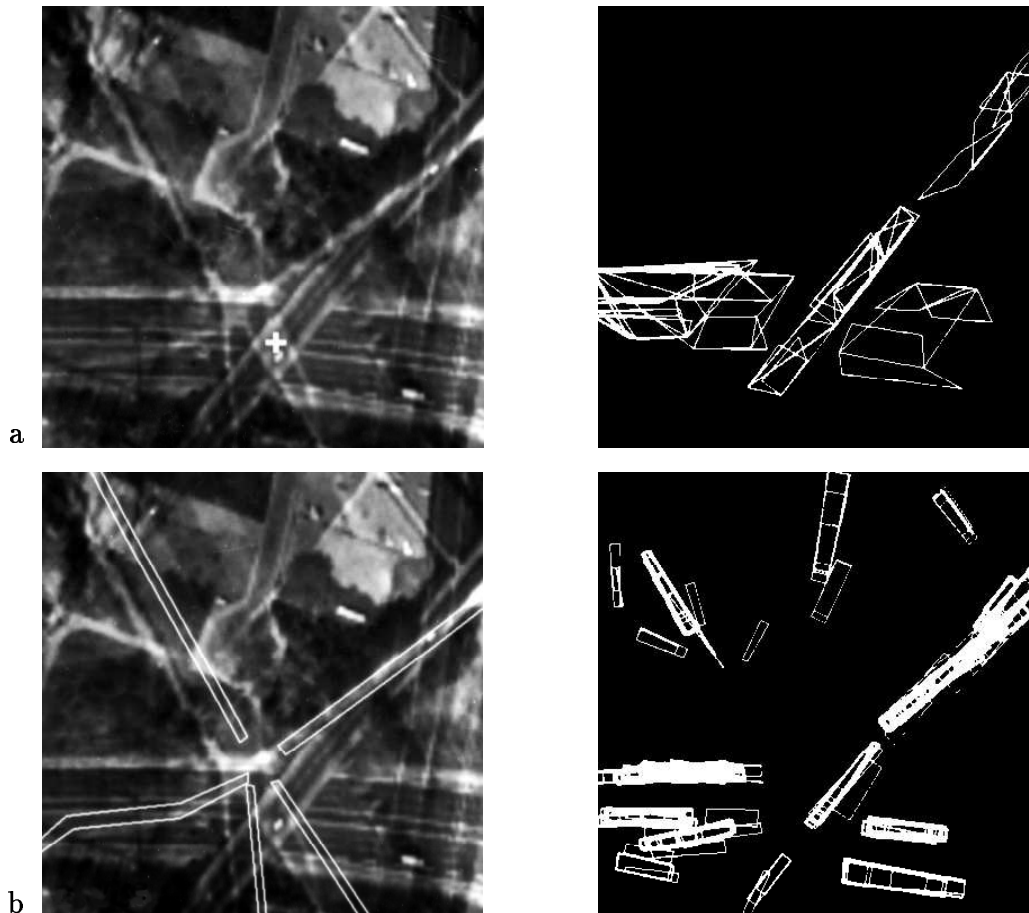
b) Kreuzung in Überlagerung Brücke-Wendehammer

(Güte=0.01, Zeit=30:30)

links: Originalbilder mit zugeordnetem Referenzobjekt

rechts: Zwischenergebnisse



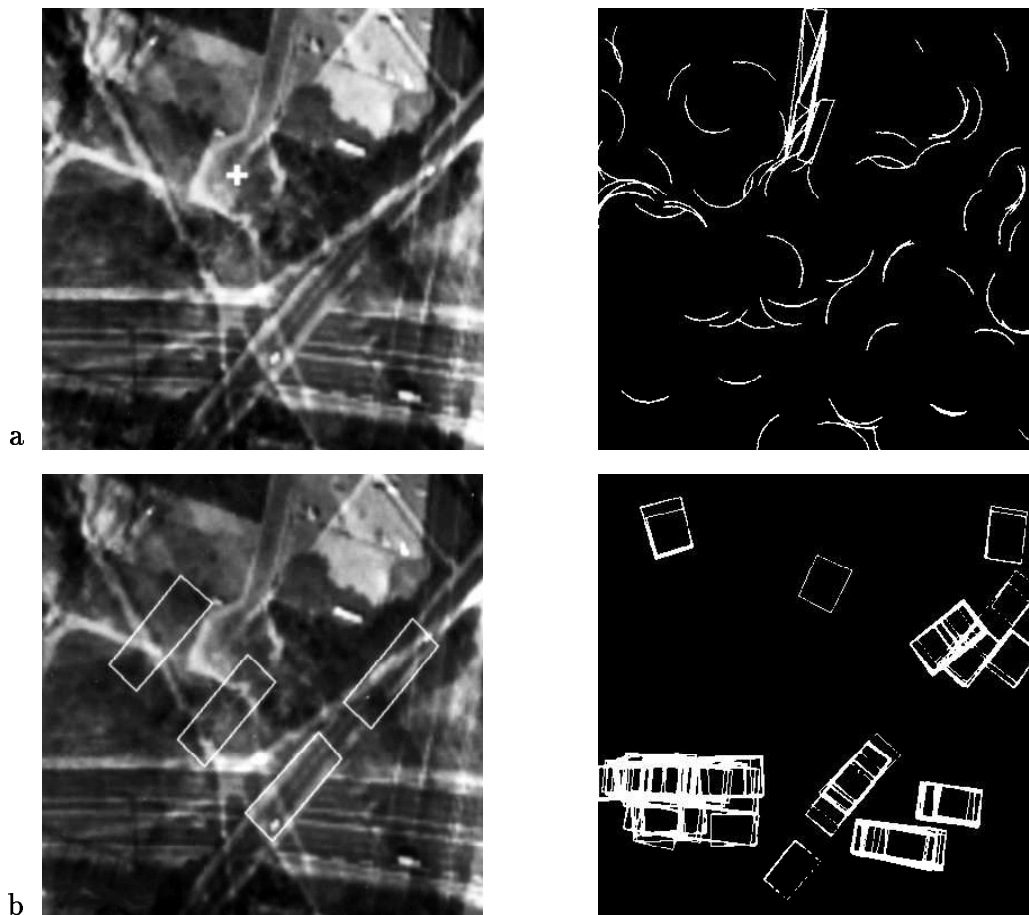


**Abb. A-18:** Verarbeitungsergebnisse mit dreifach überlagerten Bildern

a) Brücke in Überlagerung Brücke-Kreuzung-Wendehammer  
(Güte=0.72, Zeit=5:05)

b) Kreuzung in Überlagerung Brücke-Kreuzung-Wendehammer  
(Güte=0.24, Zeit=25:43)

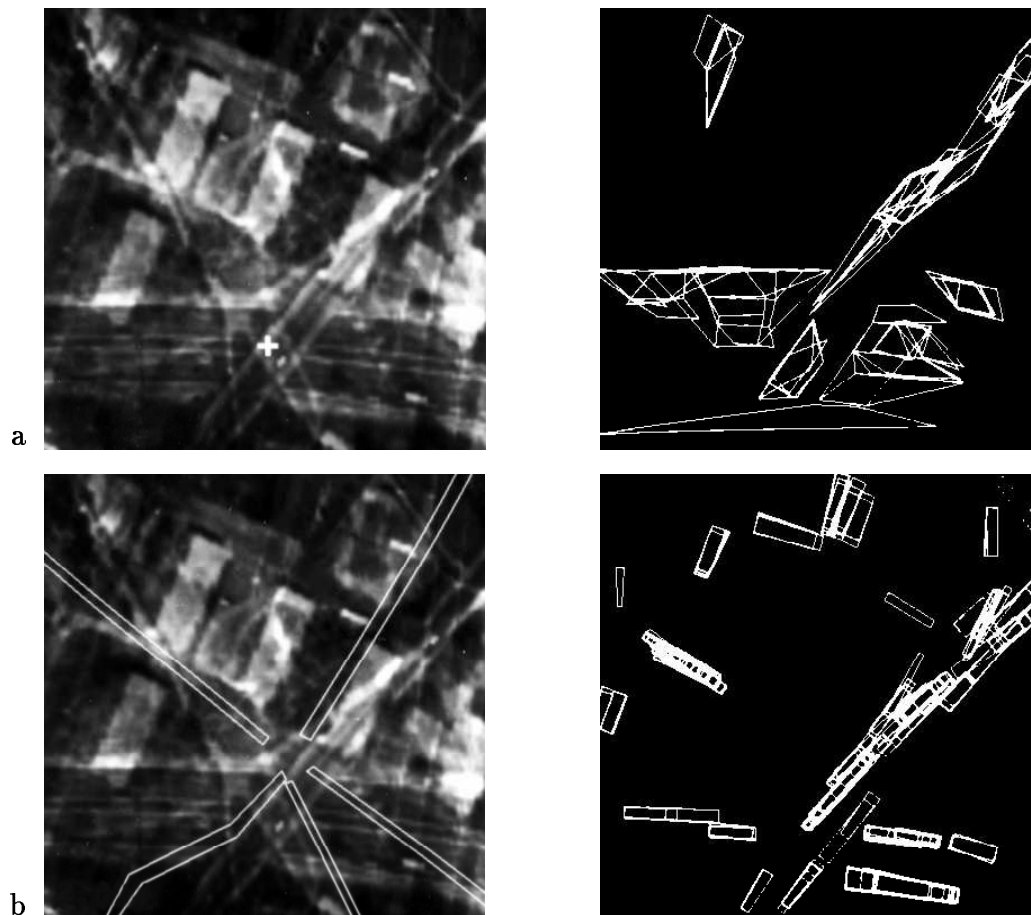
links: Originalbilder mit zugeordnetem Referenzobjekt  
rechts: Zwischenergebnisse



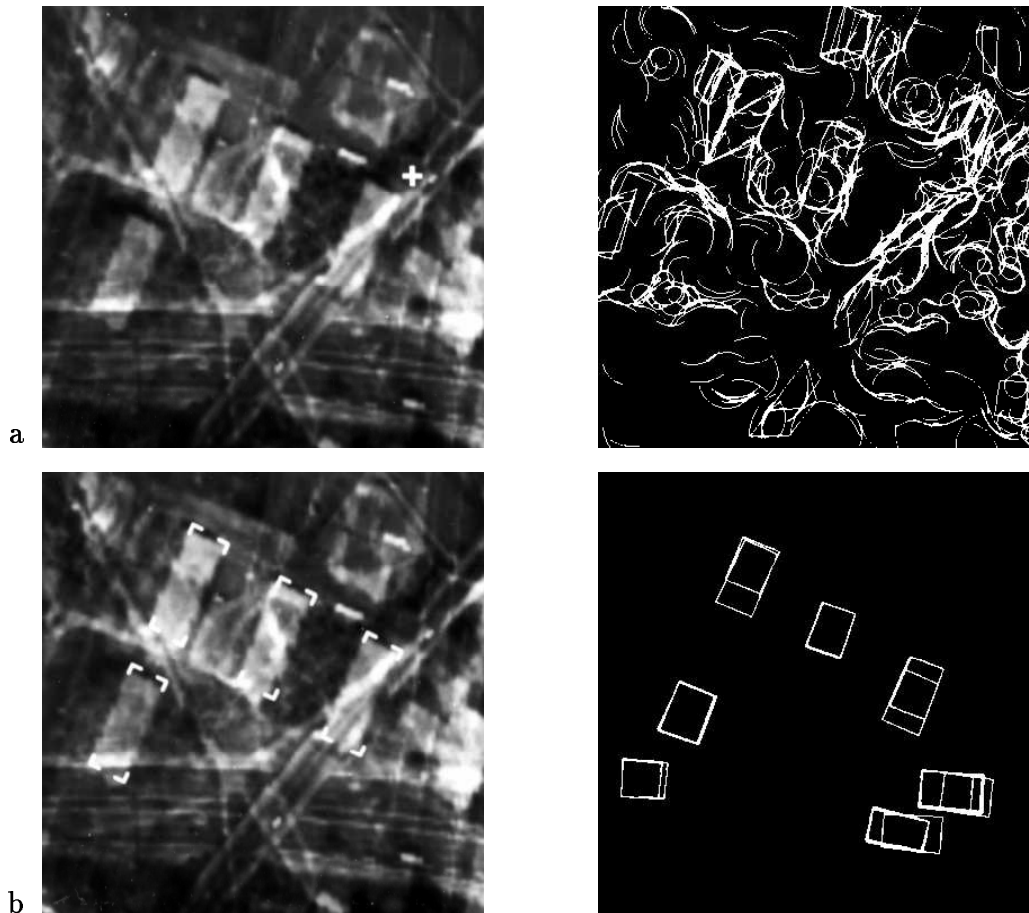
**Abb. A-19:** Verarbeitungsergebnisse mit dreifach überlagerten Bildern

- a) Wendehammer in Überlagerung Brücke-Kreuzung-Wendehammer  
(Güte=0.62, Zeit=3:43)
- b) Häuserformation in Überlagerung Brücke-Kreuzung-Wendehammer  
(Güte=0.34, Zeit=25:46)

links: Originalbilder mit zugeordnetem Referenzobjekt  
rechts: Zwischenergebnisse



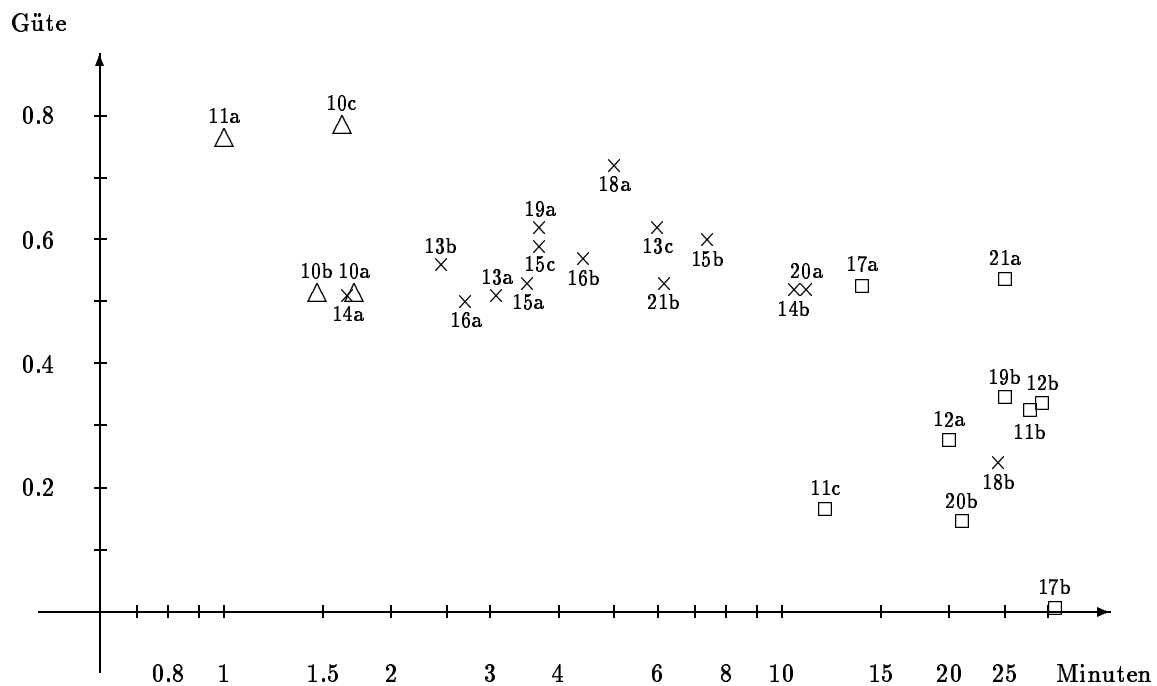
**Abb. A-20:** Verarbeitungsergebnisse mit vierfach überlagerten Bildern  
 a) Brücke in Überlagerung Brücke-Kreuzung-Wendehammer-Häuser  
 (Güte=0.52, Zeit=11:24)  
 b) Kreuzung in Überlagerung Brücke-Kreuzung-Wendehammer-Häuser  
 (Güte=0.14, Zeit=22:34)  
 links: Originalbilder mit zugeordnetem Referenzobjekt  
 rechts: Zwischenergebnisse



**Abb. A-21:** Verarbeitungsergebnisse mit vierfach überlagerten Bildern  
 a) Wendehammer in Überlagerung Brücke-Kreuzung-Wendehammer-Häuser  
 (Güte=0.53, Zeit=25:49)  
 b) Häuserformation in Überlagerung Brücke-Kreuzung-Wendehammer-Häuser  
 (Güte=0.50, Zeit=6:13)  
 links: Originalbilder mit zugeordnetem Referenzobjekt  
 rechts: Zwischenergebnisse

### A.3.2 Diskussion der Verarbeitungsergebnisse

Abbildung A-22 fasst alle Verarbeitungsergebnisse zusammen. Auf der Ordinate ist die Bewertung der Referenzobjekte aufgetragen, auf der Abszisse logarithmisch geteilt die zum Aufbau der Referenzobjekte vom Simulationssystem benötigte Rechenzeit<sup>1</sup>, die eine erste Näherung für die Anzahl der überprüften Hypothesen ist. In ungestörten Bildern gefundene Referenzobjekte wurden immer in weniger als zwei Minuten Rechenzeit aufgebaut, und es wurde die vorgegebene minimale Bewertung von 0.5 erreicht bzw. überschritten.



i△: gefunden im ungestörten Bild Abbildung A-i

i×: gefunden im gestörten Bild Abbildung A-i

i□: zugeordnet in Abbildung A-i

**Abb. A-22:** Vergleich der Verarbeitungsergebnisse

Störungen führen zu verlängerten Rechenzeiten. Bei den *leicht gestörten* Brückenbildern (Abbildung A-13b, A-14a, A-15a und A-15c) erhöhte sich beispielsweise die Rechenzeit gegenüber der ungestörten Verarbeitung maximal um den Faktor vier. Bleibt bei den untersuchten Bildern die Rechenzeit unterhalb von sechs Minuten, so kann davon ausgegangen werden, dass in einem Bild das wiederzufindende Referenzobjekt korrekt zugeordnet wurde. Wird länger als acht Minuten gerechnet, so besteht die Gefahr, dass eine Zuordnung getroffen wurde, zu der kein Referenzobjekt gehört. Ein Kriterium zur Unterscheidung von zuverlässigen

<sup>1</sup>Die Rechenzeit wird bei Einsatz eines 1 MIPS Rechners vom Typ VAX 11/780 benötigt.

und unsicheren Zuordnungen ist also die Anzahl der für eine Auswertung überprüften Hypothesen bzw. die benötigte Rechenzeit.

Ein weiteres Unterscheidungskriterium ist die Bewertung der zusammengesetzten Teilobjekte und des zugeordneten Referenzobjektes. Bei allen Verarbeitungsläufen, die nicht zur Mindestbewertung des Referenzobjektes von 0.5 führten, wurde das Programm beendet, nachdem alle Teilobjekte - einschließlich der *schlecht* und *sehr schlecht* bewerteten - bearbeitet wurden. Bewertungen kleiner 0.5 kennzeichnen also unsichere Zuordnungen zu Referenzobjekten.

Schlecht bewertete Zuordnungen können aus zwei Gründen entstehen:

1. In stark gestörten Bildern wurden keine passenden Primitivobjekte generiert.
2. Das zu suchende Objekt ist nicht im Bild enthalten, und für die im Bild enthaltenen Objekte sind keine Modelle bzw. Überprüfungsprogramme vorgesehen.

Ein Beispiel für den ersten Fall gibt Abbildung A-20b, in der die zur Kreuzung gehörenden schmalen Streifen nicht gefunden wurden, weil die Vorverarbeitung nicht genügend viele passende Geradenstücke generierte. Ein weiteres Beispiel gibt Abbildung A-21a, in der auch ein menschlicher Interpret Schwierigkeiten hat, den Wendehammer in der Mitte der oberen Bildhälfte zu erkennen. Gerade noch zugeordnet wurden die Referenzobjekte in den Abbildungen A-14b, A-18b und A-20a. Bei diesen sehr stark gestörten Bildern müßte die Vorverarbeitung optimiert werden, indem auf die jeweilige Störung und die wiederzufindenden Referenzobjekte adaptierte Vorverarbeitungsprogramme eingesetzt werden, die aus den Bilddaten zusätzliche Primitivobjekte generieren.

Ein Beispiel für den zweiten Fall, die Verarbeitung mit fehlenden Modellen, ist die Zuordnung der Häuserformation zu Teilen der Autobahnbrücke (Abbildung A-17a) (Güte=0.52, Zeit=15:04). Diese Zuordnung war nur möglich, weil erstens das wiederzufindende Referenzobjekt nicht im Bild enthalten war, weil zweitens sowohl Häuser als auch Autobahnen aus Streifen ähnlicher Breite aufgebaut werden und weil drittens bei der Suche nach der Häuserformation nicht gleichzeitig auch nach der Autobahnbrücke gesucht wurde, mit der die Häuserformation verwechselt werden kann. Wird im Bild wegen der Verwechslungsgefahr sowohl nach der Autobahnbrücke als auch nach der Häuserformation gesucht, dann wird die Autobahnbrücke mit nur einem Sechstel des Aufwandes gefunden (Abbildung A-16a) (Güte=0.50, Zeit=2:43), der für die Zuordnung der Häuserformation benötigt wird, so dass eine kombinierte Auswertung die Brücke und nicht die Häuserformation zuordnen würde. In Abbildung A-19b wird die nicht im Bild enthaltene Häuserformation im dreifachüberlagerten Bild gesucht. Das Ergebnis entspricht dem gerade für Abbildung A-17a diskutierten Ergebnis.

Weitere Beispiele für den zweiten Fall geben die Abbildungen A-17b, A-11b, A-11c und A-12. In Abbildung A-17b drückt die sehr niedrige Bewertung ( $g=0.01$ ) der relativ besten Zuordnung aus, dass die gesuchte Kreuzung mit großer Sicherheit nicht im Bild enthalten ist.

Mit den Abbildungen A-11b, A-11c und A-12 soll demonstriert werden, dass das System versucht, mit den eingesetzten Modellen das wiederzufindende Referenzobjekt zuordnen zu können.

renzobjekt möglichst gut in das Grauwertgebirge der abzusuchenden Bilder einzupassen. In Abbildung A-11b wurde beispielsweise rechts und links vom Schatten, der oberhalb der Statue senkrecht nach oben verläuft, eine *Autobahn* aus zwei *Streifen* aufgebaut, deren Abstand genau den Wert hat, der für Autobahnen erwartet wird. Diese *Autobahn* wird von der Verlängerung einer *schmalen Straße* geschnitten, die oberhalb des Eingangsportales des Schlosses zugeordnet wurde. Der Schnittpunkt wird dem Zentrum der wiederzufindenden *Autobahnbrücke* zugeordnet. In Abbildung A-12a wird eine Säule als Zufahrtsstraße des wiederzufindenden Wendehammers interpretiert, an die ein kleines Kreissegment angelagert werden konnte, so dass ein schlecht bewerteter Wendehammer zugeordnet wurde (Güte=0.27, Zeit=20:07). In Abbildung A-12b werden die den Fenstern zugeordneten Streifen aufgrund des eingesetzten Häusermodells als Haus interpretiert und zu einer schlecht bewerteten Häuserformation (Güte=0.33, Zeit=29:07) zusammengesetzt.

Diese Fähigkeit, selbst in Daten, die nicht der Modellierung entsprechen, nachvollziehbare Zuordnungen zu treffen, erlaubt, auch stark gestörte Bilder, wie beispielsweise die Abbildungen A-14b, A-15b und A-20a, auszuwerten. Gestörte Bilder, aus denen einzelne falsche Primitivobjekte extrahiert werden, und unvollständige sowie fehlerhafte Modelle führen also nicht unmittelbar zu falschen Ergebnissen, sondern veranlassen die Berücksichtigung zusätzlicher schlechter bewerteter Teilobjekte, so dass mehr Hypothesen zu überprüfen sind und ein erhöhter Aufwand entsteht. Dieses gutartige Fehlerverhalten ermöglicht in vielen Fällen eine aufgabengerechte reproduzierbare Analyse gestörter Bilder mit vagen Modellen.

## Anhang B Formale Darstellung der Systeme und Automaten

Die Darstellungen in den vorhergehenden Kapiteln sollen Probleme, Lösungswege und Zusammenhänge erläutern; formale Formulierungen erfolgen dabei in vielen Fällen verkürzt und unvollständig. Anschließend werden die mit dieser Arbeit neu eingeführten Zuordnungssysteme und Zuordnungsautomaten sowie darauf aufbauende Sätze lückenlos definiert und bewiesen. Beispiele hierzu sind in den vorhergehenden Kapiteln zu finden.

### B.1 Zuordnungssysteme

#### B.1.1 Zuordnungssysteme mit Instanzen

**Definition B.1** Ein *Zuordnungssystem* (kurz: Z-System) ist ein Paar  $Z(\mathcal{I}, \mathbf{A})$  mit

$$\begin{array}{l} \mathcal{I} : \text{Menge von Instanzen} \\ \mathbf{A} : \text{Relation der Instanz-Assoziationen} \end{array} \quad (\text{B} - 1)$$

Die Elemente  $\gamma$  der Menge  $\mathcal{I}$  sind die *Instanzen* des Z-Systems. Instanzen werden mit *kleinen griechischen* Buchstaben bezeichnet.

Mengen von Instanzen  $\mathcal{X} \subseteq \mathcal{I}$  werden *Instanz-Konfigurationen* (kurz: Konfiguration) genannt und mit *großen kalligraphischen* Buchstaben bezeichnet.

Die Elemente der *Relation der Instanz-Assoziationen*  $\mathbf{A} \subseteq \wp(\mathcal{I}) \times \wp(\mathcal{I})$  (kurz: Menge der Assoziationen)  $(\mathcal{X}, \mathcal{Y}) \in \mathbf{A}$  werden geschrieben  $\mathcal{X} \overset{\mathbf{A}}{\sim} \mathcal{Y}$ , in Worten: die Konfiguration  $\mathcal{X}$  *assoziiert* die Konfiguration  $\mathcal{Y}$ .

Die Menge der *assoziierten Instanzen*  $\mathcal{I}_a$  ist die Vereinigung der Konfigurationen aller Assoziationen:

$$\mathcal{I}_a(\mathbf{A}) := \bigcup_{(\mathcal{X}, \mathcal{Y}) \in \mathbf{A}} (\mathcal{X} \cup \mathcal{Y}) \quad (\text{B} - 2)$$

Die Konfiguration  $\mathcal{S}$  wird der Konfiguration  $\mathcal{R}$  durch das Z-System  $Z(\mathcal{I}, \mathbf{A})$  *direkt zugeordnet*, in Zeichen  $\mathcal{R} \overset{Z}{\mid} \mathcal{S}$ , wenn es eine Assoziation  $(\mathcal{X}, \mathcal{Y}) \in \mathbf{A}$  gibt, so dass  $\mathcal{X}$  vorliegt,  $\mathcal{Y}$  noch nicht vorliegt und  $\mathcal{S}$  die Vereinigung von  $\mathcal{R}$  und  $\mathcal{Y}$  ist:

$$\overset{Z}{\mid} := \left\{ (\mathcal{R}, \mathcal{S}) \in \wp(\mathcal{I}) \times \wp(\mathcal{I}) \mid \begin{array}{l} \exists (\mathcal{X}, \mathcal{Y}) \in \mathbf{A} : \\ \mathcal{X} \subseteq \mathcal{R} \\ \mathcal{Y} \not\subseteq \mathcal{R} \\ \mathcal{S} = \mathcal{R} \cup \mathcal{Y} \end{array} \right\} \quad (\text{B} - 3)$$

Die Konfiguration  $\mathcal{S}$  wird der Konfiguration  $\mathcal{R}$  durch das Z-System  $Z(\mathcal{I}, \mathbf{A})$  *zugeordnet*, in Zeichen  $\mathcal{R} \overset{*Z}{\mid} \mathcal{S}$ , wenn mit  $\mathcal{R} = \mathcal{Q}_0$  und  $\mathcal{S} = \mathcal{Q}_k$  keine ( $k = 0$ ) direkte



Zuordnung, eine ( $k = 1$ ) direkte Zuordnung oder mehrere ( $k > 1$ ) direkte Zuordnungen  $Q_i \stackrel{Z}{|} Q_{i+1}, i = 0, 1, \dots, k - 1$  ausgeführt werden:

$$\begin{aligned} \stackrel{(0)Z}{|} &:= \{(\mathcal{R}, \mathcal{R}) \mid \mathcal{R} \subseteq \mathcal{I}\} \\ \stackrel{(i+1)Z}{|} &:= \left\{ (\mathcal{R}, \mathcal{S}) \in \wp(\mathcal{I}) \times \wp(\mathcal{I}) \mid \begin{array}{l} \exists Q_i \subseteq \mathcal{I} : \\ (\mathcal{R}, Q_i) \in \stackrel{(i)Z}{|} \\ (Q_i, \mathcal{S}) \in \stackrel{Z}{|} \end{array} \right\}, i \in \mathbb{N}_0 \\ \stackrel{+Z}{|} &:= \bigcup_{i \in \mathbb{N}} \stackrel{(i)Z}{|} \\ \stackrel{*Z}{|} &:= \stackrel{+Z}{|} \cup \stackrel{(0)Z}{|} \end{aligned} \quad (\text{B-4})$$

Die Konfiguration  $\mathcal{S}$  wird der Konfiguration  $\mathcal{R}$  durch das Z-System  $Z(\mathcal{I}, \mathbf{A})$  *abschließend zugeordnet*, in Zeichen  $\mathcal{R} \stackrel{\dagger Z}{|} \mathcal{S}$ , wenn die Konfiguration  $\mathcal{S}$  der Konfiguration  $\mathcal{R}$  zugeordnet wird ( $\mathcal{R} \stackrel{*Z}{|} \mathcal{S}$ ) und  $\mathcal{S}$  keine Konfiguration direkt zugeordnet werden kann:

$$\stackrel{\dagger Z}{|} := \left\{ (\mathcal{R}, \mathcal{S}) \in \stackrel{*Z}{|} \mid \forall (\mathcal{U}, \mathcal{V}) \in \stackrel{Z}{|} : \begin{array}{l} \mathcal{S} \neq \mathcal{U} \end{array} \right\} \quad (\text{B-5})$$

□

**Lemma B.2**  $\stackrel{*Z}{|}$  ist transitiv:

$$\left. \begin{array}{l} \forall \mathcal{R}, \mathcal{S}, \mathcal{T} \subseteq \mathcal{I} : \\ \mathcal{R} \stackrel{*Z}{|} \mathcal{S} \\ \mathcal{S} \stackrel{*Z}{|} \mathcal{T} \end{array} \right\} \rightarrow \mathcal{R} \stackrel{*Z}{|} \mathcal{T} \quad (\text{B-6})$$

**Beweis**  $\stackrel{*Z}{|}$  ist transitive reflexive Hülle von  $\stackrel{Z}{|}$  (Gl. B-4): Aus  $\mathcal{R} \stackrel{*Z}{|} \mathcal{S}$  folgt  $\exists k \in \mathbb{N}_0 : (\mathcal{R}, \mathcal{S}) \in \stackrel{(k)Z}{|}$ . Analog gilt  $\exists l \in \mathbb{N}_0 : (\mathcal{S}, \mathcal{T}) \in \stackrel{(l)Z}{|}$ , so dass  $(\mathcal{R}, \mathcal{T}) \in \stackrel{(k+l)Z}{|} \subseteq \stackrel{*Z}{|}$  gilt. □

**Lemma B.3** Die der Konfiguration  $\mathcal{R}$  mit dem Z-System  $Z(\mathcal{I}, \mathbf{A})$  zugeordnete Konfiguration  $\mathcal{S}$  wird mit maximal  $|\mathcal{S}| - |\mathcal{R}|$  direkten Zuordnungsschritten zugeordnet und enthält  $\mathcal{R}$ :

$$\mathcal{R} \stackrel{*Z}{|} \mathcal{S} \rightarrow \left\{ \begin{array}{l} \forall \mathcal{R}, \mathcal{S} \subseteq \mathcal{I} : \\ \exists k \in \mathbb{N}_0 : \\ \exists \mathcal{P}_0, \mathcal{P}_1, \dots, \mathcal{P}_k \subseteq \mathcal{I} : \\ \mathcal{R} = \mathcal{P}_0 \\ \mathcal{S} = \mathcal{P}_k \\ \mathcal{P}_0 \stackrel{Z}{|} \mathcal{P}_1 \stackrel{Z}{|} \dots \stackrel{Z}{|} \mathcal{P}_k, \text{ falls } k > 0 \\ k \leq |\mathcal{S}| - |\mathcal{R}| \\ \mathcal{R} \subseteq \mathcal{S} \\ \mathcal{R} \subset \mathcal{S}, \text{ falls } k > 0 \end{array} \right. \quad (\text{B-7})$$

**Beweis** Für Zuordnungen  $\mathcal{R} \stackrel{*Z}{\mid} \mathcal{R}$  gilt  $k = |\mathcal{R}| - |\mathcal{R}| = 0$  (Gl. B-4).

Für Zuordnungen  $\mathcal{R} \stackrel{*Z}{\mid} \mathcal{S}$  mit  $\mathcal{S} \neq \mathcal{R}$  gilt  $\exists k \in \mathbb{N} : (\mathcal{R}, \mathcal{S}) \in \stackrel{(k)Z}{\mid}$  und  $\exists \mathcal{P}_0, \mathcal{P}_1, \dots, \mathcal{P}_k \subseteq \mathcal{I} : \mathcal{R} = \mathcal{P}_0, \mathcal{S} = \mathcal{P}_k, \mathcal{P}_i \stackrel{Z}{\mid} \mathcal{P}_{i+1}, i = 0, 1, \dots, k-1$  (Gl. B-4). Für jeden direkten Zuordnungsschritt  $\mathcal{P}_i \stackrel{Z}{\mid} \mathcal{P}_{i+1}$  gibt es eine Assoziation  $(\mathcal{X}_i, \mathcal{Y}_i) \in \mathbf{A}$  mit  $\mathcal{X}_i \subseteq \mathcal{P}_i$  und  $\mathcal{Y}_i \not\subseteq \mathcal{P}_i$  sowie  $\mathcal{P}_{i+1} = \mathcal{P}_i \cup \mathcal{Y}_i$  (Gl. B-3). Damit gilt  $|\mathcal{P}_{i+1} \setminus \mathcal{P}_i| \geq 1$  sowie  $\mathcal{R} = \mathcal{P}_0 \subset \mathcal{P}_1 \subset \dots \subset \mathcal{P}_k = \mathcal{S}$ ; d.h.:  $\mathcal{R} \subset \mathcal{S}$ . Die Anzahl der Schritte ist  $k$ . Da je Schritt ein oder mehrere ( $|\mathcal{P}_{i+1} \setminus \mathcal{P}_i| \geq 1$ ) Instanzen hinzukommen, gilt  $|\mathcal{S}| \geq |\mathcal{R}| + k$ , d.h.  $k \leq |\mathcal{S}| - |\mathcal{R}|$ .  $\square$

**Verwendung** Lemma B.8, Satz B.11, Satz B.19

**Satz B.4** Wenn die Menge der assoziierten Instanzen  $\mathcal{I}_a(\mathbf{A})$  endlich ist, dann terminiert das Z-System  $(\mathcal{I}, \mathbf{A})$ :

$$\mathcal{I}_a(\mathbf{A}) \text{ ist endlich} \rightarrow \left\{ \begin{array}{l} \forall \mathbf{A} \subseteq \wp(\mathcal{I}) \times \wp(\mathcal{I}) : \\ \quad \forall \mathcal{R} \subseteq \mathcal{I} : \\ \quad \exists l \in \mathbb{N}_0 : \\ \quad \forall \mathcal{S} \subseteq \mathcal{I} : \\ (\mathcal{R}, \mathcal{S}) \in \stackrel{*Z}{\mid} \rightarrow \left\{ \begin{array}{l} \exists k \in \mathbb{N}_0 : \\ (\mathcal{R}, \mathcal{S}) \in \stackrel{(k)Z}{\mid} \\ k \leq l \end{array} \right. \end{array} \right. \quad (\text{B-8})$$

Hinweis: Terminieren wird auf Seite 16 und in /Ave95/ mit Definition 1.1.3 definiert.

**Beweis** In Gleichung B-8 wird  $\mathcal{R}$  in  $\mathcal{R}_1 := \mathcal{R} \cap \mathcal{I}_a(\mathbf{A})$  und  $\mathcal{R}_2 := \mathcal{R} \setminus \mathcal{I}_a(\mathbf{A})$  aufgeteilt.  $\mathcal{R}_2$  kann nach Definition von  $\mathcal{I}_a(\mathbf{A})$  mit Assoziationen von  $\mathbf{A}$  nicht verändert werden (Gl. B-2). Aus einer Zuordnung  $\mathcal{R} \stackrel{*Z}{\mid} \mathcal{S}$  wird also eine Zuordnung  $\mathcal{R}_1 \stackrel{*Z}{\mid} \mathcal{S}_1$  mit  $\mathcal{S}_1 \subseteq \mathcal{I}_a(\mathbf{A})$  und  $\mathcal{S} = \mathcal{S}_1 \cup \mathcal{R}_2$ .

Aus  $(\mathcal{R}_1, \mathcal{S}_1) \in \stackrel{*Z}{\mid}$  folgt  $\exists k \in \mathbb{N}_0 : (\mathcal{R}_1, \mathcal{S}_1) \in \stackrel{(k)Z}{\mid}$  (Gl. B-4). Mit Lemma B.3 gilt, dass die Anzahl der Zuordnungsschritte  $k$  bei der Zuordnung  $\mathcal{R}_1 \stackrel{*Z}{\mid} \mathcal{S}_1$  kleiner gleich  $|\mathcal{S}_1| - |\mathcal{R}_1|$  ist.

Da  $\mathcal{I}_a(\mathbf{A})$  endlich ist ( $\exists l \in \mathbb{N} : |\mathcal{I}_a(\mathbf{A})| \leq l$ ), gilt auch  $|\mathcal{S}_1| \leq l$ . Mit  $|\mathcal{R}_1| \geq 0$  erhält man  $k \leq |\mathcal{S}_1| - |\mathcal{R}_1| \leq |\mathcal{S}_1| \leq l$  bzw.  $k \leq l$ .  $\square$

**Verwendung** Korollar B.5, Lemma B.6, Satz B.12

**Korollar B.5** Wenn die Menge der Instanzen  $\mathcal{I}$  endlich ist, dann terminiert das Z-System  $(\mathcal{I}, \mathbf{A})$ .

**Beweis** Aus der Aussage, dass  $\mathcal{I}$  endlich ist folgt mit  $\mathcal{I}_a(\mathbf{A}) \subseteq \mathcal{I}$  die Aussage, dass  $\mathcal{I}_a(\mathbf{A})$  endlich ist (Gl. B-2). Aus Satz B.4 folgt Korollar B.5.  $\square$

**Lemma B.6** Genau dann, wenn das Z-System  $Z(\mathcal{I}, \mathbf{A})$  terminiert, wird jeder Konfiguration  $\mathcal{R}$  abschließend eine Konfiguration  $\mathcal{S}$  zugeordnet:

$$\forall \mathbf{A} \subseteq \wp(\mathcal{I}) \times \wp(\mathcal{I}) : \quad Z(\mathcal{I}, \mathbf{A}) \text{ terminiert} \leftrightarrow \left\{ \begin{array}{l} \forall \mathcal{R} \subseteq \mathcal{I} : \\ \exists \mathcal{S} \subseteq \mathcal{I} : \\ \mathcal{R} \stackrel{\dagger Z}{\dashv} \mathcal{S} \end{array} \right. \quad (\text{B} - 9)$$

**Beweis** 1. Teil:  $\rightarrow$ : Da es mit einem terminierenden Z-System keine unendliche Folge von Zuordnungsschritten gibt (Definition von *terminieren*), gibt es bei der Fortführung jeder Folge von Zuordnungsschritten einen letzten Zuordnungsschritt.

Für die zuletzt zugeordnete Konfiguration  $\mathcal{S}$  muss  $\forall (\mathcal{U}, \mathcal{V}) \in \stackrel{Z}{\dashv} : \mathcal{S} \neq \mathcal{U}$  gelten, andernfalls wäre es nicht der letzte mögliche Zuordnungsschritt; daraus folgt  $(\mathcal{R}, \mathcal{S}) \in \stackrel{\dagger Z}{\dashv}$  (Gl. B-5).

2. Teil:  $\leftarrow$ : Es wird angenommen, dass das Z-System nicht terminiert, so dass die linke Seite von Gleichung B-9 falsch ist. In diesem Fall gibt es eine Konfiguration  $\mathcal{R}$  mit einer unendlichen Folge von Zuordnungen. Es gibt also zu jeder  $\mathcal{R}$  zugeordneten Konfiguration  $\mathcal{S}$  eine Assoziation  $(\mathcal{X}, \mathcal{Y})$  mit  $\mathcal{X} \subseteq \mathcal{R}$  und  $\mathcal{Y} \not\subseteq \mathcal{S}$  (Gl. B-3). Daraus folgt, dass es für  $\mathcal{R}$  keine abschließende Zuordnung (Gl. B-5) gibt und die rechte Seite von Gleichung B-9 falsch ist.  $\square$

**Verwendung** Satz B.9, Satz B.61

**Lemma B.7** Genau dann, wenn der Konfiguration  $\mathcal{R}$  mit dem Z-System  $Z(\mathcal{I}, \mathbf{A})$  die Konfiguration  $\mathcal{S}$  abschließend zugeordnet wird ( $\mathcal{R} \stackrel{\dagger Z}{\dashv} \mathcal{S}$ ), gilt für alle Assoziationen  $(\mathcal{X}, \mathcal{Y}) \in \mathbf{A}$ , deren Konfiguration  $\mathcal{X}$  in  $\mathcal{S}$  enthalten ist, dass auch die assoziierte Konfiguration  $\mathcal{Y}$  in  $\mathcal{S}$  enthalten ist:

$$\forall \mathcal{R}, \mathcal{S} \subseteq \mathcal{I} : \quad \left. \begin{array}{l} \mathcal{R} \stackrel{*Z}{\dashv} \mathcal{S} \\ \forall (\mathcal{X}, \mathcal{Y}) \in \mathbf{A} : \\ \mathcal{X} \subseteq \mathcal{S} \rightarrow \mathcal{Y} \subseteq \mathcal{S} \end{array} \right\} \leftrightarrow \mathcal{R} \stackrel{\dagger Z}{\dashv} \mathcal{S} \quad (\text{B} - 10)$$

**Beweis** 1. Teil:  $\rightarrow$  Aus  $\forall (\mathcal{X}, \mathcal{Y}) \in \mathbf{A} : \mathcal{X} \subseteq \mathcal{S} \rightarrow \mathcal{Y} \subseteq \mathcal{S}$  folgt mit Gleichung B-3, dass  $\mathcal{S}$  keine Konfiguration direkt zugeordnet werden kann; zusammen mit  $\mathcal{R} \stackrel{*Z}{\dashv} \mathcal{S}$  folgt  $\mathcal{R} \stackrel{\dagger Z}{\dashv} \mathcal{S}$ .

2. Teil:  $\leftarrow$  Es wird angenommen, dass es eine Assoziation  $(\mathcal{X}, \mathcal{Y}) \in \mathbf{A}$  mit  $\mathcal{X} \subseteq \mathcal{S}$  und  $\mathcal{Y} \not\subseteq \mathcal{S}$  gibt. In diesem Fall ist die linke Seite von Gleichung B-10 falsch. Aus  $\exists (\mathcal{X}, \mathcal{Y}) \in \mathbf{A} : \mathcal{X} \subseteq \mathcal{S} \wedge \mathcal{Y} \not\subseteq \mathcal{S}$  folgt dann, dass der Konfiguration  $\mathcal{S}$  die Konfiguration  $\mathcal{T} = \mathcal{S} \cup \mathcal{Y}$  direkt zugeordnet werden kann (Gl. B-3) und  $\mathcal{R} \stackrel{\dagger Z}{\dashv} \mathcal{S}$  falsch ist.  $\square$

**Verwendung** Lemma B.8, Satz B.61

**Lemma B.8** Die der Konfiguration  $\mathcal{R}$  mit dem Z-System  $Z(\mathbf{A}, \mathcal{I})$  abschließend zugeordnete Konfiguration  $\mathcal{S}$  enthält alle Konfigurationen, die  $\mathcal{R}$  mit dem Z-System

$Z(\mathbf{A}, \mathcal{I})$  zugeordnet werden können:

$$\left. \begin{array}{l} \forall \mathcal{R}, \mathcal{S}, \mathcal{T} \subseteq \mathcal{I} : \\ \mathcal{R} \left| \begin{array}{l} \dagger^Z \\ \hline \end{array} \right. \mathcal{S} \\ \mathcal{R} \left| \begin{array}{l} *^Z \\ \hline \end{array} \right. \mathcal{T} \end{array} \right\} \rightarrow \mathcal{T} \subseteq \mathcal{S} \quad (\text{B} - 11)$$

**Beweis** 1. Teil: Es ist zu zeigen, dass für ein beliebiges  $i \in \mathbb{N}$  aus

$$\left. \begin{array}{l} \mathcal{R} \left| \begin{array}{l} \dagger^Z \\ \hline \end{array} \right. \mathcal{S} \\ \mathcal{R} \left| \begin{array}{l} (i)^Z \\ \hline \end{array} \right. \mathcal{U}_i \end{array} \right\} \rightarrow \mathcal{U}_i \subseteq \mathcal{S} \quad (\text{B} - 12)$$

und  $\mathcal{U}_i \left| \begin{array}{l} Z \\ \hline \end{array} \right. \mathcal{U}_{i+1}$  folgt:

$$\left. \begin{array}{l} \mathcal{R} \left| \begin{array}{l} \dagger^Z \\ \hline \end{array} \right. \mathcal{S} \\ \mathcal{R} \left| \begin{array}{l} (i+1)^Z \\ \hline \end{array} \right. \mathcal{U}_{i+1} \end{array} \right\} \rightarrow \mathcal{U}_{i+1} \subseteq \mathcal{S} \quad (\text{B} - 13)$$

Mit der direkten Zuordnung  $\mathcal{U}_i \left| \begin{array}{l} Z \\ \hline \end{array} \right. \mathcal{U}_{i+1}$  gilt  $\mathcal{R} \left| \begin{array}{l} (i)^Z \\ \hline \end{array} \right. \mathcal{U}_i \left| \begin{array}{l} Z \\ \hline \end{array} \right. \mathcal{U}_{i+1}$  bzw.  $\mathcal{R} \left| \begin{array}{l} (i+1)^Z \\ \hline \end{array} \right. \mathcal{U}_{i+1}$  (Gl. B-4). Dabei gilt  $\exists(\mathcal{X}, \mathcal{Y}) \in \mathbf{A} : \mathcal{X} \subseteq \mathcal{U}_i \wedge \mathcal{Y} \not\subseteq \mathcal{U}_i \wedge \mathcal{U}_{i+1} = \mathcal{U}_i \cup \mathcal{Y}$  (Gl. B-3). Aus  $\mathcal{U}_i \subseteq \mathcal{S}$  folgt also  $\mathcal{X} \subseteq \mathcal{S}$ . Mit Lemma B.7 gilt  $\forall(\mathcal{X}, \mathcal{Y}) \in \mathbf{A} : \mathcal{X} \subseteq \mathcal{S} \rightarrow \mathcal{Y} \subseteq \mathcal{S}$ . Aus  $\mathcal{X} \subseteq \mathcal{S}$  folgen also  $\mathcal{Y} \subseteq \mathcal{S}$  und  $\mathcal{U}_i \cup \mathcal{Y} = \mathcal{U}_{i+1} \subseteq \mathcal{S}$ . Aus Gleichung B-12 folgt also Gleichung B-13.

2. Teil: Mit  $\mathcal{R} \subseteq \mathcal{S}$  (Lemma B.3) und  $\mathcal{U}_0 = \mathcal{R}$  (Gl. B-4) gilt Gleichung B-12 für  $i = 0$ . Mit dem 1. Teil des Beweises gilt Gleichung B-12 für alle  $i \in \mathbb{N}_0$ . Mit Gleichung B-4 erhält man Gleichung B-11.  $\square$

**Verwendung** Satz B.9, Satz B.14

**Satz B.9** Jeder Konfiguration  $\mathcal{R}$  wird mit dem terminierenden Z-System  $Z(\mathcal{I}, \mathbf{A})$  genau eine Konfiguration  $\mathcal{S}$  abschließend zugeordnet:

$$Z(\mathcal{I}, \mathbf{A}) \text{ terminiert} \rightarrow \left\{ \begin{array}{l} \forall \mathcal{R} \subseteq \mathcal{I} : \\ \exists! \mathcal{S} \subseteq \mathcal{I} : \\ \mathcal{R} \left| \begin{array}{l} \dagger^Z \\ \hline \end{array} \right. \mathcal{S} \end{array} \right. \quad (\text{B} - 14)$$

**Beweis** Jeder Konfiguration  $\mathcal{R}$  wird mit dem terminierenden Z-System  $(\mathcal{I}, \mathbf{A})$  eine Konfiguration  $\mathcal{S}$  abschließend zugeordnet (Lemma B.6). Es wird jetzt angenommen, dass  $\mathcal{R}$  zwei unterschiedliche Konfigurationen  $\mathcal{S}$  und  $\mathcal{T}$  abschließend zugeordnet werden können ( $\mathcal{R} \left| \begin{array}{l} \dagger^Z \\ \hline \end{array} \right. \mathcal{S}, \mathcal{R} \left| \begin{array}{l} \dagger^Z \\ \hline \end{array} \right. \mathcal{T}, \mathcal{S} \neq \mathcal{T}$ ), so dass  $\exists! \mathcal{S} \subseteq \mathcal{I} : \mathcal{R} \left| \begin{array}{l} \dagger^Z \\ \hline \end{array} \right. \mathcal{S}$  falsch ist.

Aus  $\mathcal{R} \left| \begin{array}{l} \dagger^Z \\ \hline \end{array} \right. \mathcal{S}$  folgt unmittelbar  $\mathcal{R} \left| \begin{array}{l} *^Z \\ \hline \end{array} \right. \mathcal{S}$ , und aus  $\mathcal{R} \left| \begin{array}{l} \dagger^Z \\ \hline \end{array} \right. \mathcal{T}$  folgt analog  $\mathcal{R} \left| \begin{array}{l} *^Z \\ \hline \end{array} \right. \mathcal{T}$ . Mit  $\mathcal{R} \left| \begin{array}{l} *^Z \\ \hline \end{array} \right. \mathcal{S}$  und  $\mathcal{R} \left| \begin{array}{l} \dagger^Z \\ \hline \end{array} \right. \mathcal{T}$  gilt dann  $\mathcal{S} \subseteq \mathcal{T}$  (Lemma B.8), und aus  $\mathcal{R} \left| \begin{array}{l} *^Z \\ \hline \end{array} \right. \mathcal{T}$  und  $\mathcal{R} \left| \begin{array}{l} \dagger^Z \\ \hline \end{array} \right. \mathcal{S}$  folgt analog  $\mathcal{T} \subseteq \mathcal{S}$ . Aus  $\mathcal{S} \subseteq \mathcal{T}$  und  $\mathcal{T} \subseteq \mathcal{S}$  folgt schließlich  $\mathcal{T} = \mathcal{S}$ , so dass  $\mathcal{S} \neq \mathcal{T}$  zu einem Widerspruch führt.  $\square$

**Verwendung** Korollar B.10, Satz B.12, Satz B.61, Korollar B.62

**Korollar B.10** Die Relation *abschließend zuordnen*  $\overset{\dagger Z}{\longmapsto}$  definiert eine Analysefunktion:

$$\begin{aligned} AF : \wp(\mathcal{I}) &\rightarrow \wp(\mathcal{I}) \\ (\mathcal{R}) &\mapsto AF(\mathcal{R}) \\ \mathcal{R} &\overset{\dagger Z}{\longmapsto} AF(\mathcal{R}) \end{aligned} \quad (\text{B} - 15)$$

**Beweis**  $\overset{\dagger Z}{\longmapsto}$  ist rechtseindeutig (Satz B.9). □

**Verwendung** Satz B.12, Lemma B.16, Korollar B.63

**Satz B.11** Z-Systeme  $(\mathcal{I}, \mathbf{A})$  sind streng konfluent:

$$\begin{aligned} \forall \mathbf{A} \subseteq \wp(\mathcal{I}) \times \wp(\mathcal{I}) : \\ \forall \mathcal{R}, \mathcal{U}, \mathcal{V} \subseteq \mathcal{I} : \\ \left. \begin{array}{l} \mathcal{R} \overset{Z}{\longmapsto} \mathcal{U} \\ \mathcal{R} \overset{Z}{\longmapsto} \mathcal{V} \end{array} \right\} \rightarrow \left\{ \begin{array}{l} \exists \mathcal{S} \subseteq \mathcal{I} : \\ \mathcal{U} \overset{*Z}{\longmapsto} \mathcal{S} \\ \mathcal{V} \overset{\epsilon Z}{\longmapsto} \mathcal{S} \end{array} \right. \end{aligned} \quad (\text{B} - 16)$$

**Hinweis:** Strenge Konfluenz wird auf Seite 16 und in /Ave95/ mit Definition 1.2.10 definiert.

**Beweis** Aus  $\mathcal{R} \overset{Z}{\longmapsto} \mathcal{U}$  folgt  $\exists (\mathcal{X}_u, \mathcal{Y}_u) \in \mathbf{A} : \mathcal{X}_u \subseteq \mathcal{R}, \mathcal{U} = \mathcal{R} \cup \mathcal{Y}_u$  (Gl. B-3, B-4).  
 Aus  $\mathcal{R} \overset{Z}{\longmapsto} \mathcal{V}$  folgt analog  $\exists (\mathcal{X}_v, \mathcal{Y}_v) \in \mathbf{A} : \mathcal{X}_v \subseteq \mathcal{R}, \mathcal{V} = \mathcal{R} \cup \mathcal{Y}_v$ .

Mit  $\mathcal{X}_v \subseteq \mathcal{R} \subseteq \mathcal{U}$  gilt  $\mathcal{X}_v \subseteq \mathcal{U}$ , so dass für  $\mathcal{Y}_v \not\subseteq \mathcal{U}$  die Aussage  $\mathcal{U} \overset{Z}{\longmapsto} (\mathcal{U} \cup \mathcal{Y}_v)$  gilt. Für  $\mathcal{Y}_v \subseteq \mathcal{U}$  gilt  $\mathcal{U} = \mathcal{U} \cup \mathcal{Y}_v$ , d. h.  $\mathcal{U} \overset{(0)}{\longmapsto} (\mathcal{U} \cup \mathcal{Y}_v)$ . In beiden Fällen gilt daher  $\mathcal{U} \overset{*Z}{\longmapsto} (\mathcal{U} \cup \mathcal{Y}_v)$  (Gl. B-4).

Analog erhält man mit  $\mathcal{X}_u \subseteq \mathcal{R} \subseteq \mathcal{V}$  für  $\mathcal{Y}_u \not\subseteq \mathcal{V}$  die Aussage  $\mathcal{V} \overset{Z}{\longmapsto} (\mathcal{V} \cup \mathcal{Y}_u)$  und für  $\mathcal{Y}_u \subseteq \mathcal{V}$  die Aussage  $\mathcal{V} = \mathcal{V} \cup \mathcal{Y}_u$  bzw.  $\mathcal{V} \overset{(0)}{\longmapsto} (\mathcal{V} \cup \mathcal{Y}_u)$ . Nach Definition von  $\overset{\epsilon Z}{\longmapsto}$  gilt  $\mathcal{V} \overset{\epsilon Z}{\longmapsto} (\mathcal{V} \cup \mathcal{Y}_u)$ .

Mit  $\mathcal{S} = \mathcal{U} \cup \mathcal{V} = \mathcal{R} \cup \mathcal{Y}_u \cup \mathcal{Y}_v$  erhält man Gleichung B-16. □

**Verwendung** Satz B.12

**Satz B.12** Terminierende Z-Systeme sind konvergent und erzeugen für jede Konfiguration  $\mathcal{R}$  genau eine Normalform: Sei  $\mathcal{R}$  eine Konfiguration und sei  $\mathcal{T}$  eine Konfiguration, die  $\mathcal{R}$  zugeordnet werden kann, dann ist  $AF(\mathcal{R}) = AF(\mathcal{T})$ :

$$\left. \begin{array}{l} \forall \mathcal{R}, \mathcal{T} \subseteq \mathcal{I} : \\ Z(\mathcal{I}, \mathbf{A}) \text{ terminiert} \\ \mathcal{R} \overset{*Z}{\longmapsto} \mathcal{T} \end{array} \right\} \rightarrow AF(\mathcal{R}) = AF(\mathcal{T}) \quad (\text{B} - 17)$$

**Hinweis:** Konvergenz wird auf Seite 16 und in /Ave95/ mit Definition 1.2.7 definiert.

**Beweis 1.** Variante ohne Satz B.11:

Der Konfiguration  $\mathcal{T}$  wird abschließend  $AF(\mathcal{T})$  zugeordnet (Korollar B.10). Aus  $\mathcal{R} \xrightarrow{*Z} \mathcal{T} \xrightarrow{\dagger Z} AF(\mathcal{T})$  folgt  $\mathcal{R} \xrightarrow{\dagger Z} AF(\mathcal{T})$  (Lemma B.2). Da jeder Konfiguration mit terminierenden  $Z$ -Systemen genau eine Konfiguration abschließend zugeordnet wird (Satz B.9) und  $\mathcal{R}$  abschließend  $AF(\mathcal{R})$  zugeordnet wird (Korollar B.10), folgt  $AF(\mathcal{R}) = AF(\mathcal{T})$ .

2. Variante mit Satz B.11:

$\xrightarrow{*Z}$  ist streng konfluent (Satz B.11) und daher konfluent (/Ave95/ Lemma 1.2.11).

$\xrightarrow{*Z}$  ist konfluent und es gibt keine unendliche Folge direkter Zuordnungsschritte (Definition von *terminieren*). Deshalb ist  $\xrightarrow{*Z}$  konvergent (/Ave95/ Lemma 1.2.7) und erzeugt für jede Konfiguration  $\mathcal{R}$  genau eine Normalform (/Ave95/ Lemma 1.2.6 und 1.2.8).  $\square$

**Verwendung** Satz B.14, Korollar B.64, Satz B.68, Satz B.71

**Lemma B.13** Sei  $\mathbf{A}_t$  Teilmenge von  $\mathbf{A}$  mit  $Z_t = (\mathcal{I}, \mathbf{A}_t)$ , dann folgt aus der Zuordnung der Konfiguration  $\mathcal{S}$  zur Konfiguration  $\mathcal{R}$  mit der Menge der Assoziationen  $\mathbf{A}_t$ , dass  $\mathcal{R}$  mit  $\mathbf{A}$  die Konfiguration  $\mathcal{S}$  zugeordnet wird:

$$\left. \begin{array}{l} \forall \mathbf{A} \subseteq \wp(\mathcal{I}) \times \wp(\mathcal{I}) : \\ \forall \mathbf{A}_t \subseteq \mathbf{A} : \\ Z_t := (\mathcal{I}, \mathbf{A}_t) \\ \mathcal{R} \xrightarrow{*Z_t} \mathcal{S} \end{array} \right\} \rightarrow \mathcal{R} \xrightarrow{*Z} \mathcal{S} \quad (\text{B} - 18)$$

**Beweis** Aus  $\mathcal{R} \xrightarrow{*Z_t} \mathcal{S}$  folgt für  $\mathcal{R} \neq \mathcal{S}$  die Aussage  $\exists k \in \mathbb{N} : \exists \mathcal{P}_0, \mathcal{P}_1, \dots, \mathcal{P}_k \subseteq \mathcal{I} : \mathcal{P}_0 = \mathcal{R}, \mathcal{P}_k = \mathcal{S}, \mathcal{P}_i \xrightarrow{Z_t} \mathcal{P}_{i+1}, \exists (\mathcal{X}_i, \mathcal{Y}_i) \in \mathbf{A}_t \subseteq \mathbf{A} : \mathcal{X}_i \subseteq \mathcal{P}_i, \mathcal{P}_{i+1} = \mathcal{P}_i \cup \mathcal{Y}_i, (\mathcal{X}_i, \mathcal{Y}_i) \in \mathbf{A}; i = 0, 1, \dots, k-1$  (Def. B.1), d. h.  $\mathcal{R} \xrightarrow{*Z} \mathcal{S}$ . Für  $\mathcal{R} = \mathcal{S}$  folgt aus  $\mathcal{R} \xrightarrow{*Z_t} \mathcal{R}$ , dass keine direkte Zuordnung erfolgt und auch  $\mathcal{R} \xrightarrow{*Z} \mathcal{R}$  gilt.  $\square$

**Verwendung** Satz B.14, Lemma B.21, Lemma B.48

**Satz B.14** Sei  $\mathbf{A}_t$  Teilmenge von  $\mathbf{A}$  mit  $Z_t = (\mathcal{I}, \mathbf{A}_t)$  und  $\mathcal{R} \xrightarrow{\dagger Z_t} \mathcal{T}$  sowie  $\mathcal{R} \xrightarrow{\dagger Z} \mathcal{S}$ , dann ist die Anzahl der für die abschließende Zuordnung  $\mathcal{R} \xrightarrow{\dagger Z} \mathcal{S}$  zugeordneten Instanzen gleich der Summe aus der Anzahl der Instanzen, die  $\mathcal{R}$  abschließend mit  $Z_t$  zugeordnet werden ( $\mathcal{R} \xrightarrow{\dagger Z_t} \mathcal{T}$ ) und der Anzahl der Instanzen, die  $\mathcal{T}$  abschließend mit  $Z$  zugeordnet werden ( $\mathcal{T} \xrightarrow{\dagger Z} \mathcal{S}$ ).

**Beweis** Aus  $\mathcal{R} \xrightarrow{\dagger Z_t} \mathcal{T}$  folgt  $\mathcal{R} \xrightarrow{*Z_t} \mathcal{T}$  (Gl. B-5). Mit  $\mathbf{A}_t \subseteq \mathbf{A}$  folgt daraus  $\mathcal{R} \xrightarrow{*Z} \mathcal{T}$  (Lemma B.13). Aus  $\mathcal{R} \xrightarrow{*Z} \mathcal{T}$  und  $\mathcal{R} \xrightarrow{\dagger Z} \mathcal{S}$  folgt  $\mathcal{T} \subseteq \mathcal{S}$  (Lemma B.8). Mit Satz B.12 folgt  $\mathcal{T} \xrightarrow{\dagger Z} \mathcal{S}$ . Mit der 1. Zuordnung  $\mathcal{R} \xrightarrow{\dagger Z_t} \mathcal{T}$  werden (da  $\mathcal{R} \subseteq$

$\mathcal{T}$  Lemma B.3)  $|\mathcal{T}| - |\mathcal{R}|$  Instanzen zugeordnet, mit der 2. Zuordnung  $\mathcal{T} \overset{\dagger Z}{\dashv} \mathcal{S}$  analog  $|\mathcal{S}| - |\mathcal{T}|$ . Insgesamt werden also  $|\mathcal{T}| - |\mathcal{R}| + |\mathcal{S}| - |\mathcal{T}| = |\mathcal{S}| - |\mathcal{R}|$  zugeordnet. Mit den zwei Zuordnungen  $\mathcal{R} \overset{*Z_t}{\dashv} \mathcal{T} \overset{*Z}{\dashv} \mathcal{S}$  werden also  $|\mathcal{S}| - |\mathcal{R}|$  Instanzen zugeordnet. Diese Anzahl wird auch bei der Zuordnung  $\mathcal{R} \overset{*Z}{\dashv} \mathcal{S}$  (bzw.  $\mathcal{R} \overset{\dagger Z}{\dashv} \mathcal{S}$ ) benötigt.  $\square$

### B.1.2 Parallele Zuordnungen

**Definition B.15** Die Konfiguration  $\mathcal{S}$  wird der Konfiguration  $\mathcal{R}$  durch das Z-System  $Z(\mathcal{I}, \mathbf{A})$  *direkt parallel zugeordnet*, in Zeichen  $\mathcal{R} \overset{\parallel Z}{\dashv} \mathcal{S}$ , wenn  $\mathcal{R}$  eine oder mehrere Konfigurationen  $\mathcal{P}_j, j = 1, 2, \dots$  zugeordnet werden und  $\mathcal{S}$  die Vereinigung der zugeordneten Konfigurationen  $\mathcal{P}_j$  ist:

$$\begin{aligned} \overset{Z(1)}{\dashv} &:= \overset{*Z}{\dashv} \\ \overset{Z(j+1)}{\dashv} &:= \left\{ (\mathcal{R}, \mathcal{S}) \in \wp(\mathcal{I}) \times \wp(\mathcal{I}) \mid \begin{array}{l} \exists (\mathcal{R}, \mathcal{P}_j) \in \overset{Z(j)}{\dashv} : \\ \exists (\mathcal{R}, \mathcal{P}_{j+1}) \in \overset{*Z}{\dashv} : \\ \mathcal{S} = \mathcal{P}_j \cup \mathcal{P}_{j+1} \end{array} \right\}, j \in \mathbb{N} \quad (\text{B} - 19) \\ \overset{\parallel Z}{\dashv} &:= \bigcup_{j \in \mathbb{N}} \overset{Z(j)}{\dashv} \end{aligned}$$

Die Konfiguration  $\mathcal{S}$  wird der Konfiguration  $\mathcal{R}$  durch das Z-System  $Z(\mathcal{I}, \mathbf{A})$  *parallel zugeordnet*, in Zeichen  $\mathcal{R} \overset{* \parallel Z}{\dashv} \mathcal{S}$ , wenn mit  $\mathcal{R} = \mathcal{Q}_0$  und  $\mathcal{S} = \mathcal{Q}_k$  keine ( $k = 0$ ) direkte parallele Zuordnung, eine ( $k = 1$ ) direkte parallele Zuordnung oder mehrere ( $k > 1$ ) direkte parallele Zuordnungen  $\mathcal{Q}_i \overset{\parallel Z}{\dashv} \mathcal{Q}_{i+1}, i = 0, 1, \dots, k - 1$  nacheinander ausgeführt werden:

$$\begin{aligned} \overset{(0) \parallel Z}{\dashv} &:= \{(\mathcal{R}, \mathcal{R}) \mid \mathcal{R} \subseteq \mathcal{I}\} \\ \overset{(i+1) \parallel Z}{\dashv} &:= \left\{ (\mathcal{R}, \mathcal{S}) \in \wp(\mathcal{I}) \times \wp(\mathcal{I}) \mid \begin{array}{l} \exists \mathcal{Q}_i \subseteq \mathcal{I} : \\ (\mathcal{R}, \mathcal{Q}_i) \in \overset{(i) \parallel Z}{\dashv} : \\ (\mathcal{Q}_i, \mathcal{S}) \in \overset{\parallel Z}{\dashv} \end{array} \right\}, i \in \mathbb{N}_0 \quad (\text{B} - 20) \\ \overset{* \parallel Z}{\dashv} &:= \bigcup_{i \in \mathbb{N}_0} \overset{(i) \parallel Z}{\dashv} \end{aligned}$$

Die Konfiguration  $\mathcal{S}$  wird der Konfiguration  $\mathcal{R}$  durch das Z-System  $(\mathcal{I}, \mathbf{A})$  *mit dem Tupel  $W$  unabhängig verteilt zugeordnet*, in Zeichen  $\mathcal{R} \overset{v \parallel Z}{W} \mathcal{S}$ , wenn es ein Tupel mit  $k$  Assoziationsmengen  $W = (\mathbf{B}_1, \mathbf{B}_2, \dots, \mathbf{B}_k); \mathbf{B}_j \subseteq \mathbf{A}; j = 1, 2, \dots, k$  gibt, mit denen Zuordnungen  $(\mathcal{U}_j, \mathcal{V}_j) \in \overset{*(\mathcal{I}, \mathbf{B}_j)}{\dashv}$  mit  $\mathcal{R} = \bigcup_{j=1}^k \mathcal{U}_j$  und  $\mathcal{S} = \bigcup_{j=1}^k \mathcal{V}_j$  durch-

geführt werden:

$$\begin{aligned}
& \forall \mathbf{A} \subseteq \wp(\mathcal{I}) \times \wp(\mathcal{I}) : \\
& \forall k \in \mathbb{N} : \forall W = (\mathbf{B}_1, \mathbf{B}_2, \dots, \mathbf{B}_k) \in \prod_{i=1}^k \wp(\mathbf{A}) : \\
& \quad \left| \frac{vZ(0)}{W} := \{(\emptyset, \emptyset)\} \right. \\
& \quad \left. \left| \frac{vZ(j)}{W} := \left\{ (\mathcal{R}, \mathcal{S}) \left| \begin{array}{l} \exists (\mathcal{U}_{j-1}, \mathcal{V}_{j-1}) \in \left| \frac{vZ(j-1)}{W} \right. : \\ \exists (\mathcal{U}_j, \mathcal{V}_j) \in \left| \frac{*(\mathcal{I}, \mathbf{B}_j)}{W} \right. : \\ \mathcal{R} = \mathcal{U}_{j-1} \cup \mathcal{U}_j \\ \mathcal{S} = \mathcal{V}_{j-1} \cup \mathcal{V}_j \end{array} \right. \right\} ; j \in \{1, 2, \dots, k\} \right. \\
& \quad \left. \left| \frac{vZ}{W} := \left| \frac{vZ(k)}{W} \right. \right. \right.
\end{aligned} \tag{B-21}$$

Die Konfiguration  $\mathcal{S}$  wird der Konfiguration  $\mathcal{R}$  mit dem Tupel  $W$  durch das  $Z$ -System  $(\mathcal{I}, \mathbf{A})$  verteilt zugeordnet, in Zeichen  $\mathcal{R} \left| \frac{v||Z}{W} \right. \mathcal{S}$ , wenn es eine Konfiguration  $\mathcal{T}$  gibt mit  $\mathcal{R} \left| \frac{v||Z}{W} \right. \mathcal{T} \left| \frac{*Z}{W} \right. \mathcal{S}$ :

$$\begin{aligned}
& \forall k \in \mathbb{N} : \forall W \in \prod_{i=1}^k \wp(\mathbf{A}) : \\
& \quad \left| \frac{v||Z}{W} := \left| \frac{v||Z}{W} \right. \circ \left| \frac{*Z}{W} \right. \right.
\end{aligned} \tag{B-22}$$

Die Konfiguration  $\mathcal{S}$  wird der Konfiguration  $\mathcal{R} = \bigcup_{i=1}^l \mathcal{U}_i; l \in \mathbb{N}$  durch das  $Z$ -System  $(\mathcal{I}, \mathbf{A})$  kaskadiert zugeordnet, in Zeichen  $\mathcal{R} \left| \frac{kZ}{W} \right. \mathcal{S}$ , wenn  $l$  Zuordnungen  $(\mathcal{U}_i \cup \mathcal{W}_{i-1}) \left| \frac{*Z}{W} \right. \mathcal{W}_i; i = 1, 2, \dots, l$ , die jeweils eine Teilkonfiguration  $\mathcal{U}_i$  von  $\mathcal{R}$  zusätzlich in die Analyse mit einbeziehen, durchgeführt werden. Dabei ist die 1. Konfiguration  $\mathcal{W}_i$  mit Zwischenergebnissen leer ( $\mathcal{W}_0 = \emptyset$ ). Die letzte Konfiguration mit Zwischenergebnissen ist die  $\mathcal{R}$  kaskadiert zugeordnete Konfiguration  $\mathcal{S} = \mathcal{W}_l$ .

$$\left| \frac{kZ}{W} := \left\{ (\mathcal{R}, \mathcal{S}) \left| \begin{array}{l} \exists l \in \mathbb{N} : \\ \exists \{\mathcal{U}_1, \mathcal{U}_2, \dots, \mathcal{U}_l\} \subseteq \wp(\mathcal{I}) : \\ \mathcal{W}_0 := \emptyset \\ \forall i = 1, 2, \dots, l : (\mathcal{U}_i \cup \mathcal{W}_{i-1}) \left| \frac{*Z}{W} \right. \mathcal{W}_i \\ \mathcal{R} = \bigcup_{i=1}^l \mathcal{U}_i \\ \mathcal{S} = \mathcal{W}_l \end{array} \right. \right\} \right. \tag{B-23}$$

Die Konfiguration  $\mathcal{S}$  wird der Konfiguration  $\mathcal{R}$  durch das  $Z$ -System  $Z(\mathcal{I}, \mathbf{A})$  direkt asynchron zugeordnet, in Zeichen  $\mathcal{R} \left| \frac{a||Z}{W} \right. \mathcal{S}$ , wenn es mit  $\mathcal{R} = \mathcal{Q}_0$  und  $\mathcal{S} = \mathcal{Q}_k$  eine Menge von Assoziationen  $\{(\mathcal{X}_0, \mathcal{Y}_0), (\mathcal{X}_1, \mathcal{Y}_1), \dots, (\mathcal{X}_{k-1}, \mathcal{Y}_{k-1})\} \subseteq \mathbf{A}$  gibt und



es für jede Assoziation  $(\mathcal{X}_i, \mathcal{Y}_i), i = 0, 1, \dots, k-1$  ein Paar von Konfigurationen  $(Q_i, Q_{i+1})$  sowie eine Konfiguration  $Q_j, j \leq i$  gibt, so dass  $\mathcal{X}_i$  in  $Q_j$  vorliegt und  $Q_{i+1}$  die Vereinigung von  $Q_i$  mit  $\mathcal{Y}_i$  ist:

$$\left. \begin{array}{l} \text{---}^{\text{a||Z}} \text{---} := \left\{ (\mathcal{R}, \mathcal{S}) \in \wp(\mathcal{I}) \times \wp(\mathcal{I}) \right. \\ \left. \begin{array}{l} \exists k \in \mathbb{N} : \\ \exists \{(\mathcal{X}_0, \mathcal{Y}_0), (\mathcal{X}_1, \mathcal{Y}_1), \dots, (\mathcal{X}_{k-1}, \mathcal{Y}_{k-1})\} \subseteq \mathbf{A} : \\ \exists \{Q_0, Q_1, \dots, Q_{k-1}\} \subseteq \wp(\mathcal{I}) : \\ \forall i \in \{0, 1, \dots, k-1\} : \\ \exists j \in \{0, 1, \dots, i\} : \\ \mathcal{X}_i \subseteq Q_j \\ Q_{i+1} = Q_i \cup \mathcal{Y}_i \\ \mathcal{R} = Q_0 \\ \mathcal{S} = Q_k \end{array} \right\} \end{array} \right\} \quad (\text{B} - 24)$$

Die Konfiguration  $\mathcal{S}$  wird der Konfiguration  $\mathcal{R}$  durch das Z-System  $Z(\mathcal{I}, \mathbf{A})$  *asynchron zugeordnet*, in Zeichen  $\mathcal{R} \text{---}^{*\text{a||Z}} \mathcal{S}$ , wenn mit  $\mathcal{R} = Q_0$  und  $\mathcal{S} = Q_k$  keine ( $k = 0$ ) direkte asynchrone Zuordnung, eine ( $k = 1$ ) direkte asynchrone Zuordnung oder mehrere ( $k > 1$ ) direkte asynchrone Zuordnungen  $Q_i \text{---}^{\text{a||Z}} Q_{i+1}, i = 0, 1, \dots, k-1$  nacheinander ausgeführt werden:

$$\begin{array}{l} \text{---}^{(0)\text{a||Z}} \text{---} := \{(\mathcal{R}, \mathcal{R}) \mid \mathcal{R} \subseteq \mathcal{I}\} \\ \text{---}^{(i+1)\text{a||Z}} \text{---} := \left\{ (\mathcal{R}, \mathcal{S}) \in \wp(\mathcal{I}) \times \wp(\mathcal{I}) \mid \begin{array}{l} \exists Q_i \subseteq \mathcal{I} : \\ (\mathcal{R}, Q_i) \in \text{---}^{(i)\text{a||Z}} \text{---} \\ (Q_i, \mathcal{S}) \in \text{---}^{\text{a||Z}} \text{---} \end{array} \right\}, i \in \mathbb{N}_0 \\ \text{---}^{*\text{a||Z}} \text{---} := \bigcup_{i \in \mathbb{N}_0} \text{---}^{(i)\text{a||Z}} \text{---} \end{array} \quad (\text{B} - 25)$$

Die Konfiguration  $\mathcal{S}$  wird der Konfiguration  $\mathcal{R}$  durch das Z-System  $Z(\mathcal{I}, \mathbf{A})$  *direkt mehrfach zugeordnet*, in Zeichen  $\mathcal{R} \text{---}^{\text{mZ}} \mathcal{S}$ , wenn es eine Assoziation  $(\mathcal{X}, \mathcal{Y}) \in \mathbf{A}$  gibt, so dass  $\mathcal{X}$  vorliegt und  $\mathcal{S}$  die Vereinigung von  $\mathcal{R}$  und  $\mathcal{Y}$  ist:

$$\text{---}^{\text{mZ}} \text{---} := \left\{ (\mathcal{R}, \mathcal{S}) \in \wp(\mathcal{I}) \times \wp(\mathcal{I}) \mid \begin{array}{l} \exists (\mathcal{X}, \mathcal{Y}) \in \mathbf{A} : \\ \mathcal{X} \subseteq \mathcal{R} \\ \mathcal{S} = \mathcal{R} \cup \mathcal{Y} \end{array} \right\} \quad (\text{B} - 26)$$

Die Konfiguration  $\mathcal{S}$  wird der Konfiguration  $\mathcal{R}$  durch das Z-System  $Z(\mathcal{I}, \mathbf{A})$  *mehrfach zugeordnet*, in Zeichen  $\mathcal{R} \text{---}^{*\text{mZ}} \mathcal{S}$ , wenn mit  $\mathcal{R} = Q_0$  und  $\mathcal{S} = Q_k$  keine ( $k = 0$ ) direkte mehrfache Zuordnung, eine ( $k = 1$ ) direkte mehrfache Zuordnung oder mehrere ( $k > 1$ ) direkte mehrfache Zuordnungen  $Q_i \text{---}^{\text{mZ}} Q_{i+1}, i = 0, 1, \dots, k-1$

nacheinander ausgeführt werden:

$$\begin{aligned} \underline{\quad}^{(0)mZ} &:= \{(\mathcal{R}, \mathcal{R}) \mid \mathcal{R} \subseteq \mathcal{I}\} \\ \underline{\quad}^{(i+1)mZ} &:= \left\{ (\mathcal{R}, \mathcal{S}) \in \wp(\mathcal{I}) \times \wp(\mathcal{I}) \mid \begin{array}{l} \exists Q_i \subseteq \mathcal{I} : \\ (\mathcal{R}, Q_i) \in \underline{\quad}^{(i)mZ} \\ (Q_i, \mathcal{S}) \in \underline{\quad}^{mZ} \end{array} \right\}, i \in \mathbb{N}_0 \quad (\text{B-27}) \\ \underline{\quad}^{*mZ} &:= \bigcup_{i \in \mathbb{N}_0} \underline{\quad}^{(i)mZ} \end{aligned}$$

□

**Lemma B.16** Die Relationen *zuordnen*  $\underline{\quad}^{*Z}$  und *mehrfach zuordnen*  $\underline{\quad}^{*mZ}$  sind identisch.

**Beweis** 1. Teil: Der Vergleich von Gleichung B-3 mit Gleichung B-26 zeigt, dass alle Elemente der Relation  $\underline{\quad}^Z$  auch Elemente der Relation  $\underline{\quad}^{mZ}$  sind ( $\underline{\quad}^Z \subseteq \underline{\quad}^{mZ}$ ). Daraus folgt  $\underline{\quad}^{*Z} \subseteq \underline{\quad}^{*mZ}$  (Gl. B-4, Gl. B-27).

2. Teil: Für die in  $\underline{\quad}^{mZ}$  enthaltenen Elemente  $(\mathcal{R}, \mathcal{S})$ , die nicht in  $\underline{\quad}^Z$  enthalten sind, gilt (Gl. B-3, Gl. B-26)

$$\underline{\quad}^{mZ} \setminus \underline{\quad}^Z = \left\{ (\mathcal{R}, \mathcal{S}) \in \wp(\mathcal{I}) \times \wp(\mathcal{I}) \mid \begin{array}{l} \exists (\mathcal{X}, \mathcal{Y}) \in \mathbf{A} : \\ \mathcal{X} \subseteq \mathcal{R} \\ \mathcal{Y} \subseteq \mathcal{R} \\ \mathcal{S} = \mathcal{R} \cup \mathcal{Y} \end{array} \right\} \quad (\text{B-28})$$

bzw.

$$\underline{\quad}^{mZ} \setminus \underline{\quad}^Z = \left\{ (\mathcal{R}, \mathcal{R}) \mid \begin{array}{l} \mathcal{R} \subseteq \mathcal{I} \\ \exists (\mathcal{X}, \mathcal{Y}) \in \mathbf{A} : \\ \mathcal{X} \subseteq \mathcal{R} \\ \mathcal{Y} \subseteq \mathcal{R} \end{array} \right\} \subseteq \underline{\quad}^{(0)Z}. \quad (\text{B-29})$$

Die restlichen Elemente von  $\underline{\quad}^{mZ}$  sind in  $\underline{\quad}^Z$  enthalten. Damit gilt  $\underline{\quad}^{mZ} \subseteq \underline{\quad}^{(0)Z} \cup \underline{\quad}^Z$ . Zusammen mit  $\underline{\quad}^{(0)Z} \subseteq \underline{\quad}^{*Z}$  und  $\underline{\quad}^Z \subseteq \underline{\quad}^{*Z}$  (Gl. B-4) sowie der Transitivität von  $\underline{\quad}^{*Z}$  (Gl. B-6) folgt daraus  $\underline{\quad}^{mZ} \subseteq \underline{\quad}^{*Z}$ . Mit  $\underline{\quad}^{(0)mZ} = \underline{\quad}^{(0)Z} \subseteq \underline{\quad}^{*Z}$  folgen  $\underline{\quad}^{(i)mZ} \subseteq \underline{\quad}^{*Z}$  für alle  $i \in \mathbb{N}_0$  (Gl. B-27) und  $\underline{\quad}^{*mZ} \subseteq \underline{\quad}^{*Z}$ .

Aus  $\underline{\quad}^{*mZ} \subseteq \underline{\quad}^{*Z}$  und  $\underline{\quad}^{*Z} \subseteq \underline{\quad}^{*mZ}$  (1. Teil des Beweises) folgt schließlich  $\underline{\quad}^{*mZ} = \underline{\quad}^{*Z}$ . □

**Verwendung** Satz B.19, Lemma B.23, Lemma B.28, Lemma B.30

**Lemma B.17** Für alle mehrfachen Zuordnungen  $\mathcal{R} \underline{\quad}^{*mZ} \mathcal{S}$  und alle Mengen  $\mathcal{T} \subseteq \mathcal{I}$  gibt es die mehrfache Zuordnung  $(\mathcal{R} \cup \mathcal{T}) \underline{\quad}^{*mZ} (\mathcal{S} \cup \mathcal{T})$ .

Für alle mehrfachen Zuordnungen  $\mathcal{R} \stackrel{*mZ}{\mid} \mathcal{S}$  und alle Obermengen  $\mathcal{T}$  der Konfiguration  $\mathcal{R}$  gibt es die mehrfache Zuordnung  $\mathcal{T} \stackrel{*mZ}{\mid} (\mathcal{S} \cup \mathcal{T})$ . In diesem Fall gilt auch  $\mathcal{T} \stackrel{*mZ}{\mid} [\mathcal{S} \cup (\mathcal{T} \setminus \mathcal{R})]$ .

Für alle Zuordnungen  $\mathcal{R} \stackrel{*Z}{\mid} \mathcal{S}$  und alle Mengen  $\mathcal{T} \subseteq \mathcal{I}$  gibt es die Zuordnung  $(\mathcal{R} \cup \mathcal{T}) \stackrel{*Z}{\mid} (\mathcal{S} \cup \mathcal{T})$ .

Für alle Zuordnungen  $\mathcal{R} \stackrel{*Z}{\mid} \mathcal{S}$  und alle Obermengen  $\mathcal{T}$  der Konfiguration  $\mathcal{R}$  gibt es die Zuordnung  $\mathcal{T} \stackrel{*Z}{\mid} (\mathcal{S} \cup \mathcal{T})$ . In diesem Fall gilt auch  $\mathcal{T} \stackrel{*Z}{\mid} [\mathcal{S} \cup (\mathcal{T} \setminus \mathcal{R})]$ .

$$\begin{aligned} & \forall \mathcal{R}, \mathcal{S}, \mathcal{T} \subseteq \mathcal{I} : \\ & \mathcal{R} \stackrel{*mZ}{\mid} \mathcal{S} \rightarrow (\mathcal{R} \cup \mathcal{T}) \stackrel{*mZ}{\mid} (\mathcal{S} \cup \mathcal{T}) \\ & \mathcal{R} \stackrel{*Z}{\mid} \mathcal{S} \rightarrow (\mathcal{R} \cup \mathcal{T}) \stackrel{*Z}{\mid} (\mathcal{S} \cup \mathcal{T}) \\ & \left. \begin{array}{l} \mathcal{R} \stackrel{*mZ}{\mid} \mathcal{S} \\ \mathcal{R} \subseteq \mathcal{T} \end{array} \right\} \rightarrow \mathcal{T} \stackrel{*mZ}{\mid} (\mathcal{S} \cup \mathcal{T}) \\ & \left. \begin{array}{l} \mathcal{R} \stackrel{*Z}{\mid} \mathcal{S} \\ \mathcal{R} \subseteq \mathcal{T} \end{array} \right\} \rightarrow \mathcal{T} \stackrel{*Z}{\mid} (\mathcal{S} \cup \mathcal{T}) \end{aligned} \quad (\text{B} - 30)$$

**Beweis** Aus  $\mathcal{R} \stackrel{*mZ}{\mid} \mathcal{S}$  folgen  $\exists k \in \mathbb{N} : \exists Q_0, Q_1, \dots, Q_k \subseteq \mathcal{I}, \exists (\mathcal{X}_i, \mathcal{Y}_i) \in \mathbf{A} : \mathcal{X}_i \subseteq Q_i, Q_{i+1} = Q_i \cup \mathcal{Y}_i, i = 0, 1, \dots, k-1$  sowie  $\mathcal{R} = Q_0$  und  $\mathcal{S} = Q_k$  (Gl. B-3, B-26). Mit  $\mathcal{R} \cup \mathcal{T}$  gilt dann  $\exists k \in \mathbb{N} : \exists P_0, P_1, \dots, P_k \subseteq \mathcal{I}, P_i = Q_i \cup \mathcal{T}, \mathcal{X}_i \subseteq P_i, P_{i+1} = P_i \cup \mathcal{Y}_i, i = 0, 1, \dots, k-1$  sowie  $\mathcal{R} \cup \mathcal{T} = P_0$  und  $\mathcal{S} \cup \mathcal{T} = P_k$ , so dass  $(\mathcal{R} \cup \mathcal{T}) \stackrel{*mZ}{\mid} (\mathcal{S} \cup \mathcal{T})$  gilt (Gl. B-26).

Mit Lemma B.16 erhält man  $(\mathcal{R} \cup \mathcal{T}) \stackrel{*Z}{\mid} (\mathcal{S} \cup \mathcal{T})$ .

Mit  $\mathcal{R} \subseteq \mathcal{T}$  erhält man  $\mathcal{T} \stackrel{*mZ}{\mid} (\mathcal{S} \cup \mathcal{T})$  bzw.  $\mathcal{T} \stackrel{*Z}{\mid} (\mathcal{S} \cup \mathcal{T})$ . Mit  $\mathcal{R} \subseteq \mathcal{S}$  (Gl. B-7) gilt auch  $\mathcal{T} \stackrel{*mZ}{\mid} [\mathcal{S} \cup (\mathcal{T} \setminus \mathcal{R})]$  bzw.  $\mathcal{T} \stackrel{*Z}{\mid} [\mathcal{S} \cup (\mathcal{T} \setminus \mathcal{R})]$ .  $\square$

**Lemma B.18** Die Vereinigung zweier  $\mathcal{R}$  zugeordneter Konfigurationen  $\mathcal{S}$  und  $\mathcal{T}$  wird  $\mathcal{R}$  zugeordnet:

$$\begin{aligned} & \forall \mathcal{R}, \mathcal{S}, \mathcal{T} \subseteq \mathcal{I} : \\ & \left. \begin{array}{l} \mathcal{R} \stackrel{*Z}{\mid} \mathcal{S} \\ \mathcal{R} \stackrel{*Z}{\mid} \mathcal{T} \end{array} \right\} \rightarrow \mathcal{R} \stackrel{*Z}{\mid} (\mathcal{S} \cup \mathcal{T}) \end{aligned} \quad (\text{B} - 31)$$

**Beweis** 1. Teil: Es ist zu zeigen, dass für ein beliebiges  $i \in \mathbb{N}$  aus

$$\left. \begin{array}{l} (\mathcal{R}, \mathcal{S}) \in \stackrel{*Z}{\mid} \\ (\mathcal{R}, \mathcal{U}_i) \in \stackrel{(i)Z}{\mid} \end{array} \right\} \rightarrow (\mathcal{R}, \mathcal{S} \cup \mathcal{U}_i) \in \stackrel{*Z}{\mid} \quad (\text{B} - 32)$$

und  $(\mathcal{U}_i, \mathcal{U}_{i+1}) \in \overset{Z}{|}$  folgt:

$$\left. \begin{array}{l} (\mathcal{R}, \mathcal{S}) \in \overset{*Z}{|} \\ (\mathcal{R}, \mathcal{U}_{i+1}) \in \overset{(i+1)Z}{|} \end{array} \right\} \rightarrow (\mathcal{R}, \mathcal{S} \cup \mathcal{U}_{i+1}) \in \overset{*Z}{|} \quad (\text{B-33})$$

Wegen der Zuordnung  $\mathcal{U}_i \overset{Z}{|} \mathcal{U}_{i+1}$ , für die  $\exists(\mathcal{X}, \mathcal{Y}) \in \mathbf{A} : \mathcal{X} \subseteq \mathcal{U}_i \wedge \mathcal{U}_{i+1} = \mathcal{U}_i \cup \mathcal{Y}$  gilt (Gl. B-3), erhält man  $\mathcal{R} \overset{(i)Z}{|} \mathcal{U}_i \overset{Z}{|} \mathcal{U}_{i+1}$  (d. h.  $(\mathcal{R}, \mathcal{U}_{i+1}) \in \overset{(i+1)Z}{|}$ ). Aus  $\mathcal{X} \subseteq \mathcal{U}_i$  folgt  $\mathcal{X} \subseteq \mathcal{S} \cup \mathcal{U}_i$ , so dass  $\mathcal{S} \cup \mathcal{U}_i \cup \mathcal{Y} = \mathcal{S} \cup \mathcal{U}_{i+1}$  mit einer direkten mehrfachen Zuordnung (Gl. B-26) von  $\mathcal{S} \cup \mathcal{U}_i$  aus erreicht wird:  $\mathcal{S} \cup \mathcal{U}_i \overset{mZ}{|} \mathcal{S} \cup \mathcal{U}_{i+1}$ . Mit  $\overset{mZ}{|} \subseteq \overset{*mZ}{|} = \overset{*Z}{|}$  (Gl. B-20, Lemma B.16) gilt  $\mathcal{S} \cup \mathcal{U}_i \overset{*Z}{|} \mathcal{S} \cup \mathcal{U}_{i+1}$ , d. h. auch  $\mathcal{R} \overset{*Z}{|} \mathcal{S} \cup \mathcal{U}_{i+1}$  bzw.  $(\mathcal{R}, \mathcal{S} \cup \mathcal{U}_{i+1}) \in \overset{*Z}{|}$ . Aus Gleichung B-32 folgt also Gleichung B-33.

2. Teil: Mit  $\mathcal{U}_0 = \mathcal{R}$  und  $\mathcal{R} \subseteq \mathcal{S}$  (Gl. B-7) bzw.  $\mathcal{S} = \mathcal{S} \cup \mathcal{U}_0$  gilt Gleichung B-32 für  $i = 0$ . Mit dem 1. Teil gilt Gleichung B-32 für alle  $i \in \mathbb{N}$ . Mit Gleichung B-4 folgt daraus Gleichung B-31.  $\square$

**Verwendung Satz B.19**

**Satz B.19** Die Relationen *zuordnen*  $\overset{*Z}{|}$  und *parallel zuordnen*  $\overset{*\|Z}{|}$  sind identisch.

**Beweis** 1. Teil: Mit Gleichung B-19 gilt  $\overset{*Z}{|} = \overset{Z(1)}{|} \subseteq \overset{\|Z}{|}$ . Mit Gleichung B-20 erhält man  $\overset{*Z}{|} \subseteq \overset{(1)\|Z}{|}$  sowie  $\overset{*Z}{|} \subseteq \overset{*\|Z}{|}$ .

2. Teil: Sei  $\overset{Z(j)}{|} \subseteq \overset{*Z}{|}$  für ein beliebiges  $j \in \mathbb{N}$ , dann gilt mit Gleichung B-19

$$\overset{Z(j+1)}{|} = \left\{ (\mathcal{R}, (\mathcal{S} \cup \mathcal{T})) \mid \begin{array}{l} (\mathcal{R}, \mathcal{S}) \in \overset{Z(j)}{|} \subseteq \overset{*Z}{|} \\ (\mathcal{R}, \mathcal{T}) \in \overset{*Z}{|} \end{array} \right\}.$$

Mit Lemma B.18 erhält man  $\overset{Z(j+1)}{|} \subseteq \overset{*Z}{|}$ .

3. Teil: Für  $j = 1$  gilt  $\overset{Z(1)}{|} = \overset{*Z}{|}$  (Gl. B-19). Mit dem 2. Teil erhält man  $\overset{Z(j)}{|} \subseteq \overset{*Z}{|}$  für alle  $j \in \mathbb{N}$  und es gilt  $\overset{\|Z}{|} \subseteq \overset{*Z}{|}$ . Mit  $\overset{(0)\|Z}{|} = \overset{(0)Z}{|} \subseteq \overset{*Z}{|}$  erhält man  $\overset{(i)\|Z}{|} \subseteq \overset{*Z}{|}$  für alle  $i \in \mathbb{N}_0$  (Gl. B-6, Gl. B-20) sowie  $\overset{*\|Z}{|} \subseteq \overset{*Z}{|}$ . Mit  $\overset{*Z}{|} \subseteq \overset{*\|Z}{|}$  (1. Teil des Beweises) folgt daraus  $\overset{*Z}{|} = \overset{*\|Z}{|}$ .  $\square$

**Verwendung** Lemma B.21, Lemma B.29, Satz B.69

**Lemma B.20** Bei transitiven Relationen  $R \subseteq M \times M$  ist das Produkt einer reflexiven Teilmenge  $T \subseteq R$ , mit der vollständigen Relation  $R$  gleich der Relation  $R$ :

$$\left. \begin{array}{l} \forall R \subseteq M \times M : \\ \forall T \subseteq R : \\ R \text{ ist transitiv} \\ T \text{ ist reflexiv} \end{array} \right\} \rightarrow T \circ R = R \quad (\text{B} - 34)$$

**Beweis** 1. Fall:  $T \circ R \subseteq R$ :

Da  $R$  transitiv ist, gilt  $R \circ R \subseteq R$ . Mit  $T \subseteq R$  erhält man  $T \circ R \subseteq R$ .

2. Fall:  $R \subseteq T \circ R$ :

Da  $T$  reflexiv ist, gilt  $I \subseteq T$  mit  $I := \{(a, a) | a \in M\}$ . Damit gilt  $I \circ R \subseteq T \circ R$ . Zusammen mit  $I \circ R = R$  erhält man  $R \subseteq T \circ R$ .

Aus  $T \circ R \subseteq R$  (1. Fall) und  $R \subseteq T \circ R$  folgt  $R = T \circ R$ .  $\square$

**Verwendung** Satz B.22

**Lemma B.21** Für jedes Tupel  $W \in \bigtimes_{i=1}^k \wp(\mathbf{A})$ ;  $k \in \mathbb{N}$  gilt, dass die Relation mit dem Tupel  $W$  unabhängig verteilt zuordnen  $\left| \frac{v|Z}{W} \right|$  Teilmenge von zuordnen  $\left| \frac{*Z}{\phantom{W}} \right|$  ist und alle Paare  $(\mathcal{R}, \mathcal{R}), \mathcal{R} \subseteq \mathcal{I}$  enthält:

$$\left. \begin{array}{l} \forall k \in \mathbb{N} : \forall W \in \bigtimes_{i=1}^k \wp(\mathbf{A}) : \\ \left| \frac{v|Z}{W} \right| \subseteq \left| \frac{*Z}{\phantom{W}} \right| \\ \left| \frac{v|Z}{W} \right| \text{ ist reflexiv} \end{array} \right\} \quad (\text{B} - 35)$$

**Beweis** In Gleichung B-21 wird für jede der  $k$  Assoziationenmengen  $\mathbf{B}_j$  von  $W$  eine Zuordnung  $(\mathcal{U}_j, \mathcal{V}_j) \in \left| \frac{*(\mathcal{I}, \mathbf{B}_j)}{\phantom{W}} \right|$ ;  $j = 1, 2, \dots, k$  durchgeführt. Mit Lemma B.13 folgt aus  $(\mathcal{U}_j, \mathcal{V}_j) \in \left| \frac{*(\mathcal{I}, \mathbf{B}_j)}{\phantom{W}} \right|$  und  $\mathbf{B}_j \subseteq \mathbf{A}$ , dass  $(\mathcal{U}_j, \mathcal{V}_j) \in \left| \frac{*(\mathcal{I}, \mathbf{A})}{\phantom{W}} \right| = \left| \frac{*Z}{\phantom{W}} \right|$ . Mit  $\mathcal{R} := \bigcup_{j=1}^k \mathcal{U}_j$ , Lemma B.17 und  $\mathcal{U}_j \subseteq \mathcal{R}$  gibt es zu jeder Zuordnung  $\mathcal{U}_j \left| \frac{*Z}{\phantom{W}} \right| \mathcal{V}_j$  die Zuordnung  $\mathcal{R} \left| \frac{*Z}{\phantom{W}} \right| (\mathcal{V}_j \cup \mathcal{R})$ . Diese Zuordnungen  $\mathcal{R} \left| \frac{*Z}{\phantom{W}} \right| (\mathcal{V}_j \cup \mathcal{R})$  werden parallel durchgeführt (Gl. B-19), so dass auch  $\left( \mathcal{R}, \bigcup_{j=1}^k \mathcal{V}_j \cup \mathcal{R} \right) \in \left| \frac{**|Z}{\phantom{W}} \right| = \left| \frac{*Z}{\phantom{W}} \right|$  ist (Satz B.19). Mit  $\mathcal{U}_j \subseteq \mathcal{V}_j$  (Lemma B.3) erhält man  $\mathcal{R} \subseteq \bigcup_{j=1}^k \mathcal{V}_j$ . Mit  $\mathcal{S} := \bigcup_{j=1}^k \mathcal{V}_j$  erhält man  $\mathcal{S} = \bigcup_{j=1}^k \mathcal{V}_j \cup \mathcal{R}$  und es gilt  $(\mathcal{R}, \mathcal{S}) \in \left| \frac{*Z}{\phantom{W}} \right|$ , d.h.  $\left| \frac{v|Z}{W} \right| \subseteq \left| \frac{*Z}{\phantom{W}} \right|$ . Jede Menge der Zuordnungen  $\left| \frac{*(\mathcal{I}, \mathbf{B}_j)}{\phantom{W}} \right|$  enthält unabhängig von der Menge der Asso-

ziationen  $\mathbf{A}$  das Paar  $(\mathcal{U}_j, \mathcal{U}_j), \mathcal{U}_j \subseteq \mathcal{I}$ , so dass in  $\frac{v|Z}{W}$  die verteilte Zuordnung  $\left( \bigcup_{j=1}^k \mathcal{U}_j, \bigcup_{j=1}^k \mathcal{U}_j \right)$  enthalten ist und  $\frac{v|Z}{W}$  reflexiv ist (Gl. B-21).  $\square$

**Verwendung** Satz B.22, Satz B.71

**Satz B.22** Für jedes Tupel  $W \in \bigtimes_{i=1}^k \wp(\mathbf{A}); k \in \mathbb{N}$  gilt, dass die Relation *mit dem Tupel  $W$  verteilt zuordnen*  $\frac{v|Z}{W}$  mit *zuordnen*  $\frac{*Z}{\quad}$  identisch ist:

$$\forall k \in \mathbb{N} : \forall W \in \bigtimes_{i=1}^k \wp(\mathbf{A}) : \quad (B-36)$$

$$\frac{v|Z}{W} = \frac{*Z}{\quad}$$

**Beweis** Da  $\frac{*Z}{\quad}$  transitiv (Lemma B.2),  $\frac{v|Z}{W}$  reflexive Teilmenge von  $\frac{*Z}{\quad}$  (Lemma B.21) und das Produkt einer reflexiven Teilmenge einer transitiven Relation mit der vollständigen Relation gleich der vollständigen Relation ist (Lemma B.20), erhält man  $\frac{v|Z}{W} \circ \frac{*Z}{\quad} = \frac{*Z}{\quad}$  bzw. mit Gleichung B-22  $\frac{v|Z}{W} = \frac{v|Z}{W} \circ \frac{*Z}{\quad} = \frac{*Z}{\quad}$ .  $\square$

**Verwendung** Lemma B.25

**Lemma B.23** Die Relationen *zuordnen*  $\frac{*Z}{\quad}$  und *direkt asynchron zuordnen*  $\frac{a||Z}{\quad}$  sind identisch.

**Beweis** 1. Teil: Zu jeder Zuordnung  $(\mathcal{R}, \mathcal{S}) \in \frac{*Z}{\quad}$  gibt es ein  $k \in \mathbb{N}$  mit  $(\mathcal{R}, \mathcal{S}) \in \frac{(k)Z}{\quad}$  (Gl. B-4). Dazu gehört eine Folge direkter Zuordnungen  $\mathcal{R} = Q_0 \xrightarrow{Z} Q_1 \xrightarrow{Z} \dots \xrightarrow{Z} Q_k = \mathcal{S}$ . Beim Schritt  $Q_{i-1} \xrightarrow{Z} Q_i; i = 1, 2, \dots, k$  wird jeweils die Konfiguration  $Q_{i-1}$  zur Überprüfung der Assoziationen  $(\mathcal{X}_i, \mathcal{Y}_i)$  verwendet, so dass in Gleichung B-24 mit  $j = i - 1$  die Aussage  $\exists j \in \{0, 1, \dots, i - 1\} : \mathcal{X}_i \subseteq Q_j$  gilt. In diesem Fall wird mit der oben angegebenen Folge direkter Zuordnungen Gleichung B-24 erfüllt, so dass aus  $(\mathcal{R}, \mathcal{S}) \in \frac{*Z}{\quad}$  die Aussage  $(\mathcal{R}, \mathcal{S}) \in \frac{a||Z}{\quad}$  folgt. Es gilt also  $\frac{*Z}{\quad} \subseteq \frac{a||Z}{\quad}$ .

2. Teil: Zu jeder asynchronen Zuordnung  $(\mathcal{R}, \mathcal{S}) \in \frac{a||Z}{\quad}$  gibt es ein  $k \in \mathbb{N}$  sowie eine Folge von Assoziationen  $(\mathcal{X}_i, \mathcal{Y}_i), i = 0, \dots, k - 1$  und Konfigurationen  $Q_i \subseteq \mathcal{I}$  mit  $Q_{i+1} = Q_i \cup \mathcal{Y}_i$  (Gl. B-24). Es gilt also für alle  $i$

$$Q_j \subseteq Q_i; j = 0, 1, \dots, i, \quad (B-37)$$

so dass aus  $\mathcal{X}_i \subseteq \mathcal{Q}_j$  mit  $j \leq i$  folgt  $\mathcal{X}_i \subseteq \mathcal{Q}_i$ . Damit gilt für jede Assoziation  $(\mathcal{X}_i, \mathcal{Y}_i)$  in Gleichung B-24 mit  $\forall i \in \{0, \dots, k-1\} : \exists j \in \{0, \dots, i\} : \mathcal{X}_i \subseteq \mathcal{Q}_j$  die Aussage  $\mathcal{X}_i \subseteq \mathcal{Q}_i$ , so dass es für jede Assoziation  $(\mathcal{X}_i, \mathcal{Y}_i)$  eine direkte mehrfache Zuordnung  $\mathcal{Q}_i \xrightarrow{mZ} \mathcal{Q}_{i+1}$  gibt. Jede direkte asynchrone Zuordnung  $\mathcal{R} \xrightarrow{a||Z} \mathcal{S}$  ist also eine Folge direkter mehrfacher Zuordnungen  $\mathcal{Q}_i \xrightarrow{mZ} \mathcal{Q}_{i+1}$ . Diese Folge definiert eine mehrfache Zuordnung  $\mathcal{R} \xrightarrow{*mZ} \mathcal{S}$ , die gleichzeitig eine Zuordnung  $\mathcal{R} \xrightarrow{*Z} \mathcal{S}$  ist (Lemma B.16). Es gilt also  $\xrightarrow{a||Z} \subseteq \xrightarrow{*Z}$ .

Mit  $\xrightarrow{*Z} \subseteq \xrightarrow{a||Z}$  (1. Teil des Beweises) und  $\xrightarrow{a||Z} \subseteq \xrightarrow{*Z}$  erhält man  $\xrightarrow{*Z} = \xrightarrow{a||Z}$ .  $\square$

**Verwendung Satz B.24**

**Satz B.24** Die Relationen *zuordnen*  $\xrightarrow{*Z}$  und *asynchron zuordnen*  $\xrightarrow{*a||Z}$  sind identisch.

**Beweis** Da die Relationen *zuordnen* und *direkt asynchron zuordnen* identisch sind (Lemma B.23) und die Relation *zuordnen* transitiv ist (Lemma B.2), sind auch die Relationen *zuordnen* und *asynchron zuordnen* identisch.  $\square$

**Verwendung Satz B.69**

**Lemma B.25** Statt die Zuordnung  $(\mathcal{R} \cup \mathcal{T}) \xrightarrow{*Z} \mathcal{V}$  auszuführen, können die Zuordnungen  $\mathcal{R} \xrightarrow{*Z} \mathcal{S}$  und  $(\mathcal{S} \cup \mathcal{T}) \xrightarrow{*Z} \mathcal{V}$  ausgeführt werden:

$$\forall \mathcal{R}, \mathcal{S}, \mathcal{T}, \mathcal{V} \subseteq \mathcal{I} : \left. \begin{array}{l} \mathcal{R} \xrightarrow{*Z} \mathcal{S} \\ (\mathcal{S} \cup \mathcal{T}) \xrightarrow{*Z} \mathcal{V} \end{array} \right\} \rightarrow (\mathcal{R} \cup \mathcal{T}) \xrightarrow{*Z} \mathcal{V} \quad (\text{B} - 38)$$

**Beweis** Mit Lemma B.17 gilt

$$\forall \mathcal{R}, \mathcal{S}, \mathcal{T} \subseteq \mathcal{I} : \mathcal{R} \xrightarrow{*Z} \mathcal{S} \rightarrow (\mathcal{R} \cup \mathcal{T}) \xrightarrow{*Z} (\mathcal{S} \cup \mathcal{T}),$$

so dass  $(\mathcal{R} \cup \mathcal{T}) \xrightarrow{*Z} (\mathcal{S} \cup \mathcal{T}) \xrightarrow{*Z} \mathcal{V}$  bzw.  $(\mathcal{R} \cup \mathcal{T}) \xrightarrow{*Z} \mathcal{V}$  gilt.  $\square$

**Verwendung Satz B.26**

**Satz B.26** Die Relationen *kaskadiert zuordnen* und *zuordnen* sind identisch.

**Beweis** 1. Teil: Wird in der 1. Zeile von Gleichung B-23  $l = 1$  gesetzt, dann gibt es nur genau eine Konfiguration  $\mathcal{U}_1 = \mathcal{R}$ . In diesem Fall definiert Gleichung B-23  $\xrightarrow{*Z}$ , so dass in diesem Fall  $\xrightarrow{*Z} = \xrightarrow{kZ}$  gilt. Daraus folgt  $\xrightarrow{*Z} \subseteq \xrightarrow{kZ}$ .

2. Teil: Sei für ein beliebiges  $i \in \mathbb{N}$  die Aussage  $\bigcup_{j=1}^i \mathcal{U}_j \xrightarrow{*Z} \mathcal{W}_i$  wahr.

Mit  $\mathcal{R}' := \bigcup_{j=1}^i \mathcal{U}_j$ ,  $\mathcal{S}' := \mathcal{W}_i$ ,  $\mathcal{T}' := \mathcal{U}_{i+1}$  und  $\mathcal{V}' := \mathcal{W}_{i+1}$  sowie Lemma B.25 folgt aus  $\bigcup_{j=1}^i \mathcal{U}_j \mid^{*Z} \mathcal{W}_i$  [entspricht  $\mathcal{R}' \mid^{*Z} \mathcal{S}'$ ] und  $(\mathcal{W}_i \cup \mathcal{U}_{i+1}) \mid^{*Z} \mathcal{W}_{i+1}$  [entspricht  $(\mathcal{S}' \cup \mathcal{T}') \mid^{*Z} \mathcal{V}'$ ] die Aussage  $\left( \bigcup_{j=1}^i \mathcal{U}_j \right) \cup \mathcal{U}_{i+1} \mid^{*Z} \mathcal{W}_{i+1}$  bzw.  $\bigcup_{j=1}^{i+1} \mathcal{U}_j \mid^{*Z} \mathcal{W}_{i+1}$  [entspricht  $(\mathcal{R}' \cup \mathcal{T}') \mid^{*Z} \mathcal{V}'$ ].

3. Teil: Für  $i = 1$  gilt in Gleichung B-23  $\mathcal{W}_{i-1} = \mathcal{W}_0 = \emptyset$ . Damit erhält man aus der Aussage  $(\mathcal{U}_i \cup \mathcal{W}_{i-1}) \mid^{*Z} \mathcal{W}_i$  die Aussage  $(\mathcal{U}_1 \cup \mathcal{W}_0) \mid^{*Z} \mathcal{W}_1$  bzw.  $\mathcal{U}_1 \mid^{*Z} \mathcal{W}_1$ , so dass  $\bigcup_{j=1}^i \mathcal{U}_j \mid^{*Z} \mathcal{W}_i$  für  $i = 1$  zutrifft. Mit dem 2. Teil des Beweises gilt die Aussage  $\bigcup_{j=1}^i \mathcal{U}_j \mid^{*Z} \mathcal{W}_i$  für alle  $i \in \mathbb{N}$ .

In Gleichung B-23 folgt also für alle  $l \in \mathbb{N}$  und alle Konfigurationenmengen  $\{\mathcal{U}_1, \mathcal{U}_2, \dots, \mathcal{U}_l\} \subseteq \wp(\mathcal{I})$  aus  $\mathcal{W}_0 := \emptyset$ ,  $\forall i = 1, 2, \dots, l : (\mathcal{U}_i \cup \mathcal{W}_{i-1}) \mid^{*Z} \mathcal{W}_i$ ,  $\mathcal{R} := \bigcup_{j=1}^l \mathcal{U}_j$  und  $\mathcal{S} := \mathcal{W}_l$  die Aussage  $\mathcal{R} \mid^{*Z} \mathcal{S}$ .  $\square$

Verwendung Satz B.68

### B.1.3 Vollständige Zuordnungen

**Definition B.27** Für alle Mengen von Assoziationen  $\mathbf{A}$  mit  $k$  Assoziationen wird mit dem Z-System  $Z(\mathcal{I}, \mathbf{A})$  der Konfiguration  $\mathcal{R}$  die Konfiguration  $\mathcal{S}$  *direkt vollständig zugeordnet*, in Zeichen  $\mathcal{R} \mid^{\forall Z} \mathcal{S}$ , wenn mit jeder Assoziation  $(\mathcal{X}_i, \mathcal{Y}_i)$ ,  $i = 0, 1, \dots, k-1$  von  $\mathbf{A}$  eine abschließende Zuordnung  $Q_i \mid^{\dagger(\mathcal{I}, \{(\mathcal{X}_i, \mathcal{Y}_i)\})} Q_{i+1}$  durchgeführt wird und  $\mathcal{R} = Q_0$  sowie  $\mathcal{S} = Q_k$  sind:

$$\mid^{\forall Z} := \left\{ (\mathcal{R}, \mathcal{S}) \in \wp(\mathcal{I}) \times \wp(\mathcal{I}) \mid \begin{array}{l} \forall \mathbf{A} \subseteq \wp(\mathcal{I}) \times \wp(\mathcal{I}) : \\ \exists k \in \mathbb{N} : \\ \exists \{(\mathcal{X}_0, \mathcal{Y}_0), (\mathcal{X}_1, \mathcal{Y}_1), \dots, (\mathcal{X}_{k-1}, \mathcal{Y}_{k-1})\} = \mathbf{A} : \\ \forall i \in \{0, 1, \dots, k-1\} : \\ Q_i \mid^{\dagger(\mathcal{I}, \{(\mathcal{X}_i, \mathcal{Y}_i)\})} Q_{i+1} \\ \mathcal{R} = Q_0 \\ \mathcal{S} = Q_k \end{array} \right\} \quad (\text{B} - 39)$$

Hinweis zu Gl. B-39: Mit  $\exists \{(\mathcal{X}_0, \mathcal{Y}_0), (\mathcal{X}_1, \mathcal{Y}_1), \dots, (\mathcal{X}_{k-1}, \mathcal{Y}_{k-1})\} = \mathbf{A}$  : wird für jede Permutation der Elemente von  $\mathbf{A}$  ein Paar  $(\mathcal{R}, \mathcal{S})$  definiert.

Für alle Mengen von Assoziationen  $\mathbf{A}$  mit  $k$  Assoziationen wird mit dem Z-System  $Z(\mathcal{I}, \mathbf{A})$  der Konfiguration  $\mathcal{R}$  die Konfiguration  $\mathcal{S}$  *direkt vollständig parallel zugeordnet*, in Zeichen  $\mathcal{R} \mid^{\forall \parallel Z} \mathcal{S}$ , wenn mit jeder Assoziation  $(\mathcal{X}_i, \mathcal{Y}_i)$ ,  $i =$



$0, 1, \dots, k-1$  von  $\mathbf{A}$  eine abschließende Zuordnung  $\mathcal{R} \xrightarrow{\dagger(\mathcal{I}, \{(\mathcal{X}_i, \mathcal{Y}_i)\})} \mathcal{Q}_{i+1}$  durchgeführt wird und  $\mathcal{S}$  die Vereinigung aller Konfigurationen  $\mathcal{Q}_i$  ist:

$$\xrightarrow{\forall \mathcal{I} \parallel \mathcal{Z}} := \left\{ (\mathcal{R}, \mathcal{S}) \in \wp(\mathcal{I}) \times \wp(\mathcal{I}) \mid \begin{array}{l} \forall \mathbf{A} \subseteq \wp(\mathcal{I}) \times \wp(\mathcal{I}) : \\ \exists k \in \mathbb{N} : \\ \exists \{(\mathcal{X}_0, \mathcal{Y}_0), (\mathcal{X}_1, \mathcal{Y}_1), \dots, (\mathcal{X}_{k-1}, \mathcal{Y}_{k-1})\} = \mathbf{A} : \\ \forall i \in \{1, 2, \dots, k\} : \\ \exists \mathcal{Q}_i \subseteq \mathcal{I} : \\ \mathcal{R} \xrightarrow{\dagger(\mathcal{I}, \{(\mathcal{X}_i, \mathcal{Y}_i)\})} \mathcal{Q}_i \\ \mathcal{S} = \bigcup_{i=1}^k \mathcal{Q}_i \end{array} \right\} \quad (\text{B} - 40)$$

Für alle Mengen von Assoziationen  $\mathbf{A}$  mit  $k$  Assoziationen wird mit dem  $\mathcal{Z}$ -System  $\mathcal{Z}(\mathcal{I}, \mathbf{A})$  der Konfiguration  $\mathcal{R}$  die Konfiguration  $\mathcal{S}$  *direkt vollständig asynchron zugeordnet*, in Zeichen  $\mathcal{R} \xrightarrow{\forall \mathbf{a} \parallel \mathcal{Z}} \mathcal{S}$ , wenn es zu jeder Assoziation  $(\mathcal{X}_i, \mathcal{Y}_i)$ ,  $i = 0, 1, \dots, k-1$  von  $\mathbf{A}$  ein Paar von Konfigurationen  $(\mathcal{Q}_i, \mathcal{Q}_{i+1})$  und eine Konfiguration  $\mathcal{Q}_j$  mit  $j \leq i$  gibt, so dass bei Vorliegen von  $\mathcal{X}_i$  in  $\mathcal{Q}_j$  (d.h.  $\mathcal{X}_i \subseteq \mathcal{Q}_j$ ) die Konfiguration  $\mathcal{Q}_{i+1}$  durch Vereinigung von  $\mathcal{Q}_i$  und  $\mathcal{Y}_i$  entsteht, bei Nichtvorliegen ( $\mathcal{X}_i \not\subseteq \mathcal{Q}_j$ ) die Konfiguration  $\mathcal{Q}_{i+1}$  gleich  $\mathcal{Q}_i$  ist und  $\mathcal{R} = \mathcal{Q}_0$  sowie  $\mathcal{S} = \mathcal{Q}_k$  gilt:

$$\xrightarrow{\forall \mathbf{a} \parallel \mathcal{Z}} := \left\{ (\mathcal{R}, \mathcal{S}) \in \wp(\mathcal{I}) \times \wp(\mathcal{I}) \mid \begin{array}{l} \forall \mathbf{A} \subseteq \wp(\mathcal{I}) \times \wp(\mathcal{I}) : \\ \exists k \in \mathbb{N} : \\ \exists \{(\mathcal{X}_0, \mathcal{Y}_0), (\mathcal{X}_1, \mathcal{Y}_1), \dots, (\mathcal{X}_{k-1}, \mathcal{Y}_{k-1})\} = \mathbf{A} : \\ \forall i \in \{0, 1, \dots, k-1\} : \\ \exists \mathcal{Q}_i, \mathcal{Q}_{i+1} \subseteq \mathcal{I} : \\ \exists j \in \{0, 1, \dots, i\} : \\ \exists \mathcal{Q}_j \subseteq \mathcal{I} : \\ \mathcal{Q}_{i+1} = \begin{cases} \mathcal{Q}_i \cup \mathcal{Y}_i & \text{falls } \mathcal{X}_i \subseteq \mathcal{Q}_j \\ \mathcal{Q}_i & \text{sonst} \end{cases} \\ \mathcal{R} = \mathcal{Q}_0 \\ \mathcal{S} = \mathcal{Q}_k \end{array} \right\} \quad (\text{B} - 41)$$

Hinweis zu Gl. B-41: Mit  $\exists \{(\mathcal{X}_0, \mathcal{Y}_0), (\mathcal{X}_1, \mathcal{Y}_1), \dots, (\mathcal{X}_{k-1}, \mathcal{Y}_{k-1})\} = \mathbf{A}$  : wird für jede Permutation der Elemente von  $\mathbf{A}$  ein Paar  $(\mathcal{R}, \mathcal{S})$  definiert.  $\square$

**Lemma B.28** Die Relation *direkt vollständig zuordnen* ist Teilmenge von *zuordnen*:

$$\xrightarrow{\forall \mathcal{Z}} \subseteq \xrightarrow{* \mathcal{Z}} .$$

**Beweis** Der Vergleich der Gleichungen B-3, B-4 und B-5 mit Gleichung B-39 zeigt, dass  $\xrightarrow{\forall \mathcal{Z}}$  Teilmenge von  $\xrightarrow{* \mathcal{Z}}$  ist: Zu jeder *direkten vollständigen Zuordnung*  $(\mathcal{R}, \mathcal{S}) \in \xrightarrow{\forall \mathcal{Z}}$  mit der Folge der Konfigurationenpaare  $(\mathcal{Q}_i, \mathcal{Q}_{i+1})$  gibt es eine Folge direkter Zuordnungen  $(\mathcal{Q}_i, \mathcal{Q}_{i+1}) \in \xrightarrow{(\mathcal{I}, \{(\mathcal{X}_i, \mathcal{Y}_i)\})}$ . Mit  $(\mathcal{X}_i, \mathcal{Y}_i) \in \mathbf{A}$  und

Lemma B.13 gilt  $(Q_i, Q_{i+1}) \in \underline{\quad}^{*Z}$ , so dass insgesamt eine Zuordnung  $(\mathcal{R}, \mathcal{S}) \in \underline{\quad}^{*Z}$  vorliegt.

**Verwendung** Lemma B.48

**Lemma B.29** Die Relation *direkt vollständig parallel zuordnen* ist Teilmenge von *zuordnen*:  $\underline{\quad}^{\forall \| Z} \subseteq \underline{\quad}^{*Z}$ .

**Beweis** Der Vergleich von Gleichung B-19 mit Gleichung B-40 zeigt, dass  $\underline{\quad}^{\forall \| Z}$  Teilmenge von  $\underline{\quad}^{\| Z}$  ist: Zu jeder *direkten vollständig parallelen Zuordnung*  $(\mathcal{R}, \mathcal{S}) \in \underline{\quad}^{\forall \| Z}$  mit der Folge der Konfigurationen  $Q_i$  gibt es eine Folge direkter Zuordnungen  $(\mathcal{R}, Q_i) \in \underline{\quad}^{(\mathcal{I}, \{(\mathcal{X}_i, \mathcal{Y}_i)\})}$ . Mit  $(\mathcal{X}_i, \mathcal{Y}_i) \in \mathbf{A}$  und Lemma B.13 gilt  $(\mathcal{R}, Q_i) \in \underline{\quad}^{*Z}$ .

Wenn die Vereinigung der Konfigurationen  $Q_i$  mit  $\mathcal{S}$  bezeichnet wird, liegt eine *direkte parallele Zuordnung*  $(\mathcal{R}, \mathcal{S}) \in \underline{\quad}^{\| Z}$  vor (Gl. B-19) mit  $\underline{\quad}^{\| Z} \subseteq \underline{\quad}^{*\| Z} = \underline{\quad}^{*Z}$  (Gl. B-20, Satz B.19).  $\square$

**Verwendung** Lemma B.48

**Lemma B.30** Die Relation *direkt vollständig asynchron zuordnen* ist Teilmenge von *zuordnen*:  $\underline{\quad}^{\forall a \| Z} \subseteq \underline{\quad}^{*Z}$ .

**Beweis** Der Vergleich von Gleichung B-27 und Gleichung B-26 mit Gleichung B-41 zeigt, dass  $\underline{\quad}^{\forall a \| Z}$  Teilmenge von  $\underline{\quad}^{*mZ}$  ist: Zu jeder *direkten vollständig asynchronen Zuordnung*  $(\mathcal{R}, \mathcal{S}) \in \underline{\quad}^{\forall a \| Z}$  gibt es eine Folge von Schritten, die entweder direkte mehrfache Zuordnungen  $(Q_i, Q_{i+1}) \in \underline{\quad}^{m(\mathcal{I}, \{(\mathcal{X}_i, \mathcal{Y}_i)\})}$  (falls  $\mathcal{X}_i \subseteq Q_j \subseteq Q_i, j < i$ , Gl. B-37) oder Zuweisungen  $(Q_i, Q_i) \in \underline{\quad}^{(0)m(\mathcal{I}, \{(\mathcal{X}_i, \mathcal{Y}_i)\})}$  (falls  $\mathcal{X}_i \not\subseteq Q_j \subseteq Q_i, j < i$ , Gl. B-37) sind. Mit Gl. B-27 gilt  $(Q_i, Q_{i+1}) \in \underline{\quad}^{*m(\mathcal{I}, \{(\mathcal{X}_i, \mathcal{Y}_i)\})}$ . Mit Lemma B.16 erhält man  $(Q_i, Q_{i+1}) \in \underline{\quad}^{*(\mathcal{I}, \{(\mathcal{X}_i, \mathcal{Y}_i)\})}$ . Mit Lemma B.13 und  $(\mathcal{X}_i, \mathcal{Y}_i) \in \mathbf{A}$  gilt  $(Q_i, Q_{i+1}) \in \underline{\quad}^{*Z}$ . Damit liegt eine Zuordnung  $(\mathcal{R}, \mathcal{S}) \in \underline{\quad}^{*Z}$  vor.  $\square$

**Verwendung** Lemma B.48

### B.1.4 Zuordnungssysteme mit Konzepten

**Definition B.31** Ein *Zuordnungssystem mit Konzepten* (kurz: ZK-System) ist ein Tripel  $Z_K = (\mathcal{I}, \mathbf{P}, i)$  mit

$$\begin{aligned} \mathcal{I} &: \text{Menge von Instanzen} \\ \mathbf{P} &: \text{endliche Relation der Konzept-Assoziationen .} \\ i &: \text{Instanzierungsfunktion} \end{aligned} \tag{B - 42}$$

Die Elemente der *Relation der Konzept-Assoziationen*  $\mathbf{P} \subseteq \wp(\mathcal{I}) \times \wp(\mathcal{I})$  (kurz: Menge der Produktionen)  $(\mathcal{A}, \mathcal{B}) \in \mathbf{P}$  werden geschrieben  $\mathcal{A} \stackrel{\mathbf{P}}{\sim} \mathcal{B}$ ; in Worten: die Konzeptkonfiguration (kurz: K-Konfiguration)  $\mathcal{A}$  *assoziiert* die Konzeptkonfiguration  $\mathcal{B}$ . Die Elemente der Menge  $\mathcal{I}_K$  mit

$$\mathcal{I}_K(\mathbf{P}) := \bigcup_{(\mathcal{A}, \mathcal{B}) \in \mathbf{P}} (\mathcal{A} \cup \mathcal{B}) \quad (\text{B} - 43)$$

sind die *Konzept-Repräsentanten* (kurz: *Konzepte*) des ZK-Systems.

Für die *Instanziierungsfunktion* (kurz: I-Funktion)  $i$  gilt

$$i : \wp(\mathcal{I}) \times \wp(\mathcal{I}_K) \times \wp(\mathcal{I}_K) \rightarrow \wp(\mathcal{I}) \\ (\mathcal{X}, \mathcal{A}, \mathcal{B}) \mapsto i(\mathcal{B}|\mathcal{A} \mapsto \mathcal{X}) \quad (\text{B} - 44)$$

In Worten:  $i(\mathcal{B}|\mathcal{A} \mapsto \mathcal{X})$  ist die *Instanzkonfiguration* für die *Konzeptkonfiguration*  $\mathcal{B}$ , wenn die *Konzeptkonfiguration*  $\mathcal{A}$  für die *Instanzkonfiguration*  $\mathcal{X}$  steht (kurz:  $i$  von  $\mathcal{B}$  mit  $\mathcal{A}$  für  $\mathcal{X}$ ).

Mit der Instanziierungsfunktion  $i$  wird für jeden Konzept-Repräsentanten  $\alpha \in \mathcal{I}_K$  die Konzept-Menge  $\mathcal{I}_\alpha$  definiert:

$$\mathcal{I}_\alpha : \mathcal{I}_K \rightarrow \wp(\mathcal{I}) \\ \alpha \mapsto \mathcal{I}_\alpha(\mathcal{I}, \mathbf{P}, i) \\ \mathcal{I}_\alpha(\mathcal{I}, \mathbf{P}, i) := \left\{ \gamma \left| \begin{array}{l} \exists \mathcal{X} \subseteq \mathcal{I} : \\ \exists (\mathcal{A}, \mathcal{B}) \in \mathbf{P} : \\ \gamma \in i(\{\alpha\}|\mathcal{A} \mapsto \mathcal{X}) \text{ falls } \alpha \in \mathcal{A} \\ \gamma \in i(\{\alpha\}|\mathcal{B} \mapsto \mathcal{X}) \text{ falls } \alpha \in \mathcal{B} \end{array} \right. \right\} \quad (\text{B} - 45)$$

$\mathbf{A}_K$  ist die *Menge der Instanz-Assoziationen* des ZK-Systems  $Z_K(\mathcal{I}, \mathbf{P}, i)$ :

$$\mathbf{A}_K(\mathcal{I}, \mathbf{P}, i) := \left\{ (\mathcal{X}, \mathcal{Y}) \left| \begin{array}{l} \mathcal{X} \subseteq \mathcal{I} \\ \exists (\mathcal{A}, \mathcal{B}) \in \mathbf{P} : \\ i(\mathcal{B}|\mathcal{A} \mapsto \mathcal{X}) \neq \emptyset \\ \mathcal{Y} = i(\mathcal{B}|\mathcal{A} \mapsto \mathcal{X}) \end{array} \right. \right\} \quad (\text{B} - 46)$$

$Z(\mathcal{I}, \mathbf{A}_K(\mathcal{I}, \mathbf{P}, i))$  ist das durch  $Z_K(\mathcal{I}, \mathbf{P}, i)$  definierte Z-System mit Instanzen. Die Menge der *instanziierten Konzepte*  $\mathcal{I}_i$  ist mit Gleichung B-2:

$$\mathcal{I}_i(\mathcal{I}, \mathbf{P}, i) := \mathcal{I}_a(\mathbf{A}_K(\mathcal{I}, \mathbf{P}, i)) \quad (\text{B} - 47)$$

$\mathbf{A}_\mathcal{X}$  ist die *Menge der Instanz-Assoziationen mit der Instanzkonfiguration*  $\mathcal{X}$  des ZK-Systems  $Z_K(\mathcal{I}, \mathbf{P}, i)$ :

$$\mathbf{A}_\mathcal{X} : \mathcal{I} \rightarrow \wp[\wp(\mathcal{I}) \times \wp(\mathcal{I})] \\ (\mathcal{X}) \mapsto \mathbf{A}_\mathcal{X}(\mathcal{I}, \mathbf{P}, i) \\ \mathbf{A}_\mathcal{X}(\mathcal{I}, \mathbf{P}, i) := \{(\mathcal{U}, \mathcal{V}) \in \mathbf{A}_K(\mathcal{I}, \mathbf{P}, i) | \mathcal{U} = \mathcal{X}\} \quad (\text{B} - 48)$$

□

**Korollar B.32**

$$\mathbf{A}_K(\mathcal{I}, \mathbf{P}, i) = \left\{ (\mathcal{X}, \mathcal{Y}) \left| \begin{array}{l} \mathcal{X} \subseteq \mathcal{I}_i(\mathcal{I}, \mathbf{P}, i) \\ \exists (\mathcal{A}, \mathcal{B}) \in \mathbf{P} : \\ i(\mathcal{B}|\mathcal{A} \mapsto \mathcal{X}) \neq \emptyset \\ \mathcal{Y} = i(\mathcal{B}|\mathcal{A} \mapsto \mathcal{X}) \end{array} \right. \right\} \quad (\text{B} - 49)$$

**Beweis** Die Gleichungen B-49 und B-46 unterscheiden sich nur durch  $\mathcal{X} \subseteq \mathcal{I}_i(\mathcal{I}, \mathbf{P}, i)$  (Gl. B-49) und  $\mathcal{X} \subseteq \mathcal{I}$  (Gl. B-46). Da die Konfigurationen  $\mathcal{X}$  und  $\mathcal{Y}$  aller Assoziationen  $(\mathcal{X}, \mathcal{Y})$  von  $\mathbf{A}_K(\mathcal{I}, \mathbf{P}, i)$  Teilmengen von  $\mathcal{I}_i(\mathcal{I}, \mathbf{P}, i)$  sind (Gl. B-47), folgt aus  $\forall (\mathcal{X}, \mathcal{Y}) \in \mathbf{A}_K : \mathcal{X} \subseteq \mathcal{I}$  die Aussage  $\mathcal{X} \subseteq \mathcal{I}_i(\mathcal{I}, \mathbf{P}, i)$ . Mit  $\mathcal{I}_i(\mathcal{I}, \mathbf{P}, i) \subseteq \mathcal{I}$  folgt aus  $\mathcal{X} \subseteq \mathcal{I}_i(\mathcal{I}, \mathbf{P}, i)$  auch  $\mathcal{X} \subseteq \mathcal{I}$ .  $\square$

**B.1.5 Zuordnungssysteme mit Ableitungen**

**Definition B.33** Ein *Zuordnungssystem mit Ableitungen* (kurz: ZA-System) ist ein Tripel  $Z_A = (\mathbf{I}, \mathbf{P}, i)$  mit

$$\begin{array}{l} \mathbf{I} = (M_1, M_2, \dots, M_m) \\ M_i : i\text{-te Merkmalmenge, } i = 2, 3, \dots, m \\ T : \text{Typ-Merkmal} \\ \text{mit } M_1 = T \\ \mathbf{P} : \text{endliche Relation der Konzept-Assoziationen} \\ i : \text{Instanzierungsfunktion} \end{array} \quad (\text{B} - 50)$$

Mit der Menge der *Merkmalkvektor-Instanzen*  $\mathcal{I}_M$  (kurz: Menge der Merkmalkvektoren)

$$\mathcal{I}_M := M_1 \times M_2 \times \dots \times M_m, \quad (\text{B} - 51)$$

der Menge der Indices  $I_X$

$$I_X := \mathbb{N} \quad (\text{B} - 52)$$

und der Familie der Indexmengen  $P_M$

$$P_M := \wp(I_X) \quad (\text{B} - 53)$$

wird für das System  $Z_A(\mathbf{I}, \mathbf{P}, i)$  die Menge der *abgeleiteten Merkmalkvektor-Instanzen*  $\mathcal{I}_A$  (kurz: Menge der Ableitungen) definiert:

$$\mathcal{I}_A := \mathcal{I}_M \times I_X \times P_M \quad (\text{B} - 54)$$

$Z_K(\mathcal{I}_A, \mathbf{P}, i)$  ist das durch  $Z_A(\mathbf{I}, \mathbf{P}, i)$  definierte Z-System mit Konzepten;  $Z(\mathcal{I}_A, \mathbf{A}_K(\mathcal{I}_A, \mathbf{P}, i))$  ist das durch  $Z_A(\mathbf{I}, \mathbf{P}, i)$  definierte Z-System mit Instanzen (Gl. B-46).

Das ZA-System  $Z_A(\mathbf{I}, \mathbf{P}, i)$  heißt genau dann *wohldefiniert* (kurz: wZA-System) und wird dann geschrieben  $wZ_A(\mathbf{I}, \mathbf{P}, i)$ , wenn folgende Aussagen gelten:

1. Für alle Assoziationen  $(\mathcal{X}, \mathcal{Y}) \in \mathbf{A}_K(\mathcal{I}_A, \mathbf{P}, i)$  gilt, dass die Instanzen der Konfiguration  $\mathcal{X}$  mit jeder Ableitung  $\beta \in \mathcal{Y}$  *verzeitert* (vgl. Korollar B.34) sind:

$$\begin{aligned} \forall (\mathcal{X}, \mathcal{Y}) \in \mathbf{A}_K(\mathcal{I}_A, \mathbf{P}, i) : \\ \forall \beta \in \mathcal{Y} : \\ \beta^{(P_M)} = \mathcal{X}^{(I_X)} \end{aligned} \quad (\text{B} - 55)$$

2. Für alle Zuordnungen  $\mathcal{K} \xrightarrow{*Z} \mathcal{L}$  gilt, dass Instanzen von  $\mathcal{L}$  eindeutig indiziert sind:

$$\begin{aligned} \forall \mathcal{K}, \mathcal{L} \subseteq \mathcal{I}_A : \\ \mathcal{K} \xrightarrow{*Z} \mathcal{L} \rightarrow \left\{ \begin{array}{l} \forall \alpha \in \mathcal{L} : \\ \forall \beta \in \mathcal{L} \setminus \{\alpha\} : \\ \alpha^{(I_X)} \neq \beta^{(I_X)} \end{array} \right. \end{aligned} \quad (\text{B} - 56)$$

Zum Zugriff auf Konfigurationen  $\subseteq \mathcal{I}_A$  über Indexmengen  $I \subseteq I_X$  wird die Funktion  $\mathcal{F}$  verwendet:

$$\begin{aligned} \mathcal{F} : I_X &\rightarrow \mathcal{I}_A \\ I &\mapsto \mathcal{F}(I) \\ \mathcal{F}(I) &:= \{\beta \mid \beta^{(I_X)} \in I\} \end{aligned} \quad (\text{B} - 57)$$

Im Folgenden werden nur noch wohldefinierte ZA-Systeme betrachtet, so dass nicht *w*ZA-System, sondern verkürzt ZA-System und nicht *w* $Z_A(\mathbf{I}, \mathbf{P}, i)$ , sondern  $Z_A(\mathbf{I}, \mathbf{P}, i)$  geschrieben wird.

Das *Produktionsnetz*  $PN$  eines ZA-Systems  $Z_A(\mathbf{I}, \mathbf{P}, i)$  ist das Paar  $(\mathcal{I}_K^{(T)} \cup \mathbf{P}, PN_E)$  mit:

$$\begin{aligned} PN_E := & \left\{ ((\mathcal{A}, \mathcal{B}), \gamma) \in \mathbf{P} \times \mathcal{I}_K^{(T)} \mid \begin{array}{l} \exists \beta \in \mathcal{B} : \\ \beta^{(T)} = \gamma \end{array} \right\} \cup \\ & \cup \left\{ (\gamma, (\mathcal{A}, \mathcal{B})) \in \mathcal{I}_K^{(T)} \times \mathbf{P} \mid \begin{array}{l} \exists \alpha \in \mathcal{A} : \\ \alpha^{(T)} = \gamma \end{array} \right\} \end{aligned} \quad (\text{B} - 58)$$

Der *Ableitungsgraph*  $AG$  einer Ableitung  $\gamma \in \mathcal{I}_A$  ist das Paar  $(AG_N(\gamma), AG_E(\gamma))$  mit:

$$\begin{aligned} AG_N : \mathcal{I}_A &\rightarrow \wp(\mathcal{I}_A) \\ \gamma &\mapsto AG_N(\gamma) \\ AG_N^0(\gamma) &:= \{\gamma\} \\ AG_N^{i+1}(\gamma) &:= \bigcup_{\alpha \in AG_N^i(\gamma)} \mathcal{F}(\alpha^{(P_M)}), i \in \mathbb{N}_0 \\ AG_N(\gamma) &:= \bigcup_{i \in \mathbb{N}_0} AG_N^i(\gamma) \end{aligned} \quad (\text{B} - 59)$$

$$\begin{aligned} AG_E : \mathcal{I}_A &\rightarrow \wp(\mathcal{I}_A \times \mathcal{I}_A) \\ \gamma &\mapsto AG_E(\gamma) \\ AG_E^i(\gamma) &:= \left\{ (\alpha, \beta) \mid \begin{array}{l} \alpha \in AG_N^i(\gamma) \\ \beta \in \mathcal{F}(\alpha^{(P_M)}) \end{array} \right\}, i \in \mathbb{N}_0 \\ AG_E(\gamma) &:= \bigcup_{i \in \mathbb{N}_0} AG_E^i(\gamma) \end{aligned} \quad (\text{B} - 60)$$

Der *Beschreibungsgraph* einer Ableitung  $\gamma \in \mathcal{I}_A$  ist das Paar  $(BG_N(\gamma), BG_E(\gamma))$  mit:

$$\begin{aligned} CN : \mathcal{I}_A &\rightarrow \wp(\mathcal{I}_A) \\ \gamma &\mapsto CN(\gamma) \\ CN^0(\gamma) &:= \{\gamma\} \\ CN^{i+1}(\gamma) &:= \bigcup_{\alpha \in CN^i(\gamma)} \left\{ \beta \left| \begin{array}{l} \beta \in \mathcal{F}(\alpha^{(PM)}) \\ \beta^{(T)} = \gamma^{(T)} \end{array} \right. \right\}; i \in \mathbb{N}_0 \\ CN(\gamma) &:= \bigcup_{i \in \mathbb{N}_0} CN^i(\gamma) \end{aligned} \quad (\text{B-61})$$

$$\begin{aligned} FN : \mathcal{I}_A &\rightarrow \wp(\mathcal{I}_A) \\ \gamma &\mapsto FN(\gamma) \\ FN(\gamma) &:= \bigcup_{\alpha \in CN(\gamma)} \mathcal{F}(\alpha^{(PM)}) \setminus CN(\gamma) \end{aligned} \quad (\text{B-62})$$

$$\begin{aligned} BG_N : \mathcal{I}_A &\rightarrow \wp(\mathcal{I}_A) \\ \gamma &\mapsto BG_N(\gamma) \\ BG_N^0(\gamma) &:= CN(\gamma) \\ BG_N^{i+1}(\gamma) &:= \bigcup_{\alpha \in BG_N^i(\gamma)} FN(\alpha); i \in \mathbb{N}_0 \\ BG_N(\gamma) &:= \bigcup_{i \in \mathbb{N}_0} BG_N^i(\gamma) \end{aligned} \quad (\text{B-63})$$

$$\begin{aligned} BG_E : \mathcal{I}_A &\rightarrow \wp(\mathcal{I}_A \times \mathcal{I}_A) \\ \gamma &\mapsto BG_E(\gamma) \\ BG_E^i(\gamma) &:= \left\{ (\alpha, \beta) \left| \begin{array}{l} \alpha \in BG_N^i(\gamma) \\ \beta \in FN(\alpha) \end{array} \right. \right\}, i \in \mathbb{N}_0 \\ BG_E(\gamma) &:= \bigcup_{i \in \mathbb{N}_0} BG_E^i(\gamma) \end{aligned} \quad (\text{B-64})$$

Die Menge der instanziierten Konzepte  $\mathcal{I}_{iA}$  ist mit Gleichung B-47:

$$\mathcal{I}_{iA}(\mathcal{I}_A, \mathbf{P}, i) := \mathcal{I}_i(\mathcal{I}_A, \mathbf{P}, i) \quad (\text{B-65})$$

□

**Korollar B.34** Bei jeder direkten Zuordnung  $\mathcal{K} \stackrel{Z}{\dashv} \mathcal{L}$  mit der Assoziation  $(\mathcal{X}, \mathcal{Y}) \in \mathbf{A}$  sind die Instanzen der Konfiguration  $\mathcal{X}$  mit jeder Ableitung  $\beta \in \mathcal{Y}$  verzeigert:

$$\mathcal{K} \stackrel{Z}{\dashv} \mathcal{L} \rightarrow \left\{ \begin{array}{l} \forall \mathcal{K}, \mathcal{L} \subseteq \mathcal{I}_A : \\ \exists (\mathcal{X}, \mathcal{Y}) \in \mathbf{A}_K(\mathcal{I}_A, \mathbf{P}, i) : \\ \quad \mathcal{X} \subseteq \mathcal{K} \\ \quad \mathcal{L} := \mathcal{K} \cup \mathcal{Y} \\ \quad \forall \beta \in \mathcal{Y} : \\ \quad \beta^{(PM)} = \mathcal{X}^{(IX)} \end{array} \right. \quad (\text{B-66})$$

**Beweis** Aus  $\mathcal{K} \stackrel{Z}{\dashv} \mathcal{L}$  folgt  $\exists (\mathcal{X}, \mathcal{Y}) \in \mathbf{A}$  mit  $\mathcal{X} \subseteq \mathcal{K}$ ,  $\mathcal{Y} \not\subseteq \mathcal{K}$  und  $\mathcal{L} = \mathcal{K} \cup \mathcal{Y}$  (Gl. B-3). Aus Gleichung B-55 folgt damit Gleichung B-66. □

### B.1.6 Eigenschaften von Assoziationen

Assoziationen lassen sich wie anschließend aufgeführt einordnen. Dabei steht das Symbol  $\xrightarrow[\bullet]{*\mathbf{A}}$  entweder für die ersetzende Relation  $\xrightarrow[e]{*\mathbf{A}}$  oder die zuordnende Relation  $\xrightarrow[z]{*\mathbf{A}}$  und definiert die jeweilige Eigenschaft der Menge der Assoziationen  $\mathbf{A}$  als ersetzend oder zuordnend. In diesem Abschnitt werden die in Kapitel 2 definierten Symbole  $\xrightarrow[e]{*\mathbf{A}}$  (Gl. 2-7, 2-1) und  $\xrightarrow[z]{*\mathbf{A}}$  (Gl. 2-8, 2-1) für Relationen von Transformationssystemen verwendet; das in diesem Anhang definierte Symbol  $\xrightarrow[*Z]{*}$  ( $= \xrightarrow[z]{*\mathbf{A}}$ ) (Gl. B-3, B-4) wird nicht verwendet.

**Definition B.35** *Vexierende* Assoziationen:

$$\left. \begin{array}{l} \forall \mathbf{A} \subseteq \wp(\mathcal{I}) \times \wp(\mathcal{I}) : \\ \exists (\mathcal{X}, \mathcal{Y}) \in \mathbf{A} : \\ \forall \gamma \in \mathcal{X} : \\ \exists (\mathcal{U}, \mathcal{V}) \in \mathbf{A} \setminus \{(\mathcal{X}, \mathcal{Y})\} : \\ \gamma \in \mathcal{U} \end{array} \right\} \leftrightarrow \mathbf{A} \text{ ist vexierend} \quad (\text{B} - 67)$$

*Nicht verkleinernd analysierende* Assoziationen:

$$\left. \begin{array}{l} \forall \mathbf{A} \subseteq \wp(\mathcal{I}) \times \wp(\mathcal{I}) : \\ \exists (\mathcal{X}, \mathcal{Y}) \in \mathbf{A} : \\ |\mathcal{Y} \setminus \mathcal{X}| \geq |\mathcal{X} \setminus \mathcal{Y}| \end{array} \right\} \leftrightarrow \left\{ \begin{array}{l} \text{Die analysierende Menge der Assoziationen} \\ \mathbf{A} \text{ ist nicht verkleinernd.} \end{array} \right. \quad (\text{B} - 68)$$

*Lokal kontextfreie* Assoziationen:

$$\left. \begin{array}{l} \forall \mathbf{A} \subseteq \wp(\mathcal{I}) \times \wp(\mathcal{I}) : \\ \forall (\mathcal{X}, \mathcal{Y}) \in \mathbf{A} : \\ \mathcal{X} \cap \mathcal{Y} = \emptyset \end{array} \right\} \leftrightarrow \mathbf{A} \text{ ist lokal kontextfrei.} \quad (\text{B} - 69)$$

*Hierarchische* Assoziationen:

$$\left. \begin{array}{l} \forall \mathbf{A} \subseteq \wp(\mathcal{I}) \times \wp(\mathcal{I}) : \\ \forall (\mathcal{R}, \mathcal{S}) \in \xrightarrow[e]{+\mathbf{A}} : \\ \forall (\mathcal{X}, \mathcal{Y}) \in \mathbf{A} : \\ \forall \mathcal{K} \subseteq \mathcal{I} : \\ \mathcal{L} := (\mathcal{K} \setminus \mathcal{X}) \cup \mathcal{Y} \\ \forall \gamma \in \mathcal{X} \setminus \mathcal{Y} : \\ \mathcal{R} \xrightarrow[e]{*\mathbf{A}} \mathcal{K} \xrightarrow[e]{\mathbf{A}} \mathcal{L} \xrightarrow[e]{*\mathbf{A}} \mathcal{S} \rightarrow \gamma \notin \mathcal{S} \end{array} \right\} \leftrightarrow \mathbf{A} \text{ ist hierarchisch.} \quad (\text{B} - 70)$$

Assoziationen-Mengen, die nicht hierarchisch sind, heißen *heterarchisch*.

Global kontextfreie Assoziationen:

$$\left. \begin{array}{l} \forall \mathbf{A} \subseteq \wp(\mathcal{I}) \times \wp(\mathcal{I}) : \\ \forall (\mathcal{R}, \mathcal{S}) \in \xrightarrow[e]{+\mathbf{A}} : \\ \forall (\mathcal{X}, \mathcal{Y}) \in \mathbf{A} : \\ \forall \mathcal{K} \subseteq \mathcal{I} : \\ \mathcal{L} := (\mathcal{K} \setminus \mathcal{X}) \cup \mathcal{Y} \\ \forall \gamma \in \mathcal{X} : \\ \mathcal{R} \xrightarrow[e]{*\mathbf{A}} \mathcal{K} \xrightarrow[e]{\mathbf{A}} \mathcal{L} \xrightarrow[e]{*\mathbf{A}} \mathcal{S} \rightarrow \gamma \notin \mathcal{S} \end{array} \right\} \leftrightarrow \mathbf{A} \text{ ist global kontextfrei.} \quad (\text{B} - 71)$$

Assoziationen-Mengen, die nicht global kontextfrei sind, heißen *kontextsensitiv*.

Mehrfach assoziierende Assoziationen:

$$\left. \begin{array}{l} \forall \mathbf{A} \subseteq \wp(\mathcal{I}) \times \wp(\mathcal{I}) : \\ \forall (\mathcal{R}, \mathcal{S}) \in \xrightarrow[e]{+\mathbf{A}} : \\ \forall (\mathcal{X}, \mathcal{Y}) \in \mathbf{A} : \\ \forall \mathcal{K} \subseteq \mathcal{I} : \\ \mathcal{L} := (\mathcal{K} \setminus \mathcal{X}) \cup \mathcal{Y} \\ \mathcal{R} \xrightarrow[e]{*\mathbf{A}} \mathcal{K} \xrightarrow[e]{\mathbf{A}} \mathcal{L} \xrightarrow[e]{*\mathbf{A}} \mathcal{S} \rightarrow \mathcal{K} \cap (\mathcal{Y} \setminus \mathcal{X}) = \emptyset \end{array} \right\} \leftrightarrow \mathbf{A} \text{ ist einfach assoziierend.} \quad (\text{B} - 72)$$

Assoziationen-Mengen, die nicht einfach assoziierend sind, heißen *mehrfach assoziierend* (vgl. Gl. B-137).

Deterministische Assoziationen:

$$\left. \begin{array}{l} \forall \bullet \in \{e, z\} : \\ \forall \mathbf{A} \subseteq \wp(\mathcal{I}_A) \times \wp(\mathcal{I}_A) : \\ \forall \mathcal{R}, \mathcal{S} \subseteq \mathcal{I}_A : \\ \forall \gamma \in \mathcal{S} : \\ \forall \alpha \in \mathcal{S} \setminus \{\gamma\} : \\ \gamma^{(\mathcal{I}_M)} \neq \alpha^{(\mathcal{I}_M)} \end{array} \right\} \leftrightarrow \mathbf{A} \text{ ist } \left\{ \begin{array}{l} \text{falls } \bullet = e: \text{ ersetzend} \\ \text{falls } \bullet = z: \text{ zuordnend} \\ \text{deterministisch.} \end{array} \right. \quad (\text{B} - 73)$$

Merkmaldifferenzierende Assoziationen:

$$\left. \begin{array}{l} \forall \mathbf{A} \subseteq \wp(\mathcal{I}_A) \times \wp(\mathcal{I}_A) : \\ \forall \gamma_1, \gamma_2 \in \mathcal{I}_{iA} : \\ \gamma_1^{(\mathcal{I}_M)} = \gamma_2^{(\mathcal{I}_M)} \leftrightarrow \gamma_1 = \gamma_2 \end{array} \right\} \leftrightarrow \mathbf{A} \text{ ist merkmaldifferenzierend} \quad (\text{B} - 74)$$

Graphdifferenzierende Assoziationen:

$$\left. \begin{array}{l} \forall \mathbf{A} \subseteq \wp(\mathcal{I}_A) \times \wp(\mathcal{I}_A) : \\ \forall \gamma_1, \gamma_2 \in \mathcal{I}_{iA} : \\ \gamma_1^{(\mathcal{I}_M)} = \gamma_2^{(\mathcal{I}_M)} \\ AG_N(\gamma_1)^{(\mathcal{I}_M)} = AG_N(\gamma_2)^{(\mathcal{I}_M)} \\ AG_E(\gamma_1)^{(\mathcal{I}_M)} = AG_E(\gamma_2)^{(\mathcal{I}_M)} \end{array} \right\} \leftrightarrow \gamma_1 = \gamma_2 \left. \right\} \leftrightarrow \mathbf{A} \text{ ist graphdifferenzierend} \quad (\text{B} - 75)$$



Beschreibende Merkmale  $\mathbf{I}_a$ , Bereich  $B(\gamma^{\mathbf{I}_a})$  und Ausdehnung  $A(\gamma^{\mathbf{I}_a})$  einer Instanz  $\gamma \in \mathcal{I}$ :

$$\begin{aligned}
 & \forall \mathbf{I}_a = (M_1, M_2, \dots) \subseteq \mathbf{I} : \\
 & \quad \mathcal{I}_{M_a} := M_1 \times M_2 \times \dots \\
 & \quad \forall \odot \in \{B, A\} : \\
 & \quad \quad \odot : \mathcal{I}_{M_a} \rightarrow \mathcal{I}_{M_a} \\
 & \quad \quad (\alpha^{(\mathcal{I}_{M_a})}) \mapsto \odot(\alpha^{(\mathcal{I}_{M_a})}) \\
 & \left. \begin{aligned}
 & \forall \alpha \in \mathcal{I}^{(\mathcal{I}_{M_a})} : \exists! \beta \in \mathcal{I}_A : \alpha = \beta^{(\mathcal{I}_{M_a})} \\
 & \forall \alpha \in \mathcal{I}_A : \exists! \beta \in \mathcal{I}^{(\mathcal{I}_{M_a})} : \alpha^{(\mathcal{I}_{M_a})} = \beta \\
 & \quad \forall (\mathcal{X}, \mathcal{Y}) \in \mathbf{A} : \\
 & \quad \quad \forall \gamma \in \mathcal{Y} \setminus \mathcal{X} : \\
 & \quad \quad B(\gamma^{(\mathcal{I}_{M_a})}) = \bigcup_{\alpha \in \mathcal{X}} B(\alpha^{(\mathcal{I}_{M_a})}) \\
 & \quad \quad A(\gamma^{(\mathcal{I}_{M_a})}) = \bigcup_{\alpha \in \mathcal{X} \setminus \mathcal{Y}} A(\alpha^{(\mathcal{I}_{M_a})})
 \end{aligned} \right\} \leftrightarrow \left\{ \begin{aligned}
 & \mathbf{I}_a \text{ ist } \left\{ \begin{array}{l} \text{ausdehnungs- und} \\ \text{bereichsbeschreibend,} \end{array} \right. \\
 & B \text{ ist der Bereich und} \\
 & A \text{ ist die Ausdehnung.}
 \end{aligned} \right.
 \end{aligned}
 \tag{B - 76}$$

Anlagernde Assoziationen:

$$\begin{aligned}
 & \forall \mathbf{I}_a = (M_1, M_2, \dots) \subseteq \mathbf{I} : \\
 & \quad \mathcal{I}_{M_a} := M_1 \times M_2 \times \dots \\
 & \quad \forall \mathbf{A} \subseteq \wp(\mathcal{I}_{M_a}) \times \wp(\mathcal{I}_{M_a}) : \\
 & \left. \begin{aligned}
 & \mathbf{I}_a \text{ ist beschreibend} \\
 & \quad \forall (\mathcal{X}, \mathcal{Y}) \in \mathbf{A} : \\
 & \quad \quad \forall \alpha \in \mathcal{X} \setminus \mathcal{Y} : \\
 & \quad \quad \forall \beta \in (\mathcal{X} \setminus \mathcal{Y}) \setminus \{\alpha\} : \\
 & \quad \quad \quad A(\alpha) \cap A(\beta) = \emptyset
 \end{aligned} \right\} \leftrightarrow \mathbf{A} \text{ ist anlagernd}
 \end{aligned}
 \tag{B - 77}$$

□

**Lemma B.36** Wenn  $\mathcal{I}_M$  endlich ist, dann ist die Menge der Projektionen der Ableitungsgraphen  $(AG_N(\gamma), AG_E(\gamma))$  auf  $\mathcal{I}_M$ , d. h.  $(AG_N(\gamma)^{(\mathcal{I}_M)}, AG_E(\gamma)^{(\mathcal{I}_M)})$  aller Instanzen  $\gamma \in \mathcal{I}_A$ , endlich.

**Beweis** Aus  $\forall \gamma \in \mathcal{I}_A : AG_N(\gamma)^{(\mathcal{I}_M)} \subseteq \mathcal{I}_M$  (Gl. B-59, B-166) folgt  $|AG_N(\gamma)^{(\mathcal{I}_M)}| \leq |\mathcal{I}_M|$ . Aus  $\forall \gamma \in \mathcal{I}_A : AG_E(\gamma)^{(\mathcal{I}_M)} \subseteq \mathcal{I}_M \times \mathcal{I}_M$  (Gl. B-60, B-167) folgt  $|AG_E(\gamma)^{(\mathcal{I}_M)}| \leq |\mathcal{I}_M|^2$ . Es gibt also maximal  $|\mathcal{I}_M|^3$  unterschiedliche Ableitungsgraphen  $(AG_E(\gamma)^{(\mathcal{I}_M)}, AG_N(\gamma)^{(\mathcal{I}_M)})$ . □

**Verwendung** Lemma B.37

**Lemma B.37** Die Menge der instanziierten Konzepte  $\mathcal{I}_{i_A}$  ist endlich, wenn die zugehörige Menge der Merkmalvektoren  $\mathcal{I}_M$  endlich ist und merkmaldifferenzierende oder graphdifferenzierende Mengen von Assoziationen  $\mathbf{A}$  vorliegen.

**Beweis** 1. Teil: Aus Gleichung B-166 folgt  $|\mathcal{I}_{i_A}^{(\mathcal{I}_M)}| \leq |\mathcal{I}_M|$ . Aus Gleichung B-74 folgt für merkmaldifferenzierende Assoziationen  $|\mathcal{I}_{i_A}| = |\mathcal{I}_{i_A}^{(\mathcal{I}_M)}|$ , d. h.  $|\mathcal{I}_{i_A}| = |\mathcal{I}_{i_A}^{(\mathcal{I}_M)}| \leq |\mathcal{I}_M|$ , so dass die Menge  $\mathcal{I}_{i_A}$  bei merkmaldifferenzierenden Mengen von Assoziationen und endlichen Mengen der Merkmalvektoren  $\mathcal{I}_M$  endlich ist.

2. Teil: Aus Gleichung B-166 folgt  $|\mathcal{I}_{iA}^{(\mathcal{I}_M)}| \leq |\mathcal{I}_M|$ . Die Anzahl der Ableitungsgraphen sei hier  $n_A$ . Diese ist bei graph-differenzierenden Mengen von Assoziationen und endlichen Mengen der Merkmalvektoren  $\mathcal{I}_M$  endlich (Lemma B.36). Aus Gleichung B-75 folgt für graph-differenzierende Assoziationen  $|\mathcal{I}_{iA}| = |\mathcal{I}_{iA}^{(\mathcal{I}_M)}| \cdot n_A$ , so dass die Menge  $\mathcal{I}_{iA}$  bei graph-differenzierenden Mengen von Assoziationen und endlichen Mengen der Merkmalvektoren  $\mathcal{I}_M$  endlich ist.

Hinweis:  $\mathcal{I}_{iA}$  ist diejenige Teilmenge von  $\mathcal{I}_A$ , deren Instanzen bei Zuordnungen mit  $\mathbf{A}_K(\mathcal{I}, \mathbf{P}, i)$  verwendet werden (Gl. B-2, B-47, B-65). Die Menge der definierten Ableitungen  $\mathcal{I}_A$  ist wie die Indexmenge  $I_X = \mathbb{N}$  unendlich (Gl. B-54).  $\square$

**Verwendung** Satz B.38

**Satz B.38** Z-Systeme  $Z(\mathcal{I}, \mathbf{A})$  terminieren, wenn merkmal-differenzierende oder graph-differenzierende Mengen von Assoziationen  $\mathbf{A}$  vorliegen und die Menge der Merkmalvektoren  $\mathcal{I}_M$  endlich ist.

**Beweis**  $\mathcal{I}_{iA}$  ist in diesem Fall endlich (Lemma B.37) und nach Definition (Gl. B-65) die Menge der assoziierten Instanzen  $\mathcal{I}_a(\mathbf{A})$  (Gl. B-2), so dass  $Z(\mathcal{I}, \mathbf{A})$  terminiert (Satz B.4).  $\square$

**Lemma B.39** Lokal kontextfreie und hierarchische Mengen von Assoziationen sind global kontextfrei:

$$\begin{array}{l} \forall \mathbf{A} \subseteq \wp(\mathcal{I}_A) \times \wp(\mathcal{I}_A) : \\ \left. \begin{array}{l} \mathbf{A} \text{ ist lokal kontextfrei} \\ \mathbf{A} \text{ ist hierarchisch} \end{array} \right\} \rightarrow \mathbf{A} \text{ ist global kontextfrei} \end{array} \quad (\text{B} - 78)$$

**Beweis** Mit lokal kontextfreien Assoziationen  $(\mathcal{X}, \mathcal{Y}) \in \mathbf{A}$  gilt  $\mathcal{X} \cap \mathcal{Y} = \emptyset$  (Gl. B-69), so dass in diesem Fall  $\mathcal{X} = \mathcal{X} \setminus \mathcal{Y}$  gilt und aus  $\gamma \in \mathcal{X} \setminus \mathcal{Y}$  die Aussage  $\gamma \in \mathcal{X}$  folgt. Die für hierarchische Assoziationen geltende Aussage (Gl. B-70)

$$\forall \gamma \in \mathcal{X} \setminus \mathcal{Y} : \\ \mathcal{R} \xrightarrow[e]{*\mathbf{A}} \mathcal{K} \xrightarrow[e]{\mathbf{A}} \mathcal{L} \xrightarrow[e]{*\mathbf{A}} \mathcal{S} \rightarrow \gamma \notin \mathcal{S}$$

gilt also bei kontextfreien Assoziationen für alle Instanzen von  $\mathcal{X}$

$$\forall \gamma \in \mathcal{X} : \\ \mathcal{R} \xrightarrow[e]{*\mathbf{A}} \mathcal{K} \xrightarrow[e]{\mathbf{A}} \mathcal{L} \xrightarrow[e]{*\mathbf{A}} \mathcal{S} \rightarrow \gamma \notin \mathcal{S},$$

so dass lokal kontextfreie und hierarchische Assoziationen auch global kontextfrei sind (Gl. B-71).  $\square$

**Verwendung** Lemma B.40, Lemma B.41, Satz B.42

**Lemma B.40** Mengen von Assoziationen, die nicht lokal kontextfrei sind, sind kontextsensitiv.

$$\begin{array}{l} \forall \mathbf{A} \subseteq \wp(\mathcal{I}_A) \times \wp(\mathcal{I}_A) : \\ \mathbf{A} \text{ ist nicht lokal kontextfrei} \rightarrow \mathbf{A} \text{ ist kontextsensitiv} \end{array} \quad (\text{B} - 79)$$

**Beweis** In Gleichung B-71 wird für alle Instanzen  $\gamma \in \mathcal{X}$  der Konfigurationen  $\mathcal{X}$  global kontextfreier Assoziationen  $(\mathcal{X}, \mathcal{Y}) \in \mathbf{A}$  die Aussage  $\mathcal{R} \xrightarrow[e]{*\mathbf{A}} \mathcal{K} \xrightarrow[e]{\mathbf{A}} \mathcal{L} \xrightarrow[e]{*\mathbf{A}} \mathcal{S} \rightarrow \gamma \notin \mathcal{S}$  gemacht, die für hierarchische Assoziationen auf Instanzen beschränkt ist, die in  $\mathcal{X} \setminus \mathcal{Y}$  enthalten sind (Gl. B-70). Mit Assoziationen, die nicht lokal kontextfrei sind, ist  $\mathcal{X} \cap \mathcal{Y} \neq \emptyset$  (Gl. B-69) möglich, so dass auch  $\mathcal{X} \neq \mathcal{X} \setminus \mathcal{Y}$  möglich ist. Es kann also Instanzen  $\gamma \in \mathcal{X} \cap \mathcal{Y}$  (d. h.  $\gamma \in \mathcal{X}$  und  $\gamma \notin \mathcal{X} \setminus \mathcal{Y}$ ) geben, für die  $\mathcal{R} \xrightarrow[e]{*\mathbf{A}} \mathcal{K} \xrightarrow[e]{\mathbf{A}} \mathcal{L} \xrightarrow[e]{*\mathbf{A}} \mathcal{S} \rightarrow \gamma \in \mathcal{S}$  gilt, so dass nicht lokal kontextfreie Assoziationen kontextsensitiv sind.  $\square$

**Verwendung** Satz B.42

**Lemma B.41** *Heterarchische Mengen von Assoziationen sind kontextsensitiv.*

$$\begin{aligned} \forall \mathbf{A} \subseteq \wp(\mathcal{I}_A) \times \wp(\mathcal{I}_A) : \\ \mathbf{A} \text{ ist heterarchisch} \rightarrow \mathbf{A} \text{ ist kontextsensitiv} \end{aligned} \quad (\text{B} - 80)$$

**Beweis** In Gleichung B-71 wird für Instanzen  $\gamma \in \mathcal{X}$  der Konfigurationen  $\mathcal{X}$  der global kontextfreien Assoziationen  $(\mathcal{X}, \mathcal{Y}) \in \mathbf{A}$  die Aussage  $\mathcal{R} \xrightarrow[e]{*\mathbf{A}} \mathcal{K} \xrightarrow[e]{\mathbf{A}} \mathcal{L} \xrightarrow[e]{*\mathbf{A}} \mathcal{S} \rightarrow \gamma \notin \mathcal{S}$  gemacht, die für heterarchische Assoziationen bei Instanzen, die in  $\mathcal{X} \setminus \mathcal{Y}$  enthalten sind, nicht gelten muss (Gl. B-70). Es kann also Instanzen  $\gamma \in \mathcal{X} \setminus \mathcal{Y} \subseteq \mathcal{X}$  geben, für die  $\mathcal{R} \xrightarrow[e]{*\mathbf{A}} \mathcal{K} \xrightarrow[e]{\mathbf{A}} \mathcal{L} \xrightarrow[e]{*\mathbf{A}} \mathcal{S} \rightarrow \gamma \in \mathcal{S}$  gilt, so dass heterarchische Assoziationen kontextsensitiv sind.  $\square$

**Verwendung** Satz B.42

**Satz B.42** Mengen von Assoziationen sind genau dann *lokal kontextfrei* und *hierarchische*, wenn sie *global kontextfrei* sind.

**Beweis** Der 1. Teil  $\rightarrow$  folgt aus Lemma B.39. Der 2. Teil  $\leftarrow$  folgt aus Lemma B.40 und Lemma B.41.  $\square$

**Satz B.43** Analysierende Mengen von Assoziationen  $\mathbf{A}^a$  sind genau dann lokal kontextfrei, wenn die korrespondierenden generierenden Assoziationen  $\mathbf{A}^g = (\mathbf{A}^a)^{-1}$  lokal kontextfrei sind.

**Beweis** Mit  $\mathcal{X} \cup \mathcal{Y} = \emptyset = \mathcal{Y} \cup \mathcal{X}$  folgt aus der lokalen Kontextfreiheit generierender Assoziationen direkt die lokale Kontextfreiheit der zugehörigen analysierenden Assoziationen:

$$\left. \begin{aligned} \forall (\mathcal{X}, \mathcal{Y}) \in \mathbf{A}^g : \\ \mathcal{X} \cap \mathcal{Y} = \emptyset \end{aligned} \right\} \leftrightarrow \left\{ \begin{aligned} \forall (\mathcal{Y}, \mathcal{X}) \in \mathbf{A}^a = (\mathbf{A}^g)^{-1} : \\ \mathcal{Y} \cap \mathcal{X} = \emptyset \end{aligned} \right. \quad (\text{B} - 81)$$

$\square$

## B.2 Zuordnungsautomaten

### B.2.1 Zuordnungsautomaten mit Konzepten

**Definition B.44** Ein *Zuordnungsautomat* (kurz: Z-Automat) ist ein Paar  $A = ((\mathcal{I}, \mathbf{P}, i), a)$  mit

$$\begin{aligned} (\mathcal{I}, \mathbf{P}, i) &: \text{Z-System mit Konzepten} \\ \mathcal{I} &: \text{Menge von Instanzen} \\ \mathbf{P} &: \text{endliche Relation der Konzept-Assoziationen .} \\ i &: \text{Instanziierungsfunktion} \\ a &: \text{Auswahlfunktion} \end{aligned} \quad (\text{B} - 82)$$

Mit der Menge der Produktionen  $\mathbf{P}$  wird die für eine Instanzkonfiguration  $\mathcal{R}$  zu generierende *Menge der Überprüfungsaufträge*  $u(\mathcal{R})$  des Z-Automaten  $A((\mathcal{I}, \mathbf{P}, i), a)$  definiert:

$$\begin{aligned} u &: \wp(\mathcal{I}) \rightarrow \wp(\mathcal{I}) \times \mathbf{P} \\ \mathcal{R} &\mapsto u(\mathcal{R}) \\ u(\mathcal{R}) &:= [\mathcal{R} \cap \mathcal{I}_i(\mathcal{I}, \mathbf{P}, i)] \times \mathbf{P} \end{aligned} \quad (\text{B} - 83)$$

$\gamma$  ist die *Instanz-Referenz* (oder *triggernde Instanz*) und  $(\mathcal{A}, \mathcal{B})$  die *Produktion* des Überprüfungsauftrages  $(\gamma, (\mathcal{A}, \mathcal{B}))$ . Mengen von Überprüfungsaufträgen werden mit *großen lateinischen Buchstaben* bezeichnet.

Für die *Auswahlfunktion*  $a$  gilt

$$\begin{aligned} a &: \wp(\mathcal{I}) \times \wp(u(\mathcal{I})) \rightarrow \wp(u(\mathcal{I})) \\ (\mathcal{R}, U) &\mapsto a(\mathcal{R}, U) \end{aligned} \quad (\text{B} - 84)$$

(vgl. Gleichung B-91).

$\mathbf{A}_u(\gamma, (\mathcal{A}, \mathcal{B}))$  ist die *Menge der für den Überprüfungsauftrag*  $(\gamma, (\mathcal{A}, \mathcal{B}))$  *zu überprüfenden Instanz-Assoziationen* (kurz: uI-Assoziationen):

$$\begin{aligned} \mathbf{A}_u &: \mathcal{I} \times \mathbf{P} \rightarrow \wp[\wp(\mathcal{I}) \times \wp(\mathcal{I})] \\ (\gamma, (\mathcal{A}, \mathcal{B})) &\mapsto \mathbf{A}_u(\gamma, (\mathcal{A}, \mathcal{B})) \\ \mathbf{A}_u(\gamma, (\mathcal{A}, \mathcal{B})) &:= \left\{ (\mathcal{X}, \mathcal{Y}) \left| \begin{array}{l} \mathcal{X} \subseteq \mathcal{I}_i(\mathcal{I}, \mathbf{P}, i) \\ \gamma \in \mathcal{X} \\ i(\mathcal{B}|\mathcal{A} \mapsto \mathcal{X}) \neq \emptyset \\ \mathcal{Y} = i(\mathcal{B}|\mathcal{A} \mapsto \mathcal{X}) \end{array} \right. \right\} \end{aligned} \quad (\text{B} - 85)$$

Eine Konfiguration  $\mathcal{R}$  und eine Menge von Überprüfungsaufträgen  $U$  definieren den jeweiligen Zustand  $q = (\mathcal{R}, U)$  des Z-Automaten  $A((\mathcal{I}, \mathbf{P}, i), a)$ . Zustände werden mit *kleinen lateinischen Buchstaben* bezeichnet.

Ein Zustand  $r = (\mathcal{R}, U_r)$  eines Z-Automaten  $A((\mathcal{I}, \mathbf{P}, i), a)$  geht *direkt über* in den Zustand  $s = (\mathcal{S}, U_s)$ , in Zeichen  $r \stackrel{A}{\mapsto} s$ , wenn

1. mit der Auswahlfunktion  $a(\mathcal{R}, U_r)$  aus der nicht leeren Menge der Überprüfungsaufträge  $U_r$  eine Menge von Überprüfungsaufträgen  $U_p$  ausgewählt wird;

2. die Menge der Überprüfungsaufträge  $U_s$  die Überprüfungsaufträge der Menge  $U_p$  nicht mehr enthält und zusätzlich alle Überprüfungsaufträge enthält, die für die neuen Instanzen der Konfiguration  $\mathcal{S}$  generiert werden ( $u(\mathcal{S} \setminus \mathcal{R})$ );
3. der Konfiguration  $\mathcal{R}$  mit den Assoziationen der Überprüfungsaufträge von  $U_p$  vollständig die Konfiguration  $\mathcal{S}$  zugeordnet wird  $[(\mathcal{R}, \mathcal{S}) \in \frac{\forall(\mathcal{I}, \mathbf{A}_g)}{\quad}]$ :

$$\frac{A}{\quad} := \left\{ \begin{array}{l} ((\mathcal{R}, U_r), (\mathcal{S}, U_s)) \in \\ \in [\wp(\mathcal{I}) \times \wp(u(\mathcal{I}))]^2 \end{array} \left| \begin{array}{l} U_r \neq \emptyset \\ U_p := a(\mathcal{R}, U_r) \\ U_s = (U_r \setminus U_p) \cup u(\mathcal{S} \setminus \mathcal{R}) \\ \mathbf{A}_g := \bigcup_{(\gamma, (\mathcal{A}, \mathcal{B})) \in U_p} \mathbf{A}_u(\gamma, (\mathcal{A}, \mathcal{B})) \\ (\mathcal{R}, \mathcal{S}) \in \frac{\forall(\mathcal{I}, \mathbf{A}_g)}{\quad} \end{array} \right. \right\} \quad (\text{B} - 86)$$

Analog werden *geht direkt parallel über*  $\frac{\parallel A}{\quad}$  mit  $(\mathcal{R}, \mathcal{S}) \in \frac{\forall \parallel (\mathcal{I}, \mathbf{A}_g)}{\quad}$

$$\frac{\parallel A}{\quad} := \left\{ \begin{array}{l} ((\mathcal{R}, U_r), (\mathcal{S}, U_s)) \in \\ \in [\wp(\mathcal{I}) \times \wp(u(\mathcal{I}))]^2 \end{array} \left| \begin{array}{l} U_r \neq \emptyset \\ U_p := a(\mathcal{R}, U_r) \\ U_s = (U_r \setminus U_p) \cup u(\mathcal{S} \setminus \mathcal{R}) \\ \mathbf{A}_g := \bigcup_{(\gamma, (\mathcal{A}, \mathcal{B})) \in U_p} \mathbf{A}_u(\gamma, (\mathcal{A}, \mathcal{B})) \\ (\mathcal{R}, \mathcal{S}) \in \frac{\forall \parallel (\mathcal{I}, \mathbf{A}_g)}{\quad} \end{array} \right. \right\} \quad (\text{B} - 87)$$

und *geht direkt asynchron über*  $\frac{aA}{\quad}$  mit  $(\mathcal{R}, \mathcal{S}) \in \frac{\forall a \parallel (\mathcal{I}, \mathbf{A}_g)}{\quad}$

$$\frac{aA}{\quad} := \left\{ \begin{array}{l} ((\mathcal{R}, U_r), (\mathcal{S}, U_s)) \in \\ \in [\wp(\mathcal{I}) \times \wp(u(\mathcal{I}))]^2 \end{array} \left| \begin{array}{l} U_r \neq \emptyset \\ U_p := a(\mathcal{R}, U_r) \\ U_s = (U_r \setminus U_p) \cup u(\mathcal{S} \setminus \mathcal{R}) \\ \mathbf{A}_g := \bigcup_{(\gamma, (\mathcal{A}, \mathcal{B})) \in U_p} \mathbf{A}_u(\gamma, (\mathcal{A}, \mathcal{B})) \\ (\mathcal{R}, \mathcal{S}) \in \frac{\forall a \parallel (\mathcal{I}, \mathbf{A}_g)}{\quad} \end{array} \right. \right\} \quad (\text{B} - 88)$$

definiert.

Ein Zustand  $r$  eines Z-Automaten  $A((\mathcal{I}, \mathbf{P}, i), a)$  *geht über* in einen Zustand  $s$ , in Zeichen  $r \xrightarrow{*A} s$ , wenn mit  $r = q_0$  und  $s = q_k$  kein ( $k = 0$ ) direkter Zustandsübergang, ein ( $k = 1$ ) direkter Zustandsübergang oder mehrere ( $k > 1$ ) direkte

Zustandsübergänge  $q_i \stackrel{A}{=} q_{i+1}, i = 0, 1, \dots, k-1$  ausgeführt werden:

$$\begin{aligned} \stackrel{(0)A}{=} &:= \{(r, r) \mid r \in \wp(\mathcal{I}) \times \wp(u(\mathcal{I}))\} \\ \stackrel{(i+1)A}{=} &:= \left\{ (r, s) \in [\wp(\mathcal{I}) \times \wp(u(\mathcal{I}))]^2 \mid \begin{array}{l} \exists q_i \in \wp(\mathcal{I}) \times \wp(u(\mathcal{I})) : \\ (r, q_i) \in \stackrel{(i)A}{=} \\ (q_i, s) \in \stackrel{A}{=} \end{array} \right\}, i \in \mathbb{N}_0 \\ \stackrel{+A}{=} &:= \bigcup_{i \in \mathbb{N}} \stackrel{(i)A}{=} \\ \stackrel{*A}{=} &:= \stackrel{+A}{=} \cup \stackrel{(0)A}{=} \end{aligned} \tag{B-89}$$

Ein Zustand  $r$  eines Z-Automaten  $A((\mathcal{I}, \mathbf{P}, i), a)$  geht *abschließend über* in einen Zustand  $s$ , in Zeichen  $r \stackrel{\dagger A}{=} s$ , wenn der Zustand  $r$  in den Zustand  $s$  übergeht und der Z-Automat in keinen weiteren Zustand direkt übergehen kann:

$$\stackrel{\dagger A}{=} := \left\{ (r, s) \in \stackrel{*A}{=} \mid \begin{array}{l} \forall (u, v) \in \stackrel{A}{=} : \\ s \neq u \end{array} \right\} \tag{B-90}$$

Ein Z-Automat  $A((\mathcal{I}, \mathbf{P}, i), a)$  heißt genau dann *wohldefiniert* (kurz: wZ-Automat) und wird dann geschrieben  $wA((\mathcal{I}, \mathbf{P}, i), a)$ , wenn die Auswahlfunktion  $a(\mathcal{R}, U)$  so lange Überprüfungsaufträge aus  $U$  übergibt, wie  $U$  nicht leer ist:

$$\begin{aligned} \forall (\mathcal{R}, U) \in \wp(\mathcal{I}) \times \wp(u(\mathcal{I})) : \\ a(\mathcal{R}, U) \subseteq U \\ |U| > 0 \leftrightarrow |a(\mathcal{R}, U)| > 0 \end{aligned} \tag{B-91}$$

Im Folgenden werden nur noch wohldefinierte Z-Automaten betrachtet, so dass nicht wZ-Automat, sondern verkürzt Z-Automat und nicht  $wA((\mathcal{I}, \mathbf{P}, i), a)$ , sondern  $A((\mathcal{I}, \mathbf{P}, i), a)$  geschrieben wird.  $\square$

**Lemma B.45**  $\stackrel{*A}{=}$  ist transitiv:

$$\forall r, s, t \in \wp(\mathcal{I}) \times \wp(u(\mathcal{I})) : \left. \begin{array}{l} r \stackrel{*A}{=} s \\ s \stackrel{*A}{=} t \end{array} \right\} \rightarrow r \stackrel{*A}{=} t \tag{B-92}$$

**Beweis**  $\stackrel{*A}{=}$  ist transitive reflexive Hülle von  $\stackrel{A}{=}$  (Gl. B-89): Aus  $r \stackrel{*A}{=} s$  folgt  $\exists k \in \mathbb{N}_0 : (r, s) \in \stackrel{(k)A}{=}$ . Analog gilt  $\exists l \in \mathbb{N}_0 : (s, t) \in \stackrel{(l)A}{=}$ , so dass  $(r, t) \in \stackrel{(k+l)A}{=} \subseteq \stackrel{*A}{=}$  gilt.  $\square$

**Lemma B.46** Die Menge der Überprüfungsaufträge einer Konfiguration ist gleich der Vereinigung der Überprüfungsaufträge der Instanzen dieser Konfiguration:

$$\begin{aligned} \forall \mathcal{X} \subseteq \mathcal{I} : \\ u(\mathcal{X}) = \bigcup_{\gamma \in \mathcal{X}} u(\gamma) \end{aligned} \tag{B-93}$$

**Beweis**  $u(\mathcal{X}) = [\mathcal{X} \cap \mathcal{I}_i(\mathcal{I}, \mathbf{P}, i)] \times \mathbf{P} = \bigcup_{\gamma \in \mathcal{X}} ([\{\gamma\} \cap \mathcal{I}_i(\mathcal{I}, \mathbf{P}, i)] \times \mathbf{P}) = \bigcup_{\gamma \in \mathcal{X}} u(\gamma)$  (Gl. B-83).  $\square$

**Verwendung** Lemma B.57

**Satz B.47** Die Menge der mit Überprüfungsaufträgen überprüfbaren Assoziationen  $\bigcup_{(\gamma, (\mathcal{A}, \mathcal{B})) \in u(\mathcal{I})} \mathbf{A}_u(\gamma, (\mathcal{A}, \mathcal{B}))$  des Z-Automaten  $A((\mathcal{I}, \mathbf{P}, i), a)$  ist die Menge der Assoziationen  $\mathbf{A}_K(\mathcal{I}, \mathbf{P}, i)$  des ZK-Systems  $Z_K(\mathcal{I}, \mathbf{P}, i)$ .

**Beweis** Nach Definition von  $\mathbf{A}_u(\gamma, (\mathcal{A}, \mathcal{B}))$  (Gl. B-85) und  $u(\gamma)$  (Gl. B-83) gilt

$$\begin{aligned} & \bigcup_{(\gamma, (\mathcal{A}, \mathcal{B})) \in \mathcal{I}_i(\mathcal{I}, \mathbf{P}, i) \times \mathbf{P} = u(\mathcal{I})} \mathbf{A}_u(\gamma, (\mathcal{A}, \mathcal{B})) = \bigcup_{\gamma \in \mathcal{I}_i(\mathcal{I}, \mathbf{P}, i)} \left[ \bigcup_{(\mathcal{A}, \mathcal{B}) \in \mathbf{P}} \mathbf{A}_u(\gamma, (\mathcal{A}, \mathcal{B})) \right] = \\ & = \bigcup_{\gamma \in \mathcal{I}_i(\mathcal{I}, \mathbf{P}, i)} \left\{ (\mathcal{X}, \mathcal{Y}) \left| \begin{array}{l} \mathcal{X} \subseteq \mathcal{I}_i(\mathcal{I}, \mathbf{P}, i) \\ \gamma \in \mathcal{X} \\ \exists (\mathcal{A}, \mathcal{B}) \in \mathbf{P} : \\ i(\mathcal{B}|\mathcal{A} \mapsto \mathcal{X}) \neq \emptyset \\ \mathcal{Y} = i(\mathcal{B}|\mathcal{A} \mapsto \mathcal{X}) \end{array} \right. \right\} = \left\{ (\mathcal{X}, \mathcal{Y}) \left| \begin{array}{l} \mathcal{X} \subseteq \mathcal{I}_i(\mathcal{I}, \mathbf{P}, i) \\ \exists (\mathcal{A}, \mathcal{B}) \in \mathbf{P} : \\ i(\mathcal{B}|\mathcal{A} \mapsto \mathcal{X}) \neq \emptyset \\ \mathcal{Y} = i(\mathcal{B}|\mathcal{A} \mapsto \mathcal{X}) \end{array} \right. \right\} \end{aligned} \quad (\text{B} - 94)$$

Diese Menge stimmt mit  $\mathbf{A}_K(\mathcal{I}, \mathbf{P}, i)$  überein (Gl. B-49).  $\square$

**Verwendung** Lemma B.48

**Lemma B.48** Wenn der Z-Automat  $A((\mathcal{I}, \mathbf{P}, i), a)$  vom Zustand  $(\mathcal{R}, U_r)$  aus

- (i) *direkt* in einen Zustand  $(\mathcal{S}, U_s)$  übergeht  $(\mathcal{R}, U_r) \stackrel{A}{\models} (\mathcal{S}, U_s)$  oder
  - (ii) *direkt parallel* in einen Zustand  $(\mathcal{S}, U_s)$  übergeht  $(\mathcal{R}, U_r) \stackrel{\parallel A}{\models} (\mathcal{S}, U_s)$  oder
  - (iii) *direkt asynchron* in einen Zustand  $(\mathcal{S}, U_s)$  übergeht  $(\mathcal{R}, U_r) \stackrel{aA}{\models} (\mathcal{S}, U_s)$ ,
- wird der Konfiguration  $\mathcal{R}$  durch das Z-System  $\widehat{Z} := (\mathcal{I}, \mathbf{A}_K(\mathcal{I}, \mathbf{P}, i))$  die Konfiguration  $\mathcal{S}$  zugeordnet  $\mathcal{R} \stackrel{* \widehat{Z}}{\mapsto} \mathcal{S}$ :

$$\begin{aligned} & \forall \bullet \in \{A, \parallel A, a\} : \\ & \forall (\mathcal{R}, U_r), (\mathcal{S}, U_s) \in \wp(\mathcal{I}) \times \wp(u(\mathcal{I})) : \\ & (\mathcal{R}, U_r) \stackrel{\bullet}{\models} (\mathcal{S}, U_s) \rightarrow \mathcal{R} \stackrel{* \widehat{Z}}{\mapsto} \mathcal{S} \end{aligned} \quad (\text{B} - 95)$$

**Beweis** Die beim Zustandsübergang überprüfte Menge der Hypothesen sei  $U_p$  (Gl. B-86 bis Gl. B-88). Mit  $U_p \subseteq u(\mathcal{I})$  gilt  $\mathbf{A}_g := \bigcup_{(\gamma, (\mathcal{A}, \mathcal{B})) \in U_p} \mathbf{A}_u(\gamma, (\mathcal{A}, \mathcal{B})) \subseteq$

$\bigcup_{(\gamma, (\mathcal{A}, \mathcal{B})) \in u(\mathcal{I})} \mathbf{A}_u(\gamma, (\mathcal{A}, \mathcal{B})) = \mathbf{A}_K(\mathcal{I}, \mathbf{P}, i)$  (Satz B.47). Mit  $Z_g := (\mathcal{I}, \mathbf{A}_g)$  und

(i)  $(\mathcal{R}, \mathcal{S}) \in \stackrel{\forall Z_g}{\mapsto}$  (Gl. B-86) bzw.

(ii)  $(\mathcal{R}, \mathcal{S}) \in \stackrel{\forall \parallel Z_g}{\mapsto}$  (Gl. B-87) bzw.

(iii)  $(\mathcal{R}, \mathcal{S}) \in \stackrel{\forall a \parallel Z_g}{\mapsto}$  (Gl. B-88) gilt  $(\mathcal{R}, \mathcal{S}) \in \stackrel{* Z_g}{\mapsto}$  (i: Lemma B.28 bzw. ii:

Lemma B.29 bzw. iii: Lemma B.30). Mit  $\mathbf{A}_g \subseteq \mathbf{A}_K(\mathcal{I}, \mathbf{P}, i)$  folgt  $(\mathcal{R}, \mathcal{S}) \in \stackrel{* \widehat{Z}}{\mapsto}$  (Lemma B.13).  $\square$

**Verwendung** Satz B.49, Lemma B.50, Satz B.59, Satz B.69

**Satz B.49** Wenn der Z-Automat  $A((\mathcal{I}, \mathbf{P}, i), a)$  vom Zustand  $(\mathcal{R}, U_r)$  aus

(i) in einen Zustand  $(\mathcal{S}, U_s)$  übergeht  $(\mathcal{R}, U_r) \stackrel{*A}{\Longrightarrow} (\mathcal{S}, U_s)$  oder

(ii) in einen Zustand  $(\mathcal{S}, U_s)$  parallel übergeht  $(\mathcal{R}, U_r) \stackrel{*||A}{\Longrightarrow} (\mathcal{S}, U_s)$  oder

(iii) in einen Zustand  $(\mathcal{S}, U_s)$  asynchron übergeht  $(\mathcal{R}, U_r) \stackrel{*aA}{\Longrightarrow} (\mathcal{S}, U_s)$  wird der Konfiguration  $\mathcal{R}$  durch das Z-System  $\widehat{Z} := (\mathcal{I}, \mathbf{A}_K(\mathcal{I}, \mathbf{P}, i))$  die Konfiguration  $\mathcal{S}$  zugeordnet  $\mathcal{R} \stackrel{*{\widehat{Z}}}{\Longrightarrow} \mathcal{S}$ :

$$\begin{aligned} & \forall \bullet \in \{*A, *||A, *a||A\} : \\ & \forall (\mathcal{R}, U_r), (\mathcal{S}, U_s) \in \wp(\mathcal{I}) \times \wp(u(\mathcal{I})) : \\ & (\mathcal{R}, U_r) \stackrel{\bullet}{\Longrightarrow} (\mathcal{S}, U_s) \rightarrow \mathcal{R} \stackrel{*{\widehat{Z}}}{\Longrightarrow} \mathcal{S} \end{aligned} \quad (\text{B} - 96)$$

**Beweis** Da die Aussage (i)  $(\mathcal{R}, U_r) \stackrel{A}{\Longrightarrow} (\mathcal{S}, U_s) \rightarrow \mathcal{R} \stackrel{*{\widehat{Z}}}{\Longrightarrow} \mathcal{S}$  bzw. (ii)  $(\mathcal{R}, U_r) \stackrel{||A}{\Longrightarrow} (\mathcal{S}, U_s) \rightarrow \mathcal{R} \stackrel{*{\widehat{Z}}}{\Longrightarrow} \mathcal{S}$  bzw. (iii)  $(\mathcal{R}, U_r) \stackrel{aA}{\Longrightarrow} (\mathcal{S}, U_s) \rightarrow \mathcal{R} \stackrel{*{\widehat{Z}}}{\Longrightarrow} \mathcal{S}$  für einen direkten Zustandsübergang gilt (Lemma B.48) und die Relation  $\stackrel{*Z}{\Longrightarrow}$  transitiv ist (Gl. B-6), gilt die Aussage auch für die Elemente der Relationen  $\stackrel{*A}{\Longrightarrow}$ ,  $\stackrel{*||A}{\Longrightarrow}$  und  $\stackrel{*aA}{\Longrightarrow}$ .  $\square$

**Verwendung** Satz B.61, Korollar B.64, Satz B.68

**Lemma B.50** Geht der Zustand  $r = (\mathcal{R}, U_r)$  eines Z-Automaten  $A((\mathcal{I}, \mathbf{P}, i), a)$  in einen Zustand  $s = (\mathcal{S}, U_s)$  über ( $r \stackrel{*A}{\Longrightarrow} s$ ), so werden Überprüfungsaufträge jeweils höchstens 1 Mal generiert.

**Beweis** Bei jedem direkten Zustandsübergang  $(Q_i, U_{q_i}) \stackrel{A}{\Longrightarrow} (Q_{i+1}, U_{q_{i+1}}), i = 0, 1, \dots, k-1$  mit  $Q_0 = \mathcal{R}$  und  $Q_k = \mathcal{S}$  werden mit  $U_{q_{i+1}} = (U_{q_i} \setminus U_p) \cup u(Q_{i+1} \setminus Q_i)$  (Gl. B-86 bis Gl. B-88) nur für neue Instanzen  $Q_{i+1} \setminus Q_i$  zusätzliche Überprüfungsaufträge  $u(Q_{i+1} \setminus Q_i)$  generiert. Bei einer Folge direkter Zustandsübergänge mit  $Q_j \subseteq Q_i, j < i; i, j \in \mathbb{N}_0$  (Lemma B.48 und Lemma B.3) ist in der Menge der zuletzt generierten Überprüfungsaufträge  $u(Q_{i+1} \setminus Q_i) = u\left(Q_{i+1} \setminus \left(\bigcup_{j=0}^i Q_j\right)\right)$  kein vorher bereits generierter Überprüfungsauftrag enthalten.  $\square$

**Verwendung** Satz B.61, Korollar B.64, Satz B.68

**Lemma B.51** Genau dann, wenn alle Zustände  $(\mathcal{R}, U_r)$  eines Z-Automaten  $((\mathcal{I}, \mathbf{P}, i), a)$  abschließend in einen Zustand  $(\mathcal{S}, \emptyset)$  übergehen, terminiert der Z-Automat:

$$\left. \begin{aligned} & \forall \mathcal{R} \subseteq \mathcal{I} : \\ & \forall U_r \subseteq u(\mathcal{I}) : \\ & \exists \mathcal{S} \subseteq \mathcal{I} : \\ & (\mathcal{R}, U_r) \stackrel{*A}{\Longrightarrow} (\mathcal{S}, \emptyset) \end{aligned} \right\} \leftrightarrow ((\mathcal{I}, \mathbf{P}, i), a) \text{ terminiert} \quad (\text{B} - 97)$$



**Beweis** 1. Teil:  $\rightarrow$ : Im Zustand  $(\mathcal{S}, \emptyset)$  ist die Menge der Überprüfungsaufträge leer, so dass keine weiteren Zustandsübergänge erfolgen (Gl. B-86 ff.) und der Z-Automat anhält. Dies gilt für alle Zustände  $(\mathcal{R}, U_r)$ , so dass der Z-Automat terminiert.

2. Teil:  $\leftarrow$ : Es wird angenommen, dass es einen Zustand  $(\mathcal{R}, U_r)$  gibt, von dem aus es eine Folge von Zustandsübergängen gibt, bei der jeder Zustand eine nicht leere Menge von Überprüfungsaufträgen besitzt, so dass der Z-Automat nicht in den Zustand  $(\mathcal{S}, U_s = \emptyset)$  übergeht. Daraus folgt, dass ausgehend von  $(\mathcal{R}, U_r)$  nach jedem Zustandsübergang weitere Zustandsübergänge erfolgen können (Gl. B-86 ff.), so dass es unendliche Folgen von Zustandsübergängen geben kann und der Z-Automat nicht terminiert. Dies ist ein Widerspruch zur Aussage, dass der Z-Automat terminiert, so dass die Annahme, dass  $U_s$  nicht leer ist, falsch ist.  $\square$

**Verwendung** Satz B.52, Satz B.61, Korollar B.63

**Satz B.52** Z-Automaten  $((\mathcal{I}, \mathbf{P}, i), a)$  mit endlichen Mengen instanzierter Konzepte  $\mathcal{I}_i(\mathcal{I}, \mathbf{P}, i)$  terminieren.

**Beweis** Mit jedem Zustandsübergang  $(\mathcal{R}, U_r) \xrightarrow{*A} (\mathcal{S}, U_s)$  wird der Konfiguration  $\mathcal{R}$  die Konfiguration  $\mathcal{S}$  zugeordnet (Satz B.49). Da  $\mathcal{I}_i(\mathcal{I}, \mathbf{P}, i)$  und  $\mathbf{P}$  endlich sind (Gl. B-82), ist auch die Menge aller Überprüfungsaufträge  $u(\mathcal{S}) = [\mathcal{S} \cap \mathcal{I}_i(\mathcal{I}, \mathbf{P}, i)] \times \mathbf{P}$  (Gl. B-83) endlich.

Da kein Überprüfungsauftrag mehrfach generiert wird (Lemma B.50) und bei jedem direkten Zustandsübergang mindestens ein Überprüfungsauftrag aus der Menge der Überprüfungsaufträge entfernt wird (Gl. B-86 bis Gl. B-88), ist nach einer endlichen Anzahl von Zustandsübergängen die Menge der Überprüfungsaufträge  $U_s$  leer.

$(\mathcal{R}, U_r)$  geht also nach  $(\mathcal{S}, \emptyset)$  über, so dass der Z-Automat  $((\mathcal{I}, \mathbf{P}, i), a)$  terminiert (Lemma B.51).  $\square$

**Verwendung** Satz B.53, Satz B.61

**Satz B.53** Z-Automaten  $((\mathcal{I}, \mathbf{P}, i), a)$  terminieren, wenn merkmaldifferenzierende oder graphdifferenzierende Mengen von Assoziationen  $\mathbf{A}_K(\mathcal{I}, \mathbf{P}, i)$  vorliegen und die Menge der Merkmalvektoren  $\mathcal{I}_M$  endlich ist.

**Beweis**  $\mathcal{I}_{iA}$  ist in diesem Fall endlich (Lemma B.37), so dass der Z-Automat  $((\mathcal{I}, \mathbf{P}, i), a)$  terminiert (Satz B.52).  $\square$

## B.2.2 Analysierbare Zustände

**Definition B.54** Der Zustand  $r = (\mathcal{R}, U_r)$  heißt *mit der Menge der Assoziationen*  $\mathbf{A}_K(\mathcal{I}, \mathbf{P}, i)$  *analysierbar*, in Zeichen  $\mathcal{R} \bowtie_{\mathbf{A}_K} U_r$ , wenn für alle Assoziationen  $(\mathcal{X}, \mathcal{Y}) \in \mathbf{A}_K(\mathcal{I}, \mathbf{P}, i)$  gilt, dass, wenn  $\mathcal{X}$  in der Konfiguration  $\mathcal{R}$  enthalten ist, auch  $\mathcal{Y}$  in  $\mathcal{R}$  oder  $(\mathcal{X}, \mathcal{Y})$  in der Menge der zu überprüfenden Assoziationen  $\mathbf{A}_u(\gamma, (\mathcal{A}, \mathcal{B}))$  eines

Überprüfungsauftrages  $(\gamma, (\mathcal{A}, \mathcal{B}))$  von  $U_r$  enthalten sind:

$$:= \left\{ (\mathcal{R}, U_r) \in \wp(\mathcal{I}) \times \wp(u(\mathcal{I})) \mid \mathcal{X} \subseteq \mathcal{R} \rightarrow \left( \mathcal{Y} \subseteq \mathcal{R} \vee \left\{ \begin{array}{l} \exists(\gamma, (\mathcal{A}, \mathcal{B})) \in U_r : \\ (\mathcal{X}, \mathcal{Y}) \in \mathbf{A}_u(\gamma, (\mathcal{A}, \mathcal{B})) \end{array} \right\} \right) \right\} \quad (\text{B-98})$$

Wenn  $\mathbf{A}_K(\mathcal{I}, \mathbf{P}, i)$  eindeutig ist, wird statt  $\mathbb{A}_{\mathbf{A}_K}$  verkürzt  $\mathbb{A}$  geschrieben.  $\square$

**Lemma B.55** Für alle Konfigurationen  $\mathcal{X}$ , in denen die Instanz  $\gamma$  enthalten ist, gilt, dass die Menge  $\mathbf{A}_X(\mathcal{I}, \mathbf{P}, i)$  der Instanz-Assoziationen mit  $\mathcal{X}$  Teil (der Vereinigung der Mengen) der zu überprüfenden Assoziationen  $\mathbf{A}_u(\gamma, (\mathcal{A}, \mathcal{B}))$  der mit  $\gamma$  zu generierenden Überprüfungsaufträge  $(\gamma, (\mathcal{A}, \mathcal{B})) \in u(\gamma)$  ist:

$$\forall \mathcal{X} \subseteq \mathcal{I}_i(\mathcal{I}, \mathbf{P}, i) : \forall \gamma \in \mathcal{X} : \quad (\text{B-99})$$

$$\mathbf{A}_X(\mathcal{I}, \mathbf{P}, i) \subseteq \bigcup_{(\gamma, (\mathcal{A}, \mathcal{B})) \in u(\gamma)} \mathbf{A}_u(\gamma, (\mathcal{A}, \mathcal{B}))$$

**Beweis** Mit Gl. B-83, Gl. B-85 und Gl. B-46 sowie Gl. B-48 gilt

$$\forall \mathcal{X} \subseteq \mathcal{I}_i(\mathcal{I}, \mathbf{P}, i) : \forall \gamma \in \mathcal{X} :$$

$$\mathbf{A}_X(\mathcal{I}, \mathbf{P}, i) \setminus \bigcup_{(\gamma, (\mathcal{C}, \mathcal{D})) \in \{\{\gamma\} \cap \mathcal{I}_i(\mathcal{I}, \mathbf{P}, i)\} \times \mathbf{P} = u(\gamma)} \mathbf{A}_u(\gamma, (\mathcal{C}, \mathcal{D})) =$$

$$= \left\{ (\mathcal{X}, \mathcal{Y}) \mid \begin{array}{l} \exists(\mathcal{A}, \mathcal{B}) \in \mathbf{P} : \\ i(\mathcal{B}|\mathcal{A} \mapsto \mathcal{X}) \neq \emptyset \\ \mathcal{Y} = i(\mathcal{B}|\mathcal{A} \mapsto \mathcal{X}) \end{array} \right\} \setminus \left\{ (\mathcal{U}, \mathcal{V}) \mid \begin{array}{l} \mathcal{U} \subseteq \mathcal{I}_i(\mathcal{I}, \mathbf{P}, i) \\ \gamma \in \mathcal{U} \\ \exists(\mathcal{C}, \mathcal{D}) \in \mathbf{P} : \\ i(\mathcal{D}|\mathcal{C} \mapsto \mathcal{U}) \neq \emptyset \\ \mathcal{V} = i(\mathcal{D}|\mathcal{C} \mapsto \mathcal{U}) \end{array} \right\} \quad (\text{B-100})$$

Die Menge, deren Elemente aus  $\mathbf{A}_X(\mathcal{I}, \mathbf{P}, i)$  entfernt wird, wird nun reduziert, indem statt aller möglicher  $\mathcal{U} \subseteq \mathcal{I}_i(\mathcal{I}, \mathbf{P}, i)$  nur noch  $\mathcal{X}$  betrachtet wird:

$$\forall \mathcal{X} \subseteq \mathcal{I}_i(\mathcal{I}, \mathbf{P}, i) : \forall \gamma \in \mathcal{X} :$$

$$\mathbf{A}_X(\mathcal{I}, \mathbf{P}, i) \setminus \bigcup_{(\gamma, (\mathcal{C}, \mathcal{D})) \in u(\gamma)} \mathbf{A}_u(\gamma, (\mathcal{C}, \mathcal{D})) \subseteq$$

$$\subseteq \left\{ (\mathcal{X}, \mathcal{Y}) \mid \begin{array}{l} \exists(\mathcal{A}, \mathcal{B}) \in \mathbf{P} : \\ i(\mathcal{B}|\mathcal{A} \mapsto \mathcal{X}) \neq \emptyset \\ \mathcal{Y} = i(\mathcal{B}|\mathcal{A} \mapsto \mathcal{X}) \end{array} \right\} \setminus \left\{ (\mathcal{X}, \mathcal{V}) \mid \begin{array}{l} \exists(\mathcal{C}, \mathcal{D}) \in \mathbf{P} : \\ i(\mathcal{D}|\mathcal{C} \mapsto \mathcal{X}) \neq \emptyset \\ \mathcal{V} = i(\mathcal{D}|\mathcal{C} \mapsto \mathcal{X}) \\ \gamma \in \mathcal{X} \end{array} \right\} \quad (\text{B-101})$$

Zusammengefasst heißt dies:

$$\forall \mathcal{X} \subseteq \mathcal{I}_i(\mathcal{I}, \mathbf{P}, i) : \forall \gamma \in \mathcal{X} :$$

$$\mathbf{A}_X(\mathcal{I}, \mathbf{P}, i) \setminus \bigcup_{(\gamma, (\mathcal{C}, \mathcal{D})) \in u(\gamma)} \mathbf{A}_u(\gamma, (\mathcal{C}, \mathcal{D})) \subseteq$$

$$\subseteq \left\{ (\mathcal{X}, \mathcal{Y}) \mid \begin{array}{l} \exists(\mathcal{C}, \mathcal{D}) \in \mathbf{P} : \\ i(\mathcal{D}|\mathcal{C} \mapsto \mathcal{X}) \neq \emptyset \\ \mathcal{Y} = i(\mathcal{D}|\mathcal{C} \mapsto \mathcal{X}) \\ \gamma \notin \mathcal{X} \end{array} \right\} \quad (\text{B-102})$$

Da die Mengendifferenz nur für  $\gamma \in \mathcal{X}$  gebildet wird, ist die untere Menge von Gl. B-102 (mit  $\gamma \notin \mathcal{X}$ ) und damit auch die Differenz  $\mathbf{A}_{\mathcal{X}}(\mathcal{I}, \mathbf{P}, i) \setminus \bigcup_{(\gamma, (\mathcal{C}, \mathcal{D})) \in u(\gamma)} \mathbf{A}_u(\gamma, (\mathcal{C}, \mathcal{D}))$  leer, so dass Gleichung B-99 gilt.  $\square$

**Verwendung** Lemma B.56

**Lemma B.56** Es gilt:

$$\begin{aligned} \forall (\mathcal{X}, \mathcal{Y}) \in \mathbf{A}_K(\mathcal{I}, \mathbf{P}, i) : \\ \forall \gamma \in \mathcal{X} : \\ \exists (\gamma, (\mathcal{A}, \mathcal{B})) \in u(\gamma) : \\ (\mathcal{X}, \mathcal{Y}) \in \mathbf{A}_u(\gamma, (\mathcal{A}, \mathcal{B})) \end{aligned} \quad (\text{B} - 103)$$

**Beweis** Aus Gleichung B-99

$$\begin{aligned} \forall \mathcal{X} \subseteq \mathcal{I}_i(\mathcal{I}, \mathbf{P}, i) : \\ \forall \gamma \in \mathcal{X} : \\ \mathbf{A}_{\mathcal{X}}(\mathcal{I}, \mathbf{P}, i) \subseteq \bigcup_{(\gamma, (\mathcal{A}, \mathcal{B})) \in u(\gamma)} \mathbf{A}_u(\gamma, (\mathcal{A}, \mathcal{B})) \end{aligned}$$

folgt mit der Definition von  $\mathbf{A}_{\mathcal{X}}(\mathcal{I}, \mathbf{P}, i)$  (Gl. B-48):

$$\begin{aligned} \forall \mathcal{X}, \mathcal{Y} \subseteq \mathcal{I} : \\ \forall \gamma \in \mathcal{X} : \\ (\mathcal{X}, \mathcal{Y}) \in \mathbf{A}_K(\mathcal{I}, \mathbf{P}, i) \rightarrow (\mathcal{X}, \mathcal{Y}) \in \bigcup_{(\gamma, (\mathcal{A}, \mathcal{B})) \in u(\gamma)} \mathbf{A}_u(\gamma, (\mathcal{A}, \mathcal{B})) \end{aligned}$$

Damit gelten auch

$$\begin{aligned} \forall (\mathcal{X}, \mathcal{Y}) \in \mathbf{A}_K(\mathcal{I}, \mathbf{P}, i) : \\ \forall \gamma \in \mathcal{X} : \\ (\mathcal{X}, \mathcal{Y}) \in \bigcup_{(\gamma, (\mathcal{A}, \mathcal{B})) \in u(\gamma)} \mathbf{A}_u(\gamma, (\mathcal{A}, \mathcal{B})) \end{aligned} \quad (\text{B} - 104)$$

sowie Gleichung B-103.  $\square$

**Verwendung** Lemma B.57, Lemma B.58

**Lemma B.57** Es gilt:

$$\begin{aligned} \forall \mathcal{W} \subseteq \mathcal{I} : \\ \forall (\mathcal{X}, \mathcal{Y}) \in \mathbf{A}_K(\mathcal{I}, \mathbf{P}, i) : \\ \mathcal{X} \cap \mathcal{W} \neq \emptyset \rightarrow \begin{cases} \exists (\gamma, (\mathcal{A}, \mathcal{B})) \in u(\mathcal{W}) : \\ (\mathcal{X}, \mathcal{Y}) \in \mathbf{A}_u(\gamma, (\mathcal{A}, \mathcal{B})) \end{cases} \end{aligned} \quad (\text{B} - 105)$$

**Beweis** Aus Lemma B.56 folgt:

$$\begin{aligned} \forall \mathcal{W} \subseteq \mathcal{I} : \\ \forall (\mathcal{X}, \mathcal{Y}) \in \mathbf{A}_K(\mathcal{I}, \mathbf{P}, i) : \\ \forall \gamma \in \mathcal{W} \cap \mathcal{X} : \\ \mathcal{W} \cap \mathcal{X} \neq \emptyset \rightarrow \begin{cases} \exists (\gamma, (\mathcal{A}, \mathcal{B})) \in u(\gamma) : \\ (\mathcal{X}, \mathcal{Y}) \in \mathbf{A}_u(\gamma, (\mathcal{A}, \mathcal{B})) \end{cases} \end{aligned} \quad (\text{B} - 106)$$

Außerdem ist  $u(\gamma) \subseteq u(\mathcal{W})$  (Lemma B.46). Damit gilt Gleichung B-105.  $\square$

**Verwendung** Satz B.65

**Lemma B.58** Geht der Zustand  $(\mathcal{R}, U_r)$  des Z-Automaten  $A((\mathcal{I}, \mathbf{P}, i), a)$  direkt in den Zustand  $(\mathcal{S}, U_s)$  über und liegt die Konfiguration  $\mathcal{X}$  einer Assoziation  $(\mathcal{X}, \mathcal{Y})$  aus der Menge der Instanz-Assoziationen  $\mathbf{A}_K(\mathcal{I}, \mathbf{P}, i)$  in  $\mathcal{S}$  vor ( $\mathcal{X} \subseteq \mathcal{S}$ ), aber nicht in  $\mathcal{R}$  ( $\mathcal{X} \not\subseteq \mathcal{R}$ ), dann gibt es einen Überprüfungsauftrag  $(\gamma, (\mathcal{A}, \mathcal{B})) \in U_s$  mit  $(\mathcal{X}, \mathcal{Y}) \in \mathbf{A}_u(\gamma, (\mathcal{A}, \mathcal{B}))$ :

$$\left. \begin{array}{l} \forall \mathcal{R}, \mathcal{S} \subseteq \mathcal{I} : \\ \forall U_r, U_s \subseteq u(\mathcal{I}) : \\ \forall (\mathcal{X}, \mathcal{Y}) \in \mathbf{A}_K(\mathcal{I}, \mathbf{P}, i) : \\ (\mathcal{R}, U_r) \stackrel{A}{\models} (\mathcal{S}, U_s) \\ \left. \begin{array}{l} \mathcal{X} \not\subseteq \mathcal{R} \\ \mathcal{X} \subseteq \mathcal{S} \end{array} \right\} \end{array} \right\} \rightarrow \left\{ \begin{array}{l} \exists (\gamma, (\mathcal{A}, \mathcal{B})) \in U_s : \\ (\mathcal{X}, \mathcal{Y}) \in \mathbf{A}_u(\gamma, (\mathcal{A}, \mathcal{B})) \end{array} \right. \quad (\text{B} - 107)$$

**Beweis** Da  $\mathcal{X}$  erst nach dem direkten Zustandsübergang vorliegt ( $\mathcal{X} \not\subseteq \mathcal{R} \wedge \mathcal{X} \subseteq \mathcal{S}$ ), werden mit dem Übergang  $(\mathcal{R}, U_r) \stackrel{A}{\models} (\mathcal{S}, U_s)$  Instanzen  $\gamma$  von  $\mathcal{X}$  generiert. Mit diesen Instanzen  $\gamma \in (\mathcal{S} \setminus \mathcal{R}) \cap \mathcal{X}$  werden beim Übergang Überprüfungsaufträge  $u(\gamma)$  generiert mit  $u(\gamma) \subseteq U_s$  (Gl. B-86 bis Gl. B-88). Mit Lemma B.56 gilt Gleichung B-107.  $\square$

**Verwendung** Satz B.59

**Satz B.59** Wenn der Zustand  $(\mathcal{R}, U_r)$  des Z-Automaten  $A((\mathcal{I}, \mathbf{P}, i), a)$  analysierbar ist, dann ist jeder Folgezustand  $(\mathcal{S}, U_s)$  analysierbar:

$$\left. \begin{array}{l} \forall \mathcal{R}, \mathcal{S} \subseteq \mathcal{I} : \\ \forall U_r, U_s \subseteq u(\mathcal{I}) : \\ \mathcal{R} \bowtie U_r \\ (\mathcal{R}, U_r) \stackrel{*A}{\models} (\mathcal{S}, U_s) \end{array} \right\} \rightarrow \mathcal{S} \bowtie U_s \quad (\text{B} - 108)$$

**Beweis** 1. Teil: Es ist zu zeigen, dass für ein beliebiges  $i \in \mathbb{N}$  aus

$$\left. \begin{array}{l} \mathcal{R} \bowtie U_r \\ (\mathcal{R}, U_r) \stackrel{(i)A}{\models} (\mathcal{V}_i, U_i) \end{array} \right\} \rightarrow \mathcal{V}_i \bowtie U_i \quad (\text{B} - 109)$$

und  $(\mathcal{V}_i, U_i) \stackrel{A}{\models} (\mathcal{V}_{i+1}, U_{i+1})$  folgt:

$$\left. \begin{array}{l} \mathcal{R} \bowtie U_r \\ (\mathcal{R}, U_r) \stackrel{(i+1)A}{\models} (\mathcal{V}_{i+1}, U_{i+1}) \end{array} \right\} \rightarrow \mathcal{V}_{i+1} \bowtie U_{i+1} \quad (\text{B} - 110)$$

Mit dem direkten Übergang  $(\mathcal{V}_i, U_i) \stackrel{A}{\models} (\mathcal{V}_{i+1}, U_{i+1})$  gilt  $(\mathcal{R}, U_r) \stackrel{(i)A}{\models} (\mathcal{V}_i, U_i) \stackrel{A}{\models} (\mathcal{V}_{i+1}, U_{i+1})$  bzw.  $(\mathcal{R}, U_r) \stackrel{(i+1)A}{\models} (\mathcal{V}_{i+1}, U_{i+1})$  (Gl. B-89). Bei diesem Übergang sind vier Fälle für Assoziationen  $(\mathcal{X}, \mathcal{Y}) \in \mathbf{A}_K(\mathcal{I}, \mathbf{P}, i)$  zu betrachten, mit denen aus der nach Voraussetzung (wegen  $\mathcal{V}_i \bowtie U_i$ , Gl. B-98) wahren Implikation  $\mathcal{X} \subseteq \mathcal{V}_i \rightarrow \mathcal{Y} \subseteq \mathcal{V}_i \vee \exists (\gamma, (\mathcal{A}, \mathcal{B})) \in U_i : (\mathcal{X}, \mathcal{Y}) \in \mathbf{A}_u(\gamma, (\mathcal{A}, \mathcal{B}))$  abgeleitet wird, dass die Implikation  $\mathcal{X} \subseteq \mathcal{V}_{i+1} \rightarrow \mathcal{Y} \subseteq \mathcal{V}_{i+1} \vee \exists (\gamma, (\mathcal{A}, \mathcal{B})) \in U_{i+1} : (\mathcal{X}, \mathcal{Y}) \in \mathbf{A}_u(\gamma, (\mathcal{A}, \mathcal{B}))$  ebenfalls wahr ist, so dass  $\mathcal{V}_{i+1} \bowtie U_{i+1}$  gilt.

1.  $\mathcal{X} \not\subseteq \mathcal{V}_i$  und  $\mathcal{X} \subseteq \mathcal{V}_{i+1}$ : In diesem Fall liegt nach dem Übergang eine Konfiguration  $\mathcal{X}$  neu vor, und es gilt mit Lemma B.58  $\exists(\gamma, (A, B)) \in U_{i+1} : (\mathcal{X}, \mathcal{Y}) \in \mathbf{A}_u(\gamma, (A, B))$ , d. h.  $\mathcal{V}_{i+1} \bowtie U_{i+1}$  (Gl. B-98).
2.  $\mathcal{X} \subseteq \mathcal{V}_i$  und  $\mathcal{Y} \subseteq \mathcal{V}_i$ : In diesem Fall werden  $\mathcal{X}$  und das Ergebnis  $\mathcal{Y}$  beim Übergang übernommen und liegen auch in  $\mathcal{V}_{i+1}$  vor: Wegen  $\mathcal{V}_i \xrightarrow{*Z} \mathcal{V}_{i+1}$  (Lemma B.48) gelten  $\mathcal{V}_i \subseteq \mathcal{V}_{i+1}$  (Lemma B.3) sowie  $\mathcal{X}, \mathcal{Y} \subseteq \mathcal{V}_{i+1}$ , d. h.  $\mathcal{V}_{i+1} \bowtie U_{i+1}$ .
3.  $\mathcal{X} \subseteq \mathcal{V}_i$ ,  $\mathcal{Y} \not\subseteq \mathcal{V}_i$  und  $\exists(\gamma, (A, B)) \in U_i : (\mathcal{X}, \mathcal{Y}) \in \mathbf{A}_u(\gamma, (A, B)) \wedge (\gamma, (A, B)) \in U_{i+1}$ : In diesem Fall wird der Überprüfungsauftrag  $(\gamma, (A, B))$  beim Übergang (nach Voraussetzung) nicht ausgewählt und weitergereicht. Außerdem gilt wieder  $\mathcal{X} \subseteq \mathcal{V}_{i+1}$ ; d. h.  $\mathcal{V}_{i+1} \bowtie U_{i+1}$ .
4.  $\mathcal{X} \subseteq \mathcal{V}_i$ ,  $\mathcal{Y} \not\subseteq \mathcal{V}_i$  und  $\exists(\gamma, (A, B)) \in U_i : (\mathcal{X}, \mathcal{Y}) \in \mathbf{A}_u(\gamma, (A, B)) \wedge (\gamma, (A, B)) \notin U_{i+1}$ : In diesem Fall wird der Überprüfungsauftrag  $(\gamma, (A, B))$  beim Übergang (nach Voraussetzung) ausgewählt, und es werden alle zugehörigen Assoziationen überprüft. Bei erfolgreichen Überprüfungen (Gl. B-86 bis B-88) wird das Ergebnis übernommen: Aus  $\mathcal{X} \subseteq \mathcal{V}_i$  bzw.  $\mathcal{X} \subseteq \mathcal{V}_{i+1}$  folgt  $\mathcal{Y} \subseteq \mathcal{V}_{i+1}$  (Gl. B-86 bis B-88); d. h.  $\mathcal{V}_{i+1} \bowtie U_{i+1}$ .

Alle Fälle mit  $\mathcal{X} \not\subseteq \mathcal{V}_{i+1}$  können bei der die Analysierbarkeit definierenden Implikation  $\mathcal{X} \subseteq \mathcal{V}_{i+1} \rightarrow \mathcal{Y} \subseteq \mathcal{V}_{i+1} \vee \exists(\gamma, (A, B)) \in U_{i+1} : (\mathcal{X}, \mathcal{Y}) \in \mathbf{A}_u(\gamma, (A, B))$  vernachlässigt werden.

Der analysierbare Zustand  $(\mathcal{V}_i, U_i)$  geht also in den analysierbaren Zustand  $(\mathcal{V}_{i+1}, U_{i+1})$  über. Aus Gleichung B-109 folgt damit Gleichung B-110.

2. Teil: Für  $i = 0$  mit  $\mathcal{R} = \mathcal{V}_0$  und  $U_r = U_0$  wird aus Gleichung B-109 die wahre Implikation

$$\left. \begin{array}{l} \mathcal{R} \bowtie U_r \\ (\mathcal{R}, U_r) \xrightarrow{(0)A} (\mathcal{R}, U_r) \end{array} \right\} \rightarrow \mathcal{R} \bowtie U_r .$$

Mit dem 1. Teil gilt Gleichung B-109 für alle  $i \in \mathbb{N}_0$ . Mit Gleichung B-89 erhält man Gleichung B-108.  $\square$

**Verwendung** Lemma B.60, Satz B.68, Satz B.64

**Lemma B.60** Wenn der Z-Automat  $((\mathcal{I}, \mathbf{P}, i), a)$  von einem analysierbaren Zustand  $(\mathcal{R}, U_r)$  aus abschließend in den Zustand  $(\mathcal{S}, \emptyset)$  übergeht, gilt für alle Assoziationen  $(\mathcal{X}, \mathcal{Y}) \in \mathbf{A}_K(\mathcal{I}, \mathbf{P}, i)$ , deren Konfiguration  $\mathcal{X}$  in  $\mathcal{S}$  enthalten ist, dass auch die assoziierte Konfiguration  $\mathcal{Y}$  in  $\mathcal{S}$  enthalten ist:

$$\left. \begin{array}{l} \forall \mathcal{R}, \mathcal{S} \subseteq \mathcal{I} : \\ \forall U_r \subseteq u(\mathcal{I}) : \\ (\mathcal{R}, U_r) \xrightarrow{\dagger A} (\mathcal{S}, \emptyset) \\ \mathcal{R} \bowtie U_r \end{array} \right\} \rightarrow \left\{ \begin{array}{l} \forall (\mathcal{X}, \mathcal{Y}) \in \mathbf{A}_K(\mathcal{I}, \mathbf{P}, i) : \\ \mathcal{X} \subseteq \mathcal{S} \rightarrow \mathcal{Y} \subseteq \mathcal{S} \end{array} \right. \quad (\text{B} - 111)$$

**Beweis** Aus  $(\mathcal{R}, U_r) \stackrel{*A}{=} (S, \emptyset)$  und  $\mathcal{R} \bowtie U_r$  folgt die Analysierbarkeit von  $(S, \emptyset)$  (Satz B.59). Daraus folgt mit Gleichung B-98

$$\begin{aligned} & \forall S \subseteq \mathcal{I} : \\ & \forall (\mathcal{X}, \mathcal{Y}) \in \mathbf{A}_K(\mathcal{I}, \mathbf{P}, i) : , \\ & \mathcal{X} \subseteq S \rightarrow \mathcal{Y} \subseteq S \end{aligned} \quad (\text{B} - 112)$$

so dass aus der linken Seite von Gleichung B-111 die rechte Seite folgt.  $\square$

**Verwendung** Satz B.61

**Satz B.61** Wenn der Zustand  $(\mathcal{R}, U_r)$  eines Z-Automaten  $A((\mathcal{I}, \mathbf{P}, i), a)$  analysierbar ist gilt, dass genau dann, wenn der Z-Automat  $A((\mathcal{I}, \mathbf{P}, i), a)$  vom Zustand  $(\mathcal{R}, U_r)$  aus abschließend in einen Zustand  $(S, \emptyset)$  übergeht, der Konfiguration  $\mathcal{R}$  durch das Z-System  $\widehat{Z} := (\mathcal{I}, \mathbf{A}_K(\mathcal{I}, \mathbf{P}, i))$  abschließend die Konfiguration  $S$  zugeordnet wird ( $\mathcal{R} \stackrel{\dagger \widehat{Z}}{=} S$ ) und der Z-Automat terminiert:

$$\begin{aligned} & \forall \mathcal{R}, S \subseteq \mathcal{I} : \\ & \forall U_r \subseteq u(\mathcal{I}) : \\ & \mathcal{R} \bowtie U_r \rightarrow \left( (\mathcal{R}, U_r) \stackrel{\dagger A}{=} (S, \emptyset) \leftrightarrow \left\{ \begin{array}{l} \mathcal{R} \stackrel{\dagger \widehat{Z}}{=} S \\ ((\mathcal{I}, \mathbf{P}, i), a) \text{ terminiert} \end{array} \right. \right) \end{aligned} \quad (\text{B} - 113)$$

**Beweis** 1. Teil:  $\rightarrow$  Mit Lemma B.60 und  $\stackrel{\dagger A}{=} \subseteq \stackrel{*A}{=}$  (Gl. B-90) gilt:

$$\begin{aligned} & \forall \mathcal{R}, S \subseteq \mathcal{I} : \\ & \forall U_r \subseteq u(\mathcal{I}) : \\ & \forall (\mathcal{X}, \mathcal{Y}) \in \mathbf{A}_K(\mathcal{I}, \mathbf{P}, i) : \\ & \left. \left. \begin{array}{l} (\mathcal{R}, U_r) \stackrel{\dagger A}{=} (S, \emptyset) \\ \mathcal{R} \bowtie U_r \end{array} \right\} \rightarrow \left\{ \begin{array}{l} (\mathcal{R}, U_r) \stackrel{*A}{=} (S, \emptyset) \\ \mathcal{X} \subseteq S \rightarrow \mathcal{Y} \subseteq S \end{array} \right. \end{aligned}$$

Mit Lemma B.7 gilt:

$$\begin{aligned} & \forall \mathcal{R}, S \subseteq \mathcal{I} : \\ & \forall (\mathcal{X}, \mathcal{Y}) \in \mathbf{A}_K(\mathcal{I}, \mathbf{P}, i) : \\ & \left. \begin{array}{l} \mathcal{R} \stackrel{* \widehat{Z}}{=} S \\ \mathcal{X} \subseteq S \rightarrow \mathcal{Y} \subseteq S \end{array} \right\} \leftrightarrow \mathcal{R} \stackrel{\dagger \widehat{Z}}{=} S \end{aligned}$$

Mit Satz B.49 gilt dann:

$$\begin{aligned} & \forall \mathcal{R}, S \subseteq \mathcal{I} : \\ & \forall U_r \subseteq u(\mathcal{I}) : \\ & \forall (\mathcal{X}, \mathcal{Y}) \in \mathbf{A}_K(\mathcal{I}, \mathbf{P}, i) : \\ & \left. \left. \begin{array}{l} (\mathcal{R}, U_r) \stackrel{\dagger A}{=} (S, \emptyset) \\ \mathcal{R} \bowtie U_r \end{array} \right\} \rightarrow \left\{ \begin{array}{l} (\mathcal{R}, U_r) \stackrel{*A}{=} (S, \emptyset) \\ \mathcal{X} \subseteq S \rightarrow \mathcal{Y} \subseteq S \end{array} \right\} \rightarrow \\ & \rightarrow \left\{ \begin{array}{l} \mathcal{R} \stackrel{* \widehat{Z}}{=} S \\ \mathcal{X} \subseteq S \rightarrow \mathcal{Y} \subseteq S \end{array} \right\} \rightarrow \mathcal{R} \stackrel{\dagger \widehat{Z}}{=} S \end{aligned}$$

Zusammengefasst erhält man:

$$\left. \begin{array}{l} \forall \mathcal{R}, \mathcal{S} \subseteq \mathcal{I} : \\ \forall U_r \subseteq u(\mathcal{I}) : \\ (\mathcal{R}, U_r) \stackrel{\dagger A}{=} (\mathcal{S}, \emptyset) \\ \mathcal{R} \bowtie U_r \end{array} \right\} \rightarrow \mathcal{R} \stackrel{\dagger \hat{Z}}{=} \mathcal{S}$$

Außerdem terminieren Z-Automaten, die von jedem analysierbaren Zustand aus in einen Zustand mit leerer Menge der Überprüfungsaufträge übergehen (Lemma B.51).

2. Teil:  $\leftarrow$  Jeder Zustand  $(\mathcal{R}, U_r)$  eines terminierenden Z-Automaten geht mit Lemma B.51 abschließend in einen Zustand  $(\mathcal{U}, \emptyset)$  über, und jeder Konfiguration  $\mathcal{R}$  wird mit Lemma B.6 abschließend eine Konfiguration  $\mathcal{V}$  zugeordnet:  $\mathcal{R} \stackrel{\dagger \hat{Z}}{=} \mathcal{V}$ . Es bleibt zu zeigen, dass  $\mathcal{U} = \mathcal{V}$  ist.

Sei  $\mathcal{U} \neq \mathcal{V}$ , dann wird  $\mathcal{R}$  mit dem 1. Teil des Beweises wegen  $(\mathcal{R}, U_r) \stackrel{\dagger A}{=} (\mathcal{U}, \emptyset)$  abschließend  $\mathcal{U}$  zugeordnet ( $\mathcal{R} \stackrel{\dagger \hat{Z}}{=} \mathcal{U}$ ), so dass  $\mathcal{R}$  zwei unterschiedliche Konfigurationen  $\mathcal{U}$  und  $\mathcal{V}$  abschließend zugeordnet werden. Dies ist ein Widerspruch zu Satz B.9, so dass  $\mathcal{U} = \mathcal{V}$  ist.  $\square$

**Verwendung** Korollar B.62, Korollar B.63, Satz B.68, Satz B.71

**Korollar B.62** Jeder analysierbare Zustand  $(\mathcal{R}, U_r)$  eines terminierende Z-Automaten  $A((\mathcal{I}, \mathbf{P}, i), a)$  geht abschließend in genau einen Zustand  $(\mathcal{S}, \emptyset)$  über:

$$\left. \begin{array}{l} \forall \mathcal{R} \subseteq \mathcal{I} : \\ \forall U_r \subseteq u(\mathcal{I}) : \\ \mathcal{R} \bowtie U_r \\ ((\mathcal{I}, \mathbf{P}, i), a) \text{ terminiert} \end{array} \right\} \rightarrow \left\{ \begin{array}{l} \exists! \mathcal{S} \subseteq \mathcal{I} : \\ (\mathcal{R}, U_r) \stackrel{\dagger A}{=} (\mathcal{S}, \emptyset) \end{array} \right. \quad (\text{B} - 114)$$

**Beweis** Mit Lemma B.51 geht jeder Zustand eines terminierenden Z-Automaten abschließend in einen Zustand ohne Überprüfungsaufträge über. Mit Satz B.61 wird der Konfiguration  $\mathcal{R}$  genau dann die Konfiguration  $\mathcal{S}$  abschließend zugeordnet, wenn der analysierbare Zustand  $(\mathcal{R}, U_r)$  abschließend in den Zustand  $(\mathcal{S}, \emptyset)$  übergeht. Da  $\mathcal{R}$  nur genau eine Konfiguration  $\mathcal{S}$  abschließend zugeordnet wird (Satz B.9), geht auch der Zustand  $(\mathcal{R}, U_r)$  nur in genau einen Zustand  $(\mathcal{S}, \emptyset)$  über.  $\square$

**Verwendung** Korollar B.63

**Korollar B.63** Ein terminierender Z-Automat  $((\mathcal{I}, \mathbf{P}, i), a)$  berechnet für jeden analysierbaren Zustand  $(\mathcal{R}, U_r)$  die Analysefunktion  $AF(\mathcal{R})$ :

$$\left. \begin{array}{l} \forall \mathcal{R} \subseteq \mathcal{I} : \\ \forall U_r \subseteq u(\mathcal{I}) : \\ \mathcal{R} \bowtie U_r \\ ((\mathcal{I}, \mathbf{P}, i), a) \text{ terminiert} \end{array} \right\} \rightarrow (\mathcal{R}, U_r) \stackrel{\dagger A}{=} (AF(\mathcal{R}), \emptyset) \quad (\text{B} - 115)$$

**Beweis**  $\stackrel{\dagger A}{\equiv}$  ist ausgehend von analysierbaren Zuständen rechtseindeutig (Korollar B.62) und definiert daher eine Funktion. Der Wert der Funktion wird mit Satz B.61 durch  $\stackrel{\dagger \widehat{Z}}{\equiv}$  (mit  $\widehat{Z} = (\mathcal{I}, \mathbf{A}_K(\mathcal{I}, \mathbf{P}, i))$ ) definiert und ist  $AF(\mathcal{R})$  (Korollar B.10).  $\square$

**Verwendung** Korollar B.64, Satz B.71

**Korollar B.64** Ausgehend von analysierbaren Zuständen konvergieren terminierende Z-Automaten: Sei  $(\mathcal{R}, U_r)$  ein analysierbarer Zustand, der in den Zustand  $(\mathcal{T}, U_t)$  übergeht, dann geht  $(\mathcal{T}, U_t)$  abschließend in den Zustand  $(AF(\mathcal{R}), \emptyset)$  über:

$$\left. \begin{array}{l} \forall \mathcal{R}, \mathcal{T} \subseteq \mathcal{I} : \\ \forall U_r, U_t \subseteq \wp(u(\mathcal{I})) : \\ \left. \begin{array}{l} \mathcal{R} \bowtie U_r \\ ((\mathcal{I}, \mathbf{P}, i), a) \text{ terminiert} \\ (\mathcal{R}, U_r) \stackrel{*A}{\equiv} (\mathcal{T}, U_t) \end{array} \right\} \rightarrow (\mathcal{T}, U_t) \stackrel{\dagger A}{\equiv} (AF(\mathcal{R}), \emptyset) \end{array} \right\} \quad (\text{B} - 116)$$

**Beweis** Der Zustand  $(\mathcal{T}, U_t)$  ist wegen  $\mathcal{R} \bowtie U_r$  analysierbar (Satz B.59) und geht abschließend in den Zustand  $(AF(\mathcal{T}), \emptyset)$  über (Korollar B.63).  $\mathcal{T}$  wird mit dem Z-System  $\widehat{Z} = (\mathcal{I}, \mathbf{A}_K(\mathcal{I}, \mathbf{P}, i))$  abschließend  $AF(\mathcal{T})$  zugeordnet ( $\mathcal{T} \stackrel{\dagger \widehat{Z}}{\equiv} AF(\mathcal{T})$ , Satz B.61), so dass das Z-System  $\widehat{Z}$  terminiert (Lemma B.6).

$\mathcal{R}$  wird die Konfiguration  $\mathcal{T}$  zugeordnet (Satz B.49). Da  $\widehat{Z}$  terminiert, wird  $\mathcal{T}$  die Konfiguration  $AF(\mathcal{T})$  und  $\mathcal{R}$  die Konfiguration  $AF(\mathcal{R})$  abschließend zugeordnet (Satz B.9, Korollar B.10). Mit Satz B.12 gilt  $AF(\mathcal{T}) = AF(\mathcal{R})$ .  $\square$

**Satz B.65** Wenn der Zustand  $(\mathcal{R}, U_r)$  analysierbar ist, dann ist auch der Zustand  $(\mathcal{R} \cup \mathcal{S}, U_r \cup u(\mathcal{S}))$  analysierbar:

$$\left. \begin{array}{l} \forall \mathcal{R}, \mathcal{S} \subseteq \mathcal{I} : \\ \forall U_r \subseteq u(\mathcal{I}) : \\ \mathcal{R} \bowtie U_r \rightarrow (\mathcal{R} \cup \mathcal{S}) \bowtie (U_r \cup u(\mathcal{S})) \end{array} \right\} \quad (\text{B} - 117)$$

**Beweis** Der Beweis betrachtet Assoziationen  $(\mathcal{X}, \mathcal{Y}) \in \mathbf{A}_K(\mathcal{I}, \mathbf{P}, i)$  und wird für die drei Fälle (i)  $\mathcal{X} \subseteq \mathcal{R}$ , (ii)  $\mathcal{X} \subseteq \mathcal{S}$  und (iii)  $\mathcal{X} \subseteq (\mathcal{R} \cup \mathcal{S}) \wedge \mathcal{X} \not\subseteq \mathcal{R} \wedge \mathcal{X} \not\subseteq \mathcal{S}$  durchgeführt, mit denen der Fall  $\mathcal{X} \subseteq (\mathcal{R} \cup \mathcal{S})$  abgedeckt wird.

Fall (i):  $(\mathcal{R}, U_r)$  ist analysierbar (Gl. B-98):

$$\left. \begin{array}{l} \forall \mathcal{R} \subseteq \mathcal{I} : \\ \forall U_r \subseteq u(\mathcal{I}) : \\ \forall (\mathcal{X}, \mathcal{Y}) \in \mathbf{A}_K(\mathcal{I}, \mathbf{P}, i) : \\ \left. \begin{array}{l} \mathcal{R} \bowtie U_r \\ \mathcal{X} \subseteq \mathcal{R} \end{array} \right\} \rightarrow \left( \mathcal{Y} \subseteq \mathcal{R} \vee \left\{ \begin{array}{l} \exists (\gamma, (\mathcal{A}, \mathcal{B})) \in U_r : \\ (\mathcal{X}, \mathcal{Y}) \in \mathbf{A}_u(\gamma, (\mathcal{A}, \mathcal{B})) \end{array} \right\} \right) \end{array} \right\} \quad (\text{B} - 118)$$

Für die Fälle (ii) und (iii) folgt aus Lemma B.57

$$\left. \begin{array}{l} \forall \mathcal{S} \subseteq \mathcal{I} : \\ \forall (\mathcal{X}, \mathcal{Y}) \in \mathbf{A}_K(\mathcal{I}, \mathbf{P}, i) : \\ \mathcal{X} \cap \mathcal{S} \neq \emptyset \rightarrow \left\{ \begin{array}{l} \exists (\gamma, (\mathcal{A}, \mathcal{B})) \in u(\mathcal{S}) : \\ (\mathcal{X}, \mathcal{Y}) \in \mathbf{A}_u(\gamma, (\mathcal{A}, \mathcal{B})) \end{array} \right\} . \end{array} \right\}$$



Damit liegen für die drei Fälle  $\mathcal{X} \subseteq \mathcal{R}$ ,  $\mathcal{X} \subseteq \mathcal{S}$  (hier ist  $\mathcal{X} \cap \mathcal{S} \neq \emptyset$ ) und  $\mathcal{X} \subseteq (\mathcal{R} \cup \mathcal{S}) \wedge \mathcal{X} \not\subseteq \mathcal{R} \wedge \mathcal{X} \not\subseteq \mathcal{S}$  (hier ist  $\mathcal{X} \cap \mathcal{S} \neq \emptyset$ ), d. h. für  $\mathcal{X} \subseteq (\mathcal{R} \cup \mathcal{S})$ , entweder Überprüfungsaufträge  $(\gamma, (\mathcal{A}, \mathcal{B})) \in u_r \cup u(\mathcal{S})$  oder Analyseergebnisse  $\mathcal{Y} \subseteq \mathcal{R}$  vor:

$$\begin{aligned} & \forall \mathcal{R}, \mathcal{S} \subseteq \mathcal{I} : \\ & \forall U_r \subseteq u(\mathcal{I}) : \\ & \forall (\mathcal{X}, \mathcal{Y}) \in \mathbf{A}_K(\mathcal{I}, \mathbf{P}, i) : \\ \mathcal{X} \subseteq (\mathcal{R} \cup \mathcal{S}) & \rightarrow \left( \mathcal{Y} \subseteq \mathcal{R} \subseteq (\mathcal{R} \cup \mathcal{S}) \vee \left\{ \begin{array}{l} \exists (\gamma, (\mathcal{A}, \mathcal{B})) \in (U_r \cup u(\mathcal{S})) : \\ (\mathcal{X}, \mathcal{Y}) \in \mathbf{A}_u(\gamma, (\mathcal{A}, \mathcal{B})) \end{array} \right. \right) \end{aligned} \quad (\text{B} - 119)$$

Der Vergleich mit Gleichung B-98 zeigt  $(\mathcal{R} \cup \mathcal{S}) \bowtie (U_r \cup u(\mathcal{S}))$ .  $\square$

**Verwendung** Korollar B.66, Satz B.68

**Korollar B.66** Der Zustand  $(\mathcal{R}, u(\mathcal{R}))$  ist analysierbar:

$$\forall \mathcal{R} \subseteq \mathcal{I} : \mathcal{R} \bowtie u(\mathcal{R}) \quad (\text{B} - 120)$$

**Beweis** Mit Satz B.65 folgt aus dem analysierbaren Zustand  $(\emptyset, \emptyset)$  (Gl. B-98), dass der Zustand  $(\emptyset \cup \mathcal{R}, \emptyset \cup u(\mathcal{R})) = (\mathcal{R}, u(\mathcal{R}))$  analysierbar ist.  $\square$

**Verwendung** Satz B.68

### B.2.3 Sequenzen von Teilanalysen und parallele Teilanalysen

**Definition B.67**  $\hat{u}(TV)$  ist die Menge der Überprüfungsaufträge zur *vereinten Überprüfung* der Konfigurationen von  $TV$ . Dabei müssen alle Überprüfungsaufträge generiert werden mit Assoziationen  $(\mathcal{X}, \mathcal{Y})$ , die Konfigurationen  $\mathcal{X}$  überprüfen, die in keiner der Mengen  $AF(\mathcal{O})$  - mit  $\mathcal{O} \in TV$  - vollständig vorliegen, sondern nur in der Zusammenfassung aller Ergebnisse  $\bigcup_{\mathcal{O} \in TV} AF(\mathcal{O})$ :

$$\begin{aligned} & \hat{u} : \wp(\wp(\mathcal{I})) \rightarrow \wp(\mathcal{I} \times \mathbf{P}) \\ & TV \mapsto \hat{u}(TV) \\ \hat{u}(TV) := & \left\{ (\gamma, (\mathcal{A}, \mathcal{B})) \in \mathcal{I} \times \mathbf{P} \mid \left. \begin{array}{l} \exists (\mathcal{X}, \mathcal{Y}) \in \mathbf{A}_K(\mathcal{I}, \mathbf{P}, i) : \\ \mathcal{X} \subseteq \left( \bigcup_{\mathcal{O} \in TV} AF(\mathcal{O}) \right) \\ \forall \mathcal{O} \in TV : \\ \mathcal{X} \not\subseteq AF(\mathcal{O}) \\ \gamma \in \mathcal{X} \end{array} \right\} \right. \end{aligned} \quad (\text{B} - 121)$$

Die Konfiguration  $\mathcal{I}_l$  ist die mit der Familie  $TV$  von Konfigurationen lokal überprüfbare Menge von Instanzen:

$$\begin{aligned} & \mathcal{I}_l : \wp(\wp(\mathcal{I})) \rightarrow \wp(\mathcal{I}) \\ & (TV) \mapsto \mathcal{I}_l(TV) \\ \mathcal{I}_l(TV) := & \{ \gamma \mid \forall (\mathcal{A}, \mathcal{B}) \in \mathbf{P} : (\gamma, (\mathcal{A}, \mathcal{B})) \notin \hat{u}(TV) \} \end{aligned} \quad (\text{B} - 122)$$

$\square$

**Satz B.68** Bei einer mit terminierenden Z-Automaten  $((\mathcal{I}, \mathbf{P}, i), a)$  durchgeführten Analyse kann die Analyse in nacheinander ablaufenden Teilanalysen erfolgen. Dabei wird dem Z-Automaten in jedem Analyseschritt eine Teilmenge aller zu analysierenden Instanzen  $\mathcal{R}$  übergeben:

$$\begin{aligned}
& \forall \mathcal{R} \subseteq \mathcal{I} : \\
& \forall TA = \{\mathcal{V}_0, \mathcal{V}_1, \dots, \mathcal{V}_{|TA|-1}\} \subseteq \wp(\mathcal{R}) : \\
& \quad \mathcal{Z}_0 := \emptyset \\
& \quad U_0 := \emptyset \\
& \left. \begin{aligned}
& (\mathcal{Z}_i \cup \mathcal{V}_i, U_i \cup u(\mathcal{V}_i)) \stackrel{*A}{=} (\mathcal{Z}_{i+1}, U_{i+1}), i = 0, 1, \dots, |TA| - 1 \\
& \left( \bigcup_{\mathcal{V} \in TA} \mathcal{V} \right) = \mathcal{R} \\
& ((\mathcal{I}, \mathbf{P}, i), a) \text{ terminiert}
\end{aligned} \right\} \rightarrow (\mathcal{Z}_{|TA|}, U_{|TA|}) \stackrel{\dagger A}{=} (AF(\mathcal{R}), \emptyset)
\end{aligned} \tag{B-123}$$

**Beweis** 1. Teil: Sei für ein beliebiges  $i \in \{0, 1, \dots, |TA| - 1\}$  der Zustand  $(\mathcal{Z}_i \cup \mathcal{V}_i, U_i \cup u(\mathcal{V}_i))$  analysierbar, dann ist der Folgezustand  $(\mathcal{Z}_{i+1}, U_{i+1})$  ebenfalls analysierbar (Satz B.59). Aus  $(\mathcal{Z}_{i+1}, U_{i+1})$  analysierbar folgt, dass  $(\mathcal{Z}_{i+1} \cup \mathcal{V}_{i+1}, U_{i+1} \cup u(\mathcal{V}_{i+1}))$  analysierbar ist (Satz B.65). Aus der Annahme  $(\mathcal{Z}_i \cup \mathcal{V}_i, U_i \cup u(\mathcal{V}_i))$  ist analysierbar folgt also, dass  $(\mathcal{Z}_{i+1} \cup \mathcal{V}_{i+1}, U_{i+1} \cup u(\mathcal{V}_{i+1}))$  analysierbar ist.

Für  $i = 0$  ist der Zustand  $(\mathcal{Z}_0 \cup \mathcal{V}_0, U_0 \cup u(\mathcal{V}_0)) = (\mathcal{V}_0, u(\mathcal{V}_0))$  analysierbar (Korollar B.66).

Damit ist  $(\mathcal{Z}_i \cup \mathcal{V}_i, U_i \cup u(\mathcal{V}_i))$  für alle  $i \in \{0, 1, \dots, |TA|\}$  analysierbar, d.h., dass  $(\mathcal{Z}_{|TA|}, U_{|TA|})$  analysierbar ist.

2. Teil: Zu jedem Zustandsübergang  $(\mathcal{Z}_i \cup \mathcal{V}_i, U_i \cup u(\mathcal{V}_i)) \stackrel{*A}{=} (\mathcal{Z}_{i+1}, U_{i+1}), i = 0, 1, \dots, |TA| - 1$  in Gleichung B-123 gibt es mit dem Z-System  $\widehat{\mathcal{Z}} = (\mathcal{I}, \mathbf{A}_K(\mathcal{I}, \mathbf{P}, i))$  die Zuordnung  $(\mathcal{Z}_i \cup \mathcal{V}_i) \stackrel{*{\widehat{\mathcal{Z}}}}{\mapsto} \mathcal{Z}_{i+1}$  (Lemma B.48).

Mit  $\bigcup_{\mathcal{V} \in TA} \mathcal{V} = \mathcal{R}$  liegt eine *kaskadierte Zuordnung*  $\mathcal{R} \stackrel{k{\widehat{\mathcal{Z}}}}{\mapsto} \mathcal{Z}_{|TA|}$  (Gl. B-23), die gleichzeitig eine Zuordnung  $\mathcal{R} \stackrel{*{\widehat{\mathcal{Z}}}}{\mapsto} \mathcal{Z}_{|TA|}$  ist (Satz B.26), vor.

3. Teil: Da der Z-Automat  $((\mathcal{I}, \mathbf{P}, i), a)$  terminiert, geht der analysierbare Zustand  $(\mathcal{Z}_{|TA|}, U_{|TA|})$  abschließend in den Zustand  $(AF(\mathcal{Z}_{|TA|}), \emptyset)$  über (Korollar B.63).  $\mathcal{Z}_{|TA|}$  wird abschließend die Konfiguration  $AF(\mathcal{Z}_{|TA|})$  zugeordnet (Satz B.61). Es gilt also  $\mathcal{R} \stackrel{*{\widehat{\mathcal{Z}}}}{\mapsto} \mathcal{Z}_{|TA|} \stackrel{\dagger{\widehat{\mathcal{Z}}}}{\mapsto} AF(\mathcal{Z}_{|TA|})$ . Mit Satz B.12 erhält man  $\mathcal{R} \stackrel{\dagger{\widehat{\mathcal{Z}}}}{\mapsto} AF(\mathcal{R})$  bzw.  $\mathcal{Z}_{|TA|} \stackrel{\dagger{\widehat{\mathcal{Z}}}}{\mapsto} AF(\mathcal{R})$ .

4. Teil: Da (i) der Zustand  $(\mathcal{Z}_{|TA|}, U_{|TA|})$  analysierbar ist (1. Teil des Beweises), (ii)  $\mathcal{Z}_{|TA|}$  abschließend  $AF(\mathcal{R})$  zugeordnet wird (3. Teil des Beweises) und (iii) der Z-Automat terminiert, erhält man  $(\mathcal{Z}_{|TA|}, U_{|TA|}) \stackrel{\dagger A}{=} (AF(\mathcal{R}), \emptyset)$  (Satz B.61).  $\square$

**Satz B.69** Direkte Zustandsübergänge können mit mehreren Prozessoren, die einen gemeinsamen Speicher verwenden, parallel durchgeführt werden.

**Beweis**

**Direkte parallele Zustandsübergänge** Bei direkten parallelen Zustandsübergängen  $\stackrel{\parallel^A}{=}$  (Gl. B-87) wird direkt vollständig parallel zugeordnet ( $\stackrel{\forall||Z}{=}$ , Gl. B-40) und damit auch zugeordnet (Lemma B.48). Dabei werden alle Assoziationen  $(\mathcal{X}_i, \mathcal{Y}_i)$  der Menge  $\{(\mathcal{X}_0, \mathcal{Y}_0), (\mathcal{X}_1, \mathcal{Y}_1), \dots\} = \mathbf{A}_g$  daraufhin überprüft, ob  $\mathcal{X}_i \subseteq \mathcal{R}$  gilt. Falls  $\mathcal{X}_i \subseteq \mathcal{R}$  zutrifft, wird eine (temporäre) Konfiguration  $Q_i = \mathcal{R} \cup \mathcal{Y}_i$  aufgebaut. Nach der Generierung aller Konfigurationen  $Q_i$  werden diese mit  $\mathcal{R}$  vereinigt. Diese Vereinigung ist die Konfiguration  $\mathcal{S}$  des Folgezustandes (Def. B.27). Die  $|\mathbf{A}_g|$  Überprüfungen  $\mathcal{X}_i \subseteq \mathcal{R}$  können also mit  $|\mathbf{A}_g|$  Prozessoren parallel durchgeführt werden.

**Direkte asynchrone Zustandsübergänge** Bei direkten asynchronen Zustandsübergängen  $\stackrel{aA}{=}$  (Gl. B-88) wird direkt vollständig asynchron zugeordnet ( $\stackrel{\forall a||Z}{=}$ , Gl. B-41) und damit auch zugeordnet (Lemma B.48). Dabei entsteht für jede Assoziation  $(\mathcal{X}_i, \mathcal{Y}_i)$  der Menge  $\{(\mathcal{X}_0, \mathcal{Y}_0), (\mathcal{X}_1, \mathcal{Y}_1), \dots\} = \mathbf{A}_g$  eine (temporäre) Konfiguration  $Q_i$ . Die erste Konfiguration ist  $\mathcal{R}$ , die letzte ist  $\mathcal{S}$  ( $\mathcal{R} = Q_0$  und  $\mathcal{S} = Q_{|\mathbf{A}_g|}$ ).  $\mathcal{X}_i$  liegt vor, wenn  $\exists j \leq i : \mathcal{X}_i \subseteq Q_j$  gilt. In diesem Fall wird  $Q_{i+1} = Q_i \cup \mathcal{X}_i$  berechnet (Def. B.27). Dabei gilt  $Q_j \subseteq Q_i$ . Überprüft wird  $Q_j$ , modifiziert wird  $Q_i$ , so dass die Überprüfung mit der Assoziation  $(\mathcal{X}_i, \mathcal{Y}_i)$  und die sich daraus eventuell ergebende Modifikation keine Rücksicht auf Überprüfungen und Modifikationen mit anderen Assoziationen zu nehmen brauchen. Assoziationen können von mehreren Prozessoren parallel und seriell in beliebiger Reihenfolge überprüft und zur Ergänzung von Konfigurationen verwendet werden. Die  $|\mathbf{A}_g|$  Überprüfungen können also mit 1 bis  $|\mathbf{A}_g|$  Prozessoren durchgeführt werden.  $\square$

**Lemma B.70** Wenn die Zustände  $(\mathcal{P}_1, U_{\mathcal{P}_1}), (\mathcal{P}_2, U_{\mathcal{P}_2}), \dots, (\mathcal{P}_l, U_{\mathcal{P}_l})$  analysierbar sind, dann ist - nachdem die Übergänge  $(\mathcal{P}_i, U_{\mathcal{P}_i}) \stackrel{*A}{=} (Q_i, U_{Q_i}), i = 1, 2, \dots, l$  erfolgten - auch der Zustand  $\left( \bigcup_{j=1}^l Q_j, \bigcup_{j=1}^l U_{Q_j} \cup \hat{u}(\{\mathcal{P}_1, \mathcal{P}_2, \dots, \mathcal{P}_l\}) \right)$  analysierbar.

**Beweis**

1. Teil: Es wird gezeigt, dass - nachdem die Übergänge  $(\mathcal{P}_i, U_{\mathcal{P}_i}) \stackrel{*A}{=} (Q_i, U_{Q_i})$  erfolgten - nur Überprüfungen  $\mathcal{X} \subseteq AF(\mathcal{R})$  (mit  $\mathcal{R} = \bigcup_{i=1}^l \mathcal{P}_i$  und  $(\mathcal{X}, \mathcal{Y}) \in \mathbf{A}$ ), die nicht gleichzeitig Überprüfungen  $\mathcal{X} \subseteq AF(\mathcal{P}_i)$  sind, nachgeholt werden müssen:

Für alle  $Q_i$  gilt  $(\mathcal{P}_i, U_{\mathcal{P}_i}) \stackrel{*A}{=} (Q_i, U_{Q_i}) \stackrel{\dagger A}{=} (AF(\mathcal{P}_i), \emptyset)$  (Satz B.64). Daraus folgt  $Q_i \stackrel{*Z}{=} AF(\mathcal{P}_i)$  (Lemma B.48) sowie  $Q_i \subseteq AF(\mathcal{P}_i)$  (Lemma B.3).

Aus  $\mathcal{X} \subseteq Q_i$  folgt also  $\mathcal{X} \subseteq AF(\mathcal{P}_i)$ . Alle Assoziationen  $(\mathcal{X}, \mathcal{Y}) \in \mathbf{A}_K(\mathcal{I}, \mathbf{P}, i)$  mit  $\mathcal{X} \subseteq AF(\mathcal{P}_i)$  können also bei allen Übergängen  $(\mathcal{P}_i, U_{\mathcal{P}_i}) \stackrel{*A}{=} (Q_i, U_{Q_i})$  überprüft werden, ohne dass die Gefahr besteht, dass die Überprüfung nur aufgrund der Aufteilung von  $\mathcal{R}$  in  $\mathcal{P}_1, \mathcal{P}_2, \dots, \mathcal{P}_l$  erfolglos ist.

Mit  $\mathcal{X} \not\subseteq AF(\mathcal{P}_i)$  und  $\mathcal{X} \subseteq AF(\mathcal{R})$  besteht dagegen bei den Übergängen  $(\mathcal{P}_i, U_{\mathcal{P}_i}) \stackrel{*A}{=} (\mathcal{Q}_i, U_{\mathcal{Q}_i})$  die Gefahr, dass eine Überprüfung  $\mathcal{X} \subseteq \mathcal{Q}_i$  nicht zutrifft, weil  $\mathcal{R}$  in  $\mathcal{P}_1, \mathcal{P}_2, \dots, \mathcal{P}_l$  aufgeteilt wurde.

2. Teil: Überprüfungsaufträge, die nur aufgrund der Aufteilung von  $\mathcal{R}$  in  $\mathcal{P}_1, \mathcal{P}_2, \dots, \mathcal{P}_l$  Hypothesen nicht bestätigen konnten, werden nach der Zusammenfassung der Konfigurationen  $\mathcal{Q}_1, \mathcal{Q}_2, \dots, \mathcal{Q}_l$  zu  $\mathcal{T}$  noch einmal in die Menge der mit  $\mathcal{T}$  zu überprüfenden Überprüfungsaufträge eingetragen. Eine Obermenge dieser Überprüfungsaufträge liefert  $\hat{u}(\mathcal{P}_1, \mathcal{P}_2, \dots, \mathcal{P}_l)$  (Gl. B-121), so dass der Zustand  $(\mathcal{T}, \hat{u}(\mathcal{P}_1, \mathcal{P}_2, \dots, \mathcal{P}_l) \cup \bigcup_{i=1}^l U_{\mathcal{Q}_i})$  analysierbar ist.  $\square$

**Verwendung Satz B.71**

**Satz B.71** Eine mit terminierenden Z-Automaten  $A = ((\mathcal{I}, \mathbf{P}, i), a)$  durchgeführte Analyse  $(\mathcal{R}, u(\mathcal{R})) \stackrel{\dagger A}{=} (AF(\mathcal{R}), \emptyset)$  kann mit mehreren Z-Automaten erfolgen:

$$\begin{aligned}
 & \forall \mathcal{R} \subseteq \mathcal{I} : \\
 & \forall k \in \mathbb{N} : \\
 & \forall TV = \{\mathcal{V}_1, \mathcal{V}_2, \dots, \mathcal{V}_k\} \subseteq \wp(\mathcal{R}) : \\
 & \quad \forall \{\mathcal{W}_1, \mathcal{W}_2, \dots, \mathcal{W}_k\} \subseteq \wp(\mathcal{I}) : \\
 & \quad \forall \{U_{\mathcal{V}_1}, U_{\mathcal{V}_2}, \dots, U_{\mathcal{V}_k}\} \subseteq \wp(u(\mathcal{I})) : \\
 & \quad \forall \{U_{\mathcal{W}_1}, U_{\mathcal{W}_2}, \dots, U_{\mathcal{W}_k}\} \subseteq \wp(u(\mathcal{I})) : \\
 & \left. \begin{array}{l}
 \left( \bigcup_{j=1}^k \mathcal{V}_j \right) = \mathcal{R} \\
 \forall j \in \{1, 2, \dots, k\} : \\
 (\mathcal{V}_j, U_{\mathcal{V}_j}) \stackrel{*A}{=} (\mathcal{W}_j, U_{\mathcal{W}_j}) \\
 \mathcal{V}_j \bowtie U_{\mathcal{V}_j} \\
 ((\mathcal{I}, \mathbf{P}, i), a) \text{ terminiert}
 \end{array} \right\} \rightarrow \left( \bigcup_{j=1}^k \mathcal{W}_j, \hat{u}(TV) \cup \bigcup_{j=1}^k U_{\mathcal{W}_j} \right) \stackrel{\dagger A}{=} (AF(\mathcal{R}), \emptyset)
 \end{aligned}
 \tag{B - 124}$$

**Beweis** Mit Lemma B.70 ist  $\left( \bigcup_{j=1}^k \mathcal{W}_j, \hat{u}(TV) \cup \bigcup_{j=1}^k U_{\mathcal{W}_j} \right)$  analysierbar. Aus  $(\mathcal{V}_j, U_{\mathcal{V}_j}) \stackrel{*A}{=} (\mathcal{W}_j, U_{\mathcal{W}_j})$  folgt  $\mathcal{V}_j \stackrel{*Z}{=} \mathcal{W}_j$  (Lemma B.48). Der Konfiguration  $\mathcal{R} = \bigcup_{j=1}^k \mathcal{V}_j$  wird mit dem Tupel  $W = \underbrace{(\mathbf{A}, \mathbf{A}, \dots, \mathbf{A})}_{k\text{-Mal}}$  unabhängig verteilt  $\mathcal{T} := \bigcup_{j=1}^k \mathcal{W}_j$  zugeordnet (Gl. B-21):  $\mathcal{R} \stackrel{v|Z}{=} W \mathcal{T}$ . Mit  $\stackrel{v|Z}{=} \subseteq \stackrel{*Z}{=}$  (Lemma B.21) gilt  $\mathcal{R} \stackrel{*Z}{=} \mathcal{T}$ .

Da der Automat  $((\mathcal{I}, \mathbf{P}, i), a)$  terminiert, geht der analysierbare Zustand  $(\mathcal{T}, U_t)$  mit der Abkürzung  $U_t := \hat{u}(TV) \cup \bigcup_{j=1}^k U_{\mathcal{W}_j}$  abschließend nach  $(AF(\mathcal{T}), \emptyset)$  über (Korollar B.63) und das Z-System  $\hat{Z} = (\mathcal{I}, \mathbf{A}_K(\mathcal{I}, \mathbf{P}, i))$  ordnet  $\mathcal{T}$  abschließend  $AF(\mathcal{T})$  zu:  $\mathcal{T} \stackrel{\dagger \hat{Z}}{=} AF(\mathcal{T})$  (Satz B.61).

Mit Satz B.12 erhält man  $\mathcal{R} \stackrel{\dagger \hat{Z}}{\vdash} AF(\mathcal{R})$  bzw.  $\mathcal{R} \stackrel{*Z}{\vdash} \mathcal{T} \stackrel{\dagger \hat{Z}}{\vdash} AF(\mathcal{R})$  und  $\mathcal{T} \stackrel{\dagger \hat{Z}}{\vdash} AF(\mathcal{R})$ . Da der Z-Automat terminiert und der Zustand  $(\mathcal{T}, U_t)$  analysierbar ist, geht der Zustand  $(\mathcal{T}, U_t)$  abschließend nach  $(AF(\mathcal{R}), \emptyset)$  über (Satz B.61):  

$$\left( \bigcup_{j=1}^k \mathcal{W}_j, \hat{u}(TV) \cup \bigcup_{j=1}^k U_{\mathcal{W}_j} \right) \stackrel{\dagger A}{\vdash} (AF(\mathcal{R}), \emptyset) \quad \square$$

### B.2.4 Zuordnungsautomaten mit Ableitungen

**Definition B.72** Ein *Zuordnungsautomat mit Ableitungen* (kurz: ZA-Automat) ist ein Paar  $A_A = ((\mathbf{I}, \mathbf{P}, i), a)$  mit

$$\begin{aligned} & (\mathbf{I}, \mathbf{P}, i) : \text{Z-System mit Ableitungen} \\ & \mathbf{I} = (M_1, M_2, \dots, M_m) \\ & M_i : i\text{-tes Merkmal, } i = 2, 3, \dots, m \\ & T : \text{Typ-Merkmal} \\ & \text{mit } M_1 = T \end{aligned} \quad (B - 125)$$

$\mathbf{P}$  : endliche Relation der Konzept-Assoziationen  
 $i$  : Instanziierungsfunktion  
 $a$  : Auswahlfunktion

Ein ZA-Automat  $A_A((\mathbf{I}, \mathbf{P}, i), a)$  heißt genau dann *wohldefiniert* (kurz: wZA-Automat) und wird dann geschrieben  $wA_A((\mathbf{I}, \mathbf{P}, i), a)$ , wenn folgende Aussagen gelten:

1.  $((\mathcal{I}_A, \mathbf{P}, i), a)$  ist mit  $\mathcal{I}_A = \mathcal{I}_M \times I_X \times P_M$  (Gl. B-54),  $I_X := \mathbb{N}$  und  $P_M = \wp(I_X)$  (Gl. B-53) sowie  $\mathcal{I}_M = M_1 \times M_2 \times \dots \times M_m$  (Gl. B-51) ein wohldefinierter Z-Automat.
2. Für die Instanziierungsfunktion  $i$  und das *Typ-Merkmal*  $\gamma^{(T)}$  der Ableitungen  $\gamma \in \mathcal{I}_A$  gilt:

$$\begin{aligned} & \forall \mathcal{X} \subseteq \mathcal{I}_A : \\ & \forall (\mathcal{A}, \mathcal{B}) \in \mathbf{P} : \\ & i(\mathcal{B}|\mathcal{A} \mapsto \mathcal{X}) \neq \emptyset \rightarrow \begin{cases} \forall \gamma \in \mathcal{X} : \\ \exists \alpha \in \mathcal{A} : \\ \alpha^{(T)} = \gamma^{(T)} \end{cases} \end{aligned} \quad (B - 126)$$

Im Folgenden werden nur noch wohldefinierte ZA-Automaten betrachtet, so dass nicht wZA-Automat, sondern verkürzt ZA-Automat und nicht  $wA_A((\mathbf{I}, \mathbf{P}, i), a)$ , sondern  $A_A((\mathbf{I}, \mathbf{P}, i), a)$  geschrieben wird.

Mit der Relation der Konzept-Assoziationen  $\mathbf{P}$  wird die aus einer Konfiguration  $\mathcal{R} \subseteq \mathcal{I}_A$  zu bildende Menge der *reduzierten Überprüfungsaufträge*  $u^r(\mathcal{R})$  des ZA-Automaten  $A_A((\mathbf{I}, \mathbf{P}, i), a)$  definiert:

$$\begin{aligned} & u^r : \wp(\mathcal{I}_A) \rightarrow \wp(\mathcal{I}_A \times \mathbf{P}) \\ & \mathcal{R} \mapsto u^r(\mathcal{R}) \end{aligned} \quad (B - 127)$$

$$u^r(\mathcal{R}) := \left\{ (\gamma, (\mathcal{A}, \mathcal{B})) \in [\mathcal{R} \cap \mathcal{I}_i(\mathcal{I}, \mathbf{P}, i)] \times \mathbf{P} \mid \begin{array}{l} \exists \alpha \in \mathcal{A} : \\ \gamma^{(T)} = \alpha^{(T)} \end{array} \right\}$$

**Satz B.73** Die Menge der mit reduzierten Überprüfungsaufträgen überprüfbaren Assoziationen des ZA-Automaten  $A_A((\mathbf{I}, \mathbf{P}, i), a)$  ist gleich der Menge der mit nicht reduzierten Überprüfungsaufträgen (Gl. B-83) überprüfbaren Assoziationen:

$$\forall \mathcal{R} \subseteq \mathcal{I}_A : \quad \bigcup_{(\gamma, (\mathcal{A}, \mathcal{B})) \in u(\mathcal{R})} \mathbf{A}_u(\gamma, (\mathcal{A}, \mathcal{B})) = \bigcup_{(\gamma, (\mathcal{A}, \mathcal{B})) \in u^r(\mathcal{R})} \mathbf{A}_u(\gamma, (\mathcal{A}, \mathcal{B})) \quad (\text{B} - 128)$$

**Beweis** Für alle  $\mathcal{R} \subseteq \mathcal{I}_A$  ist

$$\bigcup_{(\gamma, (\mathcal{A}, \mathcal{B})) \in u(\mathcal{R})} \mathbf{A}_u(\gamma, (\mathcal{A}, \mathcal{B})) = \left\{ (\mathcal{X}, \mathcal{Y}) \left| \begin{array}{l} \exists \gamma \in \mathcal{R} : \\ \mathcal{X} \subseteq \mathcal{I}_A \\ \gamma \in \mathcal{X} \\ \exists (\mathcal{A}, \mathcal{B}) \in \mathbf{P} : \\ i(\mathcal{B}|\mathcal{A} \mapsto \mathcal{X}) \neq \emptyset \\ \mathcal{Y} = i(\mathcal{B}|\mathcal{A} \mapsto \mathcal{X}) \end{array} \right. \right\} \quad (\text{B} - 129)$$

(Gl. B-83, B-85) Obermenge<sup>1</sup> von

$$\bigcup_{(\gamma, (\mathcal{A}, \mathcal{B})) \in u^r(\mathcal{R})} \mathbf{A}_u(\gamma, (\mathcal{A}, \mathcal{B})) = \left\{ (\mathcal{X}, \mathcal{Y}) \left| \begin{array}{l} \exists \gamma \in \mathcal{R} : \\ \mathcal{X} \subseteq \mathcal{I}_A \\ \gamma \in \mathcal{X} \\ \exists (\mathcal{A}, \mathcal{B}) \in \mathbf{P} : \\ i(\mathcal{B}|\mathcal{A} \mapsto \mathcal{X}) \neq \emptyset \\ \mathcal{Y} = i(\mathcal{B}|\mathcal{A} \mapsto \mathcal{X}) \\ \exists \alpha \in \mathcal{A} : \\ \alpha^{(T)} = \gamma^{(T)} \end{array} \right. \right\}. \quad (\text{B} - 130)$$

Die nur in der Definition der Teilmenge (Gl. B-130) zusätzlich auftretende Bedingung

$$\begin{array}{l} \exists \alpha \in \mathcal{A} : \\ \alpha^{(T)} = \gamma^{(T)} \end{array}$$

folgt mit Gl. B-126 aus  $i(\mathcal{B}|\mathcal{A} \mapsto \mathcal{X}) \neq \emptyset$  und kann ignoriert werden. In allen anderen Bedingungen stimmen die Definitionen der Mengen nach Gleichung B-129 und Gleichung B-130 überein, so dass Gleichung B-128 gilt.  $\square$

### B.3 Erkennung erzeugter Konfigurationen

Es folgen Beweise für Aussagen über Zusammenhänge zwischen generierten und analysierten Mengen von Konfigurationen, die im Abschnitt 2.4 aufgeführt sind.

---

<sup>1</sup>Alle in Gleichung B-129 aufgeführten Bedingungen für die Definition der Assoziationen der Menge sind auch in Gleichung B-130 aufgeführt.

**Lemma B.74** Zu jeder Ersetzung  $\mathcal{R} \xrightarrow[e]{*\mathbf{A}} \mathcal{S}$  mit einfach assoziierenden Assoziationsmengen  $\mathbf{A}$  gibt es eine Ersetzung  $\mathcal{S} \xrightarrow[e]{*\mathbf{A}^{-1}} \mathcal{R}$ :

$$\forall \mathbf{A} \subseteq \wp(\mathcal{I}) \times \wp(\mathcal{I}) : \quad \mathbf{A} \text{ ist einfach assoziierend} \rightarrow \begin{cases} \forall \mathcal{R}, \mathcal{S} \subseteq \mathcal{I} : \\ \mathcal{R} \xrightarrow[e]{*\mathbf{A}} \mathcal{S} \rightarrow \mathcal{S} \xrightarrow[e]{*\mathbf{A}^{-1}} \mathcal{R} \end{cases} \quad (\text{B} - 131)$$

**Beweis** 1. Teil: Es ist zu zeigen, dass für ein beliebiges  $i \in \mathbb{N}$  für einfach assoziierende Assoziationsmengen  $\mathbf{A}$  aus

$$\mathcal{R} \xrightarrow[e]{(i)\mathbf{A}} \mathcal{U}_i \rightarrow \mathcal{U}_i \xrightarrow[e]{(i)\mathbf{A}^{-1}} \mathcal{R} \quad (\text{B} - 132)$$

und  $\mathcal{U}_i \xrightarrow[e]{\mathbf{A}} \mathcal{U}_{i+1}$  folgt:

$$\mathcal{R} \xrightarrow[e]{(i+1)\mathbf{A}} \mathcal{U}_{i+1} \rightarrow \mathcal{U}_{i+1} \xrightarrow[e]{(i+1)\mathbf{A}^{-1}} \mathcal{R} \quad (\text{B} - 133)$$

Mit  $\mathcal{U}_i \xrightarrow[e]{\mathbf{A}} \mathcal{U}_{i+1}$  erhält man  $\mathcal{R} \xrightarrow[e]{(i)\mathbf{A}} \mathcal{U}_i \xrightarrow[e]{\mathbf{A}} \mathcal{U}_{i+1}$  bzw.  $\mathcal{R} \xrightarrow[e]{(i+1)\mathbf{A}} \mathcal{U}_{i+1}$ . Außerdem gilt (Gl. 2-7):

$$\begin{aligned} \exists (\mathcal{X}_i, \mathcal{Y}_i) \in \mathbf{A} : \\ \mathcal{X}_i \subseteq \mathcal{U}_i \\ \mathcal{U}_{i+1} = (\mathcal{U}_i \setminus \mathcal{X}_i) \cup \mathcal{Y}_i \end{aligned} \quad (\text{B} - 134)$$

Damit gilt auch  $\mathcal{Y}_i \subseteq \mathcal{U}_{i+1}$ . Mit  $(\mathcal{Y}_i, \mathcal{X}_i) \in \mathbf{A}^{-1}$  und

$$\mathcal{T} := (\mathcal{U}_{i+1} \setminus \mathcal{Y}_i) \cup \mathcal{X}_i \quad (\text{B} - 135)$$

gibt es also die Ersetzung

$$\mathcal{U}_{i+1} \xrightarrow[e]{\mathbf{A}^{-1}} \mathcal{T}. \quad (\text{B} - 136)$$

$\mathcal{U}_{i+1}$  wird jetzt in der Definition von  $\mathcal{T}$  (Gl. B-135) entsprechend Gleichung B-134 durch  $\mathcal{U}_i$  ausgedrückt:

$$\mathcal{T} = ((\mathcal{U}_i \setminus \mathcal{X}_i) \cup \mathcal{Y}_i) \setminus \mathcal{Y}_i \cup \mathcal{X}_i$$

Wegen des letzten Terms  $\cup \mathcal{X}_i$  wird  $\setminus \mathcal{X}_i$  gestrichen. Wegen des vorletzten Terms  $\setminus \mathcal{Y}_i$  wird  $\cup \mathcal{Y}_i$  gestrichen. Man erhält  $\mathcal{T} = (\mathcal{U}_i \setminus \mathcal{Y}_i) \cup \mathcal{X}_i$  bzw.  $\mathcal{T} = (\mathcal{U}_i \cup \mathcal{X}_i) \setminus (\mathcal{Y}_i \setminus \mathcal{X}_i)$ . Mit  $\mathcal{X}_i \subseteq \mathcal{U}_i$  (Gl. B-134) gilt  $\mathcal{T} = \mathcal{U}_i \setminus (\mathcal{Y}_i \setminus \mathcal{X}_i)$  bzw.

$$\mathcal{T} = \mathcal{U}_i \setminus [\mathcal{U}_i \cap (\mathcal{Y}_i \setminus \mathcal{X}_i)]. \quad (\text{B} - 137)$$

Da  $\mathbf{A}$  bei der Ersetzung  $\mathcal{U}_i \xrightarrow[e]{\mathbf{A}} \mathcal{U}_{i+1}$  einfach assoziierend ist, gilt  $\mathcal{U}_i \cap (\mathcal{Y}_i \setminus \mathcal{X}_i) = \emptyset$  (Gl. B-72), so dass  $\mathcal{U}_i = \mathcal{T}$  gilt. Aus  $\mathcal{U}_i \xrightarrow[e]{\mathbf{A}} \mathcal{U}_{i+1}$  folgt also  $\mathcal{U}_{i+1} \xrightarrow[e]{\mathbf{A}^{-1}} \mathcal{U}_i$

(Gl. B-136), so dass aus  $\mathcal{R} \xrightarrow[e]{(i)\mathbf{A}} \mathcal{U}_i \rightarrow \mathcal{U}_i \xrightarrow[e]{(i)\mathbf{A}^{-1}} \mathcal{R}$  folgt  $\mathcal{R} \xrightarrow[e]{(i+1)\mathbf{A}} \mathcal{U}_{i+1} \rightarrow \mathcal{U}_{i+1} \xrightarrow[e]{(i+1)\mathbf{A}^{-1}} \mathcal{R}$ . Aus Gleichung B-132 folgt also Gleichung B-133.

2. Teil: Mit  $\mathcal{U}_0 = \mathcal{R}$  gilt  $\mathcal{R} \xrightarrow[e]{(0)\mathbf{A}} \mathcal{U}_0 \xrightarrow[e]{(0)\mathbf{A}} \mathcal{R}$  (Gl. 2-1). Gleichung B-132 gilt also für  $i = 0$ . Mit dem 1. Teil des Beweises gilt Gleichung B-132 für alle  $i \in \mathbb{N}$ . Mit Gleichung 2-1 erhält man Gleichung B-131.  $\square$

**Lemma B.75** Sei  $\mathbf{A}^a$  die Menge der analysierenden Assoziationen und Umkehrrelation der einfach assoziierenden generierenden Menge der Assoziationen  $\mathbf{A}^g$  ( $\mathbf{A}^a := (\mathbf{A}^g)^{-1}$ ),  $\mathcal{I}_T \cup \mathcal{I}_N$  die Menge der Instanzen und  $\Sigma \subseteq \mathcal{I}_N$  die Menge der Startsymbole, dann ist die generierte Menge der Konfigurationen  $S_g(\mathcal{I}_T, \mathcal{I}_N, \Sigma, \mathbf{A}^g)$  in der vollständig ersetzend analysierten Menge der Konfigurationen  $S_a^{ve}(\mathcal{I}_T, \mathcal{I}_N, \Sigma, \mathbf{A}^a)$  enthalten:

$$\begin{aligned} & \forall \mathcal{I}_T, \mathcal{I}_N \subseteq \mathcal{I} : \\ & \forall \mathbf{A}^g \subseteq \wp(\mathcal{I}_T \cup \mathcal{I}_N) \times \wp(\mathcal{I}_T \cup \mathcal{I}_N) : \\ & \quad \forall \Sigma \subseteq \mathcal{I}_N : \\ & \quad \mathbf{A}^a := (\mathbf{A}^g)^{-1} \\ & \mathbf{A}^g \text{ ist einfach assoziierend} \rightarrow \\ & \rightarrow S_g(\mathcal{I}_T, \mathcal{I}_N, \Sigma, \mathbf{A}^g) \subseteq S_a^{ve}(\mathcal{I}_T, \mathcal{I}_N, \Sigma, \mathbf{A}^a) \end{aligned} \quad (\text{B} - 138)$$

**Beweis** Für jedes  $\sigma \in \Sigma$  und jede Ersetzung  $\{\sigma\} \xrightarrow[e]{+\mathbf{A}^g} \mathcal{R}$  gibt es eine Ersetzung  $\mathcal{R} \xrightarrow[e]{+\mathbf{A}^a} \{\sigma\}$  (Lemma B.74), so dass aus  $\mathcal{R} \in S_g(\mathcal{I}_T, \mathcal{I}_N, \Sigma, \mathbf{A}^g)$  (Gl. 2-15) folgt  $\mathcal{R} \in S_a^{ve}(\mathcal{I}_T, \mathcal{I}_N, \Sigma, (\mathbf{A}^g)^{-1})$  (Gl. 2-17), d. h.  $S_g(\mathcal{I}_T, \mathcal{I}_N, \Sigma, \mathbf{A}^g) \subseteq S_a^{ve}(\mathcal{I}_T, \mathcal{I}_N, \Sigma, (\mathbf{A}^g)^{-1})$ .  $\square$

**Lemma B.76** Sei  $\mathbf{A}^g$  die Menge der generierenden Assoziationen und Umkehrrelation der einfach assoziierenden analysierenden Menge der Assoziationen  $\mathbf{A}^a$  ( $\mathbf{A}^a := (\mathbf{A}^g)^{-1}$ ),  $\mathcal{I}_T \cup \mathcal{I}_N$  die Menge der Instanzen und  $\Sigma \subseteq \mathcal{I}_N$  die Menge der Startsymbole, dann ist die vollständig ersetzend analysierte Menge der Konfigurationen  $S_a^{ve}(\mathcal{I}_T, \mathcal{I}_N, \Sigma, \mathbf{A}^a)$  in der generierten Menge der Konfigurationen  $S_g(\mathcal{I}_T, \mathcal{I}_N, \Sigma, \mathbf{A}^g)$  enthalten:

$$\begin{aligned} & \forall \mathcal{I}_T, \mathcal{I}_N \subseteq \mathcal{I} : \\ & \forall \mathbf{A}^g \subseteq \wp(\mathcal{I}_T \cup \mathcal{I}_N) \times \wp(\mathcal{I}_T \cup \mathcal{I}_N) : \\ & \quad \forall \Sigma \subseteq \mathcal{I}_N : \\ & \quad \mathbf{A}^a := (\mathbf{A}^g)^{-1} \\ & \mathbf{A}^a \text{ ist einfach assoziierend} \rightarrow \\ & \rightarrow S_a^{ve}(\mathcal{I}_T, \mathcal{I}_N, \Sigma, \mathbf{A}^a) \subseteq S_g(\mathcal{I}_T, \mathcal{I}_N, \Sigma, \mathbf{A}^g) \end{aligned} \quad (\text{B} - 139)$$

**Beweis** Für jedes  $\sigma \in \Sigma$  und jede Ersetzung  $\mathcal{R} \xrightarrow[e]{+\mathbf{A}^a} \{\sigma\}$  gibt es eine Ersetzung  $\{\sigma\} \xrightarrow[e]{+\mathbf{A}^g} \mathcal{R}$  (Lemma B.74), so dass aus  $\mathcal{R} \in S_a^{ve}(\mathcal{I}_T, \mathcal{I}_N, \Sigma, (\mathbf{A}^g)^{-1})$  (Gl. 2-17) folgt  $\mathcal{R} \in S_g(\mathcal{I}_T, \mathcal{I}_N, \Sigma, \mathbf{A}^g)$  (Gl. 2-15), d. h.  $S_a^{ve}(\mathcal{I}_T, \mathcal{I}_N, \Sigma, (\mathbf{A}^g)^{-1}) \subseteq S_g(\mathcal{I}_T, \mathcal{I}_N, \Sigma, \mathbf{A}^g)$ .  $\square$



**Satz B.77** Sei  $\mathbf{A}^g$  die Menge der einfach assoziierend generierenden Assoziationen und Umkehrrelation der einfach assoziierend analysierenden Menge der Assoziationen  $\mathbf{A}^a$  ( $\mathbf{A}^a := (\mathbf{A}^g)^{-1}$ ),  $\mathcal{I}_T \cup \mathcal{I}_N$  die Menge der Instanzen und  $\Sigma \subseteq \mathcal{I}_N$  die Menge der Startsymbole, dann ist die vollständig ersetzend analysierte Menge der Konfigurationen  $S_a^{ve}(\mathcal{I}_T, \mathcal{I}_N, \Sigma, \mathbf{A}^a)$  gleich der generierten Menge der Konfigurationen  $S_g(\mathcal{I}_T, \mathcal{I}_N, \Sigma, \mathbf{A}^g)$ :

$$\begin{aligned} & \forall \mathcal{I}_T, \mathcal{I}_N \subseteq \mathcal{I} : \\ & \forall \mathbf{A}^g \subseteq \wp(\mathcal{I}_T \cup \mathcal{I}_N) \times \wp(\mathcal{I}_T \cup \mathcal{I}_N) : \\ & \quad \forall \Sigma \subseteq \mathcal{I}_N : \\ & \quad \mathbf{A}^a := (\mathbf{A}^g)^{-1} \\ & \mathbf{A}^g \text{ und } \mathbf{A}^a \text{ sind einfach assoziierend} \rightarrow \\ & \rightarrow S_a^{ve}(\mathcal{I}_T, \mathcal{I}_N, \Sigma, \mathbf{A}^a) = S_g(\mathcal{I}_T, \mathcal{I}_N, \Sigma, \mathbf{A}^g) \end{aligned} \quad (\text{B} - 140)$$

**Beweis** Aus  $S_g(\mathcal{I}_T, \mathcal{I}_N, \Sigma, \mathbf{A}^g) \subseteq S_a^{ve}(\mathcal{I}_T, \mathcal{I}_N, \Sigma, \mathbf{A}^a)$  (Lemma B.75) und  $S_a^{ve}(\mathcal{I}_T, \mathcal{I}_N, \Sigma, \mathbf{A}^a) \subseteq S_g(\mathcal{I}_T, \mathcal{I}_N, \Sigma, \mathbf{A}^g)$  (Lemma B.76) folgt  $S_a^{ve}(\mathcal{I}_T, \mathcal{I}_N, \Sigma, \mathbf{A}^a) = S_g(\mathcal{I}_T, \mathcal{I}_N, \Sigma, \mathbf{A}^g)$ .  $\square$

**Satz B.78** Für  $S_g^O(\mathcal{I}_T, \mathcal{I}_N, \Sigma, \mathbf{A}^g)$  und  $S_a^{te}(\mathcal{I}_T, \mathcal{I}_N, \Sigma, (\mathbf{A}^g)^{-1})$  gilt die Aussage:

$$\begin{aligned} & \forall \mathcal{I}_T, \mathcal{I}_N \subseteq \mathcal{I} : \\ & \forall \mathbf{A}^g \subseteq \wp(\mathcal{I}) \times \wp(\mathcal{I}) : \\ & \quad \forall \Sigma \subseteq \mathcal{I}_N : \\ & \quad \mathbf{A}^a := (\mathbf{A}^g)^{-1} \\ & \mathbf{A}^g \text{ und } \mathbf{A}^a \text{ sind einfach assoziierend} \rightarrow \\ & \rightarrow S_g^O(\mathcal{I}_T, \mathcal{I}_N, \Sigma, \mathbf{A}^g) = S_a^{te}(\mathcal{I}_T, \mathcal{I}_N, \Sigma, \mathbf{A}^a) \end{aligned} \quad (\text{B} - 141)$$

**Beweis** Aus den Definitionen Gl. 2-17 und Gl. 2-18 folgt, dass jede Obermenge  $\mathcal{T}$  einer Konfiguration  $\mathcal{R}$  von  $S_a^{ve}(\mathcal{I}_T, \mathcal{I}_N, \Sigma, \mathbf{A}^a)$  in  $S_a^{te}(\mathcal{I}_T, \mathcal{I}_N, \Sigma, \mathbf{A}^a)$  enthalten ist. Mit  $S_g(\mathcal{I}_T, \mathcal{I}_N, \Sigma, \mathbf{A}^g) = S_a^{ve}(\mathcal{I}_T, \mathcal{I}_N, \Sigma, \mathbf{A}^a)$  (Satz B.77) ist jede Obermenge einer Konfiguration  $\mathcal{R}$  von  $S_g(\mathcal{I}_T, \mathcal{I}_N, \Sigma, \mathbf{A}^g)$  in  $S_a^{te}(\mathcal{I}_T, \mathcal{I}_N, \Sigma, \mathbf{A}^a)$  enthalten. Darüber hinaus sind in  $S_a^{te}(\mathcal{I}_T, \mathcal{I}_N, \Sigma, \mathbf{A}^a)$  keine weiteren Konfigurationen enthalten. Mit Gleichung 2-16 definieren diese Obermengen  $S_g^O(\mathcal{I}_T, \mathcal{I}_N, \Sigma, \mathbf{A}^g)$ .  $\square$

**Satz B.79** Die Vereinigung der Konfigurationen der Mengen  $S_g$  aller vorstellbaren Generierungsverfahren  $S_g(\mathcal{I}_T, \mathcal{I}_N, \Sigma_2, \mathbf{A}_2^g)$  ist in jeder Menge  $S_g^O(\mathcal{I}_T, \mathcal{I}_N, \Sigma_1, \mathbf{A}_1^g)$  enthalten:

$$\begin{aligned} & \forall \mathbf{A}_1^g \subseteq \wp(\mathcal{I}) \times \wp(\mathcal{I}) : \\ & \quad \forall \Sigma_1 \subseteq \mathcal{I}_N : \\ & \bigcup_{\Sigma_2 \subseteq \mathcal{I}_N} \left[ \bigcup_{\mathbf{A}_2^g \subseteq \wp(\mathcal{I}) \times \wp(\mathcal{I})} \left( \bigcup_{\mathcal{R} \in S_g(\mathcal{I}_T, \mathcal{I}_N, \Sigma_2, \mathbf{A}_2^g)} \mathcal{R} \right) \right] \in S_g^O(\mathcal{I}_T, \mathcal{I}_N, \Sigma_1, \mathbf{A}_1^g) \end{aligned} \quad (\text{B} - 142)$$

**Beweis** Mit

$$\mathcal{O} := \bigcup_{\Sigma_2 \subseteq \mathcal{I}_N} \left[ \bigcup_{\mathbf{A}_2^g \subseteq \wp(\mathcal{I}) \times \wp(\mathcal{I})} \left( \bigcup_{\mathcal{R} \in S_g(\mathcal{I}_T, \mathcal{I}_N, \Sigma_2, \mathbf{A}_2^g)} \mathcal{R} \right) \right] \subseteq \mathcal{I}$$

und

$$\begin{aligned} & \forall T \subseteq \mathcal{I} : \\ & \forall \mathcal{R} \in S_g(\mathcal{I}_T, \mathcal{I}_N, \Sigma_1, \mathbf{A}_1^g) : \\ & \mathcal{R} \cup T \in S_g^O(\mathcal{I}_T, \mathcal{I}_N, \Sigma_1, \mathbf{A}_1^g) \end{aligned}$$

(Gl. 2-16) folgt (mit  $T = \mathcal{O}$ ):

$$\begin{aligned} & \forall \mathcal{R} \in S_g(\mathcal{I}_T, \mathcal{I}_N, \Sigma_1, \mathbf{A}_1^g) : \\ & \mathcal{R} \cup \mathcal{O} \in S_g^O(\mathcal{I}_T, \mathcal{I}_N, \Sigma_1, \mathbf{A}_1^g) \end{aligned}$$

□

**Lemma B.80** Zu jeder Ersetzung  $\mathcal{R} \xrightarrow[e]{*\mathbf{A}} \mathcal{S}$  gibt es eine Konfiguration  $\mathcal{Q}$ , die  $\mathcal{S}$  mit  $\mathbf{A}^{-1}$  zugeordnet wird und in der  $\mathcal{R}$  enthalten ist:

$$\begin{aligned} & \forall \mathbf{A} \subseteq \wp(\mathcal{I}) \times \wp(\mathcal{I}) : \\ & \forall \mathcal{R}, \mathcal{S} \subseteq \mathcal{I} : \\ & \mathcal{R} \xrightarrow[e]{*\mathbf{A}} \mathcal{S} \rightarrow \left\{ \begin{array}{l} \exists \mathcal{Q} \subseteq \mathcal{I} : \\ \mathcal{S} \xrightarrow[z]{*\mathbf{A}^{-1}} \mathcal{Q} \\ \mathcal{R} \subseteq \mathcal{Q} \end{array} \right. \quad (\text{B} - 143) \end{aligned}$$

**Beweis** 1. Teil: Es ist zu zeigen, dass für ein beliebiges  $i \in \mathbb{N}$  mit  $Z^{-1} := (\mathcal{I}, \mathbf{A}^{-1})$  aus

$$\mathcal{R} \xrightarrow[e]{(i)\mathbf{A}} \mathcal{U}_i \rightarrow \left\{ \begin{array}{l} \exists \mathcal{Q}_i \subseteq \mathcal{I} : \\ \mathcal{U}_i \mid \frac{(i)mZ^{-1}}{\mathcal{Q}_i} \\ \mathcal{R} \subseteq \mathcal{Q}_i \end{array} \right. \quad (\text{B} - 144)$$

und  $\mathcal{U}_i \xrightarrow[e]{\mathbf{A}} \mathcal{U}_{i+1}$  folgt:

$$\mathcal{R} \xrightarrow[e]{(i+1)\mathbf{A}} \mathcal{U}_{i+1} \rightarrow \left\{ \begin{array}{l} \exists \mathcal{Q}_{i+1} \subseteq \mathcal{I} : \\ \mathcal{U}_{i+1} \mid \frac{(i+1)mZ^{-1}}{\mathcal{Q}_{i+1}} \\ \mathcal{R} \subseteq \mathcal{Q}_{i+1} \end{array} \right. \quad (\text{B} - 145)$$

Mit  $\mathcal{U}_i \xrightarrow[e]{\mathbf{A}} \mathcal{U}_{i+1}$  erhält man  $\mathcal{R} \xrightarrow[e]{(i)\mathbf{A}} \mathcal{U}_i \xrightarrow[e]{\mathbf{A}} \mathcal{U}_{i+1}$  bzw.  $\mathcal{R} \xrightarrow[e]{(i+1)\mathbf{A}} \mathcal{U}_{i+1}$ . Außerdem gilt (Gl. 2-7):

$$\begin{aligned} & \exists (\mathcal{X}_i, \mathcal{Y}_i) \in \mathbf{A} : \\ & \mathcal{X}_i \subseteq \mathcal{U}_i \\ & \mathcal{U}_{i+1} = (\mathcal{U}_i \setminus \mathcal{X}_i) \cup \mathcal{Y}_i \end{aligned} \quad (\text{B} - 146)$$

Damit gilt auch  $\mathcal{Y}_i \subseteq \mathcal{U}_{i+1}$ . Mit  $(\mathcal{Y}_i, \mathcal{X}_i) \in \mathbf{A}^{-1}$  und

$$\mathcal{T} := \mathcal{U}_{i+1} \cup \mathcal{X}_i \quad (\text{B} - 147)$$

gibt es also die direkte mehrfache Zuordnung (Gl. B-26)

$$\mathcal{U}_{i+1} \mid \frac{mZ^{-1}}{\mathcal{T}}. \quad (\text{B} - 148)$$

$\mathcal{U}_{i+1}$  wird jetzt in der Definition von  $\mathcal{T}$  (Gl. B-147) entsprechend Gleichung B-146 durch  $\mathcal{U}_i$  ausgedrückt:

$$\mathcal{T} = [(\mathcal{U}_i \setminus \mathcal{X}_i) \cup \mathcal{Y}_i] \cup \mathcal{X}_i$$

Mit  $\mathcal{X}_i \subseteq \mathcal{U}_i$  (Gl. B-146) erhält man

$$\mathcal{T} = \mathcal{U}_i \cup \mathcal{Y}_i. \quad (\text{B} - 149)$$

Aus  $\mathcal{U}_i \stackrel{(i)mZ^{-1}}{\mid} \mathcal{Q}_i$  (Gl. B-144) folgt mit Lemma B.17  $(\mathcal{U}_i \cup \mathcal{Y}_i) \stackrel{(i)mZ^{-1}}{\mid} (\mathcal{Q}_i \cup \mathcal{Y}_i)$ . Man erhält dann mit Gleichung B-148 und  $\mathcal{T} = \mathcal{U}_i \cup \mathcal{Y}_i$  (Gl. B-149)

$$\mathcal{U}_{i+1} \stackrel{mZ^{-1}}{\mid} (\mathcal{U}_i \cup \mathcal{Y}_i) \stackrel{(i)mZ^{-1}}{\mid} (\mathcal{Q}_i \cup \mathcal{Y}_i)$$

bzw.

$$\mathcal{U}_{i+1} \stackrel{(i+1)mZ^{-1}}{\mid} (\mathcal{Q}_i \cup \mathcal{Y}_i).$$

Und es gilt  $\mathcal{R} \subseteq \mathcal{Q}_i \subseteq \mathcal{Q}_i \cup \mathcal{Y}_i$ . Aus Gleichung B-144 folgt also Gleichung B-145. Dabei ist  $\mathcal{Q}_{i+1} = \mathcal{Q}_i \cup \mathcal{Y}_i$ .

2. Teil: Für  $i = 0$  gelten  $\mathcal{R} \stackrel{(0)\mathbf{A}}{\underset{e}{\longrightarrow}} \mathcal{R}$  und  $\mathcal{R} \stackrel{(0)mZ^{-1}}{\mid} \mathcal{R}$ , so dass Gleichung B-144 für  $i = 0$  gilt. Damit gilt Gleichung B-144 für alle  $i \in \mathbb{N}$ .

Mit Gleichung B-27, Gleichung 2-7 und Gleichung 2-1 gilt:

$$\begin{aligned} & \forall \mathbf{A} \subseteq \wp(\mathcal{I}) \times \wp(\mathcal{I}) : \\ & \forall \mathcal{R}, \mathcal{S} \subseteq \mathcal{I} : \\ & \mathcal{R} \stackrel{* \mathbf{A}}{\underset{e}{\longrightarrow}} \mathcal{S} \rightarrow \left\{ \begin{array}{l} \exists \mathcal{Q} \subseteq \mathcal{I} : \\ \mathcal{S} \stackrel{* mZ^{-1}}{\mid} \mathcal{Q} \\ \mathcal{R} \subseteq \mathcal{Q} \end{array} \right. \quad (\text{B} - 150) \end{aligned}$$

Mit  $\stackrel{* mZ^{-1}}{\mid} = \stackrel{* Z^{-1}}{\mid}$  (Lemma B.16) und der Definition  $\stackrel{* Z^{-1}}{\mid} = \stackrel{* \mathbf{A}^{-1}}{\underset{z}{\longrightarrow}}$  (vgl. Seite 30) erhält man Gleichung B-143.  $\square$

**Satz B.81** Sei  $\mathbf{A}^a$  die Menge der analysierenden Assoziationen und Umkehrrelation der generierenden Menge der Assoziationen  $\mathbf{A}^g$ ,  $\mathcal{I}_T \cup \mathcal{I}_N$  die Menge der Instanzen und  $\Sigma \subseteq \mathcal{I}_N$  die Menge der Startsymbole, dann ist das generierte Szenario  $S_g(\mathcal{I}_T, \mathcal{I}_N, \Sigma, \mathbf{A}^g)$  im teilweise zuordnend analysierten Szenario  $S_a^{tz}(\mathcal{I}_T, \mathcal{I}_N, \Sigma, \mathbf{A}^a)$  enthalten:

$$\begin{aligned} & \forall \mathcal{I}_T, \mathcal{I}_N \subseteq \mathcal{I} : \\ & \forall \mathbf{A}^g \subseteq \wp(\mathcal{I}_T \cup \mathcal{I}_N) \times \wp(\mathcal{I}_T \cup \mathcal{I}_N) : \\ & \forall \Sigma \subseteq \mathcal{I}_N : \\ & S_g(\mathcal{I}_T, \mathcal{I}_N, \Sigma, \mathbf{A}^g) \subseteq S_a^{tz}(\mathcal{I}_T, \mathcal{I}_N, \Sigma, (\mathbf{A}^g)^{-1}) \end{aligned} \quad (\text{B} - 151)$$

**Beweis** Zu jedem  $\sigma \in \Sigma$  und jeder Ersetzung  $\{\sigma\} \stackrel{+\mathbf{A}^g}{\underset{e}{\longrightarrow}} \mathcal{R}$  gibt es eine Konfiguration  $\mathcal{Q}$  und eine Zuordnung  $\mathcal{R} \stackrel{+\mathbf{A}^a}{\underset{z}{\longrightarrow}} \mathcal{Q}$  mit  $\{\sigma\} \subseteq \mathcal{Q}$  (Lemma B.80), so dass aus  $\mathcal{R} \in S_g(\mathcal{I}_T, \mathcal{I}_N, \Sigma, \mathbf{A}^g)$  (Gl. 2-15) die Aussage  $\mathcal{R} \in S_a^{tz}(\mathcal{I}_T, \mathcal{I}_N, \Sigma, (\mathbf{A}^g)^{-1})$  (Gl. 2-19) folgt, d. h.  $S_g(\mathcal{I}_T, \mathcal{I}_N, \Sigma, \mathbf{A}^g) \subseteq S_a^{tz}(\mathcal{I}_T, \mathcal{I}_N, \Sigma, (\mathbf{A}^g)^{-1})$ .  $\square$

**Satz B.82** Sei  $\mathbf{A}^a$  die Menge der analysierenden Assoziationen und Umkehrrelation der generierenden Menge der Assoziationen  $\mathbf{A}^g$ ,  $\mathcal{I}_T \cup \mathcal{I}_N$  die Menge der Instanzen und  $\Sigma \subseteq \mathcal{I}_N$  die Menge der Startsymbole, dann sind die Obermengen  $\mathcal{S}$  jeder Szene  $\mathcal{R}$  des generierten Szenarios  $S_g(\mathcal{I}_T, \mathcal{I}_N, \Sigma, \mathbf{A}^g)$  im teilweise zuordnend analysierten Szenario  $S_a^{tz}(\mathcal{I}_T, \mathcal{I}_N, \Sigma, \mathbf{A}^a)$  enthalten:

$$\begin{aligned} & \forall \mathcal{I}_T, \mathcal{I}_N \subseteq \mathcal{I} : \\ & \forall \mathbf{A}^g \subseteq \wp(\mathcal{I}_T \cup \mathcal{I}_N) \times \wp(\mathcal{I}_T \cup \mathcal{I}_N) : \\ & \quad \forall \Sigma \subseteq \mathcal{I}_N : \\ & \quad \forall \mathcal{R} \in S_g(\mathcal{I}_T, \mathcal{I}_N, \Sigma, \mathbf{A}^g) : \\ & \quad \quad \forall \mathcal{T} \subseteq \mathcal{I}_T : \\ & \mathcal{R} \subseteq \mathcal{T} \rightarrow \mathcal{T} \in S_a^{tz}(\mathcal{I}_T, \mathcal{I}_N, \Sigma, (\mathbf{A}^g)^{-1}) : \end{aligned} \tag{B - 152}$$

**Beweis** Mit  $S_g(\mathcal{I}_T, \mathcal{I}_N, \Sigma, \mathbf{A}^g) \subseteq S_a^{tz}(\mathcal{I}_T, \mathcal{I}_N, \Sigma, (\mathbf{A}^g)^{-1})$  (Satz B.81) sind alle generierten Szenen  $\mathcal{R} \in S_g(\mathcal{I}_T, \mathcal{I}_N, \Sigma, \mathbf{A}^g)$  im teilweise zuordnenden Szenario  $S_a^{tz}(\mathcal{I}_T, \mathcal{I}_N, \Sigma, (\mathbf{A}^g)^{-1})$  enthalten. Mit Gl. 2-19 gilt  $\forall \mathcal{R} \in S_a^{tz}(\mathcal{I}_T, \mathcal{I}_N, \Sigma, \mathbf{A}^g) : \exists \mathcal{Q} \subseteq \mathcal{I} : \mathcal{R} \xrightarrow[z]{*\mathbf{A}} \mathcal{Q} \wedge \exists \sigma \in \Sigma : \sigma \in \mathcal{Q}$ . Mit Lemma B.17 gilt für alle  $\mathcal{T} \supseteq \mathcal{R}$  die Aussage  $\mathcal{T} \xrightarrow[z]{*\mathbf{A}} (\mathcal{Q} \cup \mathcal{T})$  bzw.  $\exists \sigma \in \Sigma : \sigma \in (\mathcal{Q} \cup \mathcal{T})$ , so dass  $\mathcal{T}$  ebenfalls in  $S_a^{tz}(\mathcal{I}_T, \mathcal{I}_N, \Sigma, \mathbf{A}^a)$  enthalten ist.  $\square$

**Satz B.83** Es kann generierte Konfigurationen  $\mathcal{R} \in S_g(\mathcal{I}_T, \mathcal{I}_N, \Sigma, \mathbf{A}^g)$  geben, die vollständig ersetzend nicht analysiert werden können:  $\mathcal{R} \notin S_a^{ve}(\mathcal{I}_T, \mathcal{I}_N, \Sigma, (\mathbf{A}^g)^{-1})$ .

**Beweis** Wenn  $\mathbf{A}^g$  mehrfach assoziierend ist, dann ist im Beweis von Lemma B.74 in Gleichung B-137  $\mathcal{U}_i$  ungleich  $\mathcal{T}$ . Das bedeutet, dass zu einem generierenden Ersetzungsschritt  $\mathcal{U}_i \xrightarrow[e]{\mathbf{A}} \mathcal{U}_{i+1}$  der korrespondierende analysierende Ersetzungsschritt  $\mathcal{U}_{i+1} \xrightarrow[e]{\mathbf{A}^{-1}} \mathcal{T}$  nicht passt und zu einer generierenden Ersetzung  $\{\sigma\} \xrightarrow[e]{*\mathbf{A}^g} \mathcal{R}$  die analysierende Ersetzung  $\mathcal{R} \xrightarrow[e]{*(\mathbf{A}^g)^{-1}} \{\sigma\}$  fehlt. In diesem Fall fehlt  $\mathcal{R}$  in  $S_a^{ve}(\mathcal{I}_T, \mathcal{I}_N, \Sigma, (\mathbf{A}^g)^{-1})$ .  $\square$

**Satz B.84** Es kann generierte Konfigurationen  $\mathcal{R} \in S_g(\mathcal{I}_T, \mathcal{I}_N, \Sigma, \mathbf{A}^g)$  geben, die in keiner teilweise ersetzend analysierten Konfiguration  $\mathcal{T} \in S_a^{te}(\mathcal{I}_T, \mathcal{I}_N, \Sigma, (\mathbf{A}^g)^{-1})$  enthalten sind.

**Beweis** Mit Satz B.83 kann es generierte Konfigurationen  $\mathcal{R}$  geben, die nicht vollständig ersetzend analysiert werden können. Grund hierfür ist, dass eine Instanz zweimal analysierend ersetzt werden müßte. Da sie ab der ersten Ersetzung nicht mehr zur Verfügung steht, kann die zweite Ersetzung nicht erfolgen, so dass die Analyse abbricht. Teilweise ersetzend gelten für diese doppelt zu ersetzende Instanz die gleichen Bedingungen, so dass auch in diesem Fall die Analyse erfolglos abbrechen kann.  $\square$

**Lemma B.85** Die Vereinigung von ersetzend generierten Konfigurationen kann nur mit mehrfach assoziierenden Assoziationen generiert werden.

**Beweis** Wenn zwei ersetzend generierte Konfigurationen  $\mathcal{S}_1$  und  $\mathcal{S}_2$  vereinigt werden und dabei beide Konfigurationen identische Instanzen enthalten, sind diese Instanzen nur einfach in der Vereinigungsmenge enthalten. Die erste Konfiguration sei dabei mit  $\mathbf{A}_1$ , die zweite Konfiguration mit  $\mathbf{A}_2$  generiert worden.  $\mathbf{A}_1$  und  $\mathbf{A}_2$  seien einfach assoziierend (Gl. B-72).

Die Vereinigungsmenge  $\mathcal{S}_1 \cup \mathcal{S}_2$  soll jetzt nicht durch die Vereinigung zweier terminaler Konfigurationen generiert werden, sondern durch zwei nacheinander auszuführende Folgen von Ersetzungsschritten. Die erste Folge generiert die erste Konfiguration  $\mathcal{S}_1$ , die zweite Folge ergänzt die erste Konfiguration. Dazu wird die Vereinigung  $\mathbf{A}_1 \cup \mathbf{A}_2$  der Mengen der Assoziationen verwendet. Wenn die zwei zu vereinigenden Konfigurationen identische Instanzen enthalten, gibt es bei der Ausführung der zweiten Folge einen Ersetzungsschritt  $\mathcal{K} \xrightarrow[e]{\mathbf{A}} \mathcal{L}$  mit einer Assoziation  $(\mathcal{X}, \mathcal{Y}) \in \mathbf{A}_2 \subseteq \mathbf{A}_1 \cup \mathbf{A}_2$ , bei dem neu hinzuzufügende Instanzen von  $\mathcal{Y}$  (d. h. von  $\mathcal{Y} \setminus \mathcal{X}$ ) bereits in der zu ändernden Konfiguration  $\mathcal{K}$  vorliegen. Bezogen auf Gleichung B-72 heißt dies, dass  $\mathcal{K} \cap (\mathcal{Y} \setminus \mathcal{X}) \neq \emptyset$  ist, so dass mehrfach assoziierende Assoziationen  $\mathbf{A}_1 \cup \mathbf{A}_2$  vorliegen.  $\square$

**Satz B.86** Die Vereinigung von ersetzend generierten Konfigurationen kann nur zuordnend, nicht aber ersetzend analysiert werden.

**Beweis** Aus Lemma B.85 folgt, dass die Vereinigung generierter Konfigurationen nur mit mehrfach assoziierenden Assoziationen generiert werden kann. Diese können zuordnend analysiert werden (Satz B.81). Ersetzend kann es Konfigurationen geben, die nicht analysiert werden können (Satz B.83 und Satz B.84).  $\square$

**Lemma B.87** Genau dann, wenn Instanzen Teil mehrerer anderer Instanzen sind, werden diese Instanzen mit mehrfach assoziierenden Assoziationen generiert:

$$\left. \begin{array}{l} \forall \mathbf{A} \in \wp(\mathcal{I}) \times \wp(\mathcal{I}) : \\ \exists (\mathcal{X}, \mathcal{Y}), (\mathcal{U}, \mathcal{V}) \in \mathbf{A} : \\ \exists \gamma \in \mathcal{Y} \setminus \mathcal{X} : \\ \gamma \in \mathcal{U} \setminus \mathcal{V} \end{array} \right\} \leftrightarrow \mathbf{A} \text{ ist mehrfach assoziierend} \quad (\text{B} - 153)$$

**Beweis** 1. Teil  $\rightarrow$ : Wenn eine Instanz  $\gamma$  Teil zweier Instanzen  $\alpha$  und  $\beta$  ist (d. h. z. B.  $\exists(\{\alpha\}, \{\gamma, \epsilon_{\alpha_1}, \epsilon_{\alpha_2}, \dots\}), (\{\beta\}, \{\gamma, \epsilon_{\beta_1}, \epsilon_{\beta_2}, \dots\}) \in \mathbf{A}$ ), so wird  $\gamma$  zuerst durch Ersetzung der ersten Instanz  $\alpha$  erzeugt (mit  $\mathcal{K} \xrightarrow[e]{\mathbf{A}} \mathcal{L}$ ). Anschließend wird  $\beta$  ersetzt. Dabei wird  $\gamma$  zu einer resultierenden Konfiguration  $\mathcal{K}'$  hinzugefügt, obwohl  $\gamma$  wegen der 1. Ersetzung bereits vorliegt (d. h.  $\gamma \in \mathcal{K}'$  und  $\gamma \in \mathcal{Y} \setminus \mathcal{X}$ ). Bezogen auf Gleichung B-72 heißt dies, dass  $\mathcal{K}' \cap (\mathcal{Y} \setminus \mathcal{X}) \neq \emptyset$  ist, so dass mehrfach assoziierende Assoziationen vorliegen.

2. Teil  $\leftarrow$ : Mengen von Assoziationen sind mehrfach assoziierend, wenn die Ersetzung zusammengesetzter Instanzen eine elementare Instanz generiert, die bereits vorliegt (d. h. vorher durch Ersetzung einer anderen zusammengesetzten Instanz generiert wurde; Gl. B-72). Damit ist diese elementare Instanz Teil mehrerer zusammengesetzter Instanzen.  $\square$

**Satz B.88** Sind Instanzen Teil mehrerer anderer Instanzen, so können diese Instanzen nur zuordnend, nicht aber ersetzend analysiert werden.

**Beweis** Wenn Instanzen Teil mehrerer anderer Instanzen sind, so können diese nur mit mehrfach assoziierenden Assoziationen generiert werden (Lemma B.87). Diese können zuordnend analysiert werden (Satz B.81). Ersetzend kann es Konfigurationen geben, die nicht analysiert werden können (Satz B.83 und Satz B.84).  $\square$

**Lemma B.89** Bei der Analyse von Instanzen, die Teil mehrerer Instanzen sind, entstehen Ableitungsgraphen.

**Beweis** Bei Aussagen über Ableitungsgraphen werden nur Instanzen  $\alpha, \beta$  betrachtet, die eine gemeinsame Wurzel  $\rho$  besitzen, d.h. Teil von  $\rho$  sind. Wenn diese Instanzen  $\alpha$  und  $\beta$  ihrerseits eine gemeinsame Instanz  $\gamma$  als Teil enthalten, entstehen Zyklen  $\rho, \alpha, \gamma, \beta, \rho$ , so dass ein allgemeiner Graph und kein Baum vorliegt.  $\square$

**Satz B.90** Bei der Analyse von Instanzen, die mehrfach assoziierend generiert werden, entstehen Ableitungsgraphen.

**Beweis** Instanzen, die Teil mehrerer Instanzen sind, werden mit mehrfach assoziierenden Assoziationen generiert (Lemma B.87). Dabei entstehen Ableitungsgraphen (Lemma B.89).  $\square$

## B.4 Aussagen für Aufwandsabschätzungen

Bei der Analyse mit regulär binären balancierten (rbb) Ableitungsbäumen werden Zwischenergebnisse, die nicht gespeichert werden, in der Regel mehrfach generiert (vgl. Gl. 3-13 und 3-14). Zum Vergleich der Algorithmen 3-10, 3-19 sowie der mit bewerteten Analysezuständen bzw. bewerteten Zwischenergebnissen arbeitenden Algorithmen (vgl. Abschnitte 3.3.2 und 3.3.3) wird ein Algorithmus (B-161) angegeben, der die Anzahl der Mehrfachgenerierungen für rbb-Ableitungsbäume berechnet.

Die Anzahl der unterschiedlichen Zwischenergebnisse  $\mu_{i,j}$  (mit Speicherung und ohne Mehrfachgenerierungen) bzw.  $\nu_{i,j}$  (ohne Speicherung und mit Mehrfachgenerierungen) wird mit  $t_{i,j}$  bezeichnet. Die Indices  $i$  und  $j$  beziehen sich dabei auf die Abbildungen B-1 bis B-3. Die Anzahl der insgesamt generierten bzw. analysierten Zwischenergebnisse  $\nu_{i,j}$  ist  $t_{i,j}^m$ . Die Anzahl der Mehrfachgenerierungen heißt  $m_{i,j}$  mit  $t_{i,j}^m = m_{i,j} \cdot t_{i,j}$  (Gl. 3-13).

**Mehrfachgenerierungen durch Schachtelung der Suchschleifen** In der nur ein Mal durchlaufenen ersten Doppelschleife von Algorithmus 3-10 mit den Laufindices  $l_1$  und  $l_2$  zur Generierung von  $\nu_{1,1}$  wird jedes dieser Ergebnisse nur ein Mal generiert, so dass keine Mehrfachgenerierungen auftreten:

$$m_{1,1} = 1 \quad (\text{B} - 154)$$

Jedes Ergebnis  $\nu_{1,1}$  veranlasst die Ausführung der zweiten Doppelschleife mit den Laufindices  $l_3$  und  $l_4$  zur Generierung von  $\nu_{1,2}$ , so dass die Anzahl deren

Mehrfachgenerierungen  $m_{1,2}$  gleich der Anzahl  $t_{1,1}^m$  der Zwischenergebnisse  $\nu_{1,1}$  ist ( $m_{1,2} = t_{1,1}^m$ ).

Entsprechend veranlasst jedes Ergebnis  $\nu_{1,2j-1}; j = 2, 3, \dots, |\mathcal{R}|/4$  mit ungeradem Index  $2j - 1$  die Ausführung der nachfolgenden Doppelschleife mit den Laufindices  $l_{4j-1}$  und  $l_{4j}$  zur Generierung von  $\nu_{1,2j}$  mit geradem Index  $2j$  (Abbildung B-1). Dabei gilt analog zu  $m_{1,2} = t_{1,1}^m$ , dass die Anzahl der Mehrfachgenerierungen  $m_{1,2j}$  der Zwischenergebnisse  $\nu_{1,2j}$  gleich der Anzahl  $t_{1,2j-1}^m$  der Zwischenergebnisse  $\nu_{1,2j-1}$  ist ( $m_{1,2j} = t_{1,2j-1}^m$ ).

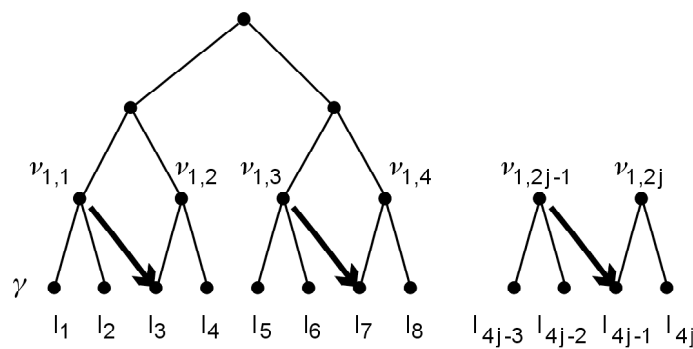


Abb. B-1: Mehrfachgenerierungen durch geschachtelte Suchschleifen (1. Ebene)

Zwischenergebnisse  $\nu_{i,2j-1}$  mit ungeradem Index auf höherer Stufe  $i$  veranlassen analog, dass die Suchschleifen mit den Indices  $(2j - 1) \cdot 2^i + 1$  und  $(2j - 1) \cdot 2^i + 2$  durchlaufen werden (Abbildung B-2 mit  $i = m$  und  $2j - 1 = n$ ). Dabei werden auf der 1. Stufe Zwischenergebnisse  $\nu_{1,(2j-1) \cdot 2^{i-1} + 1}$  generiert. Analog  $m_{1,2j} = t_{1,2j-1}^m$  erhält man jetzt

$$m_{1,(2j-1) \cdot 2^{i-1} + 1} = t_{i,2j-1}^m. \tag{B - 155}$$

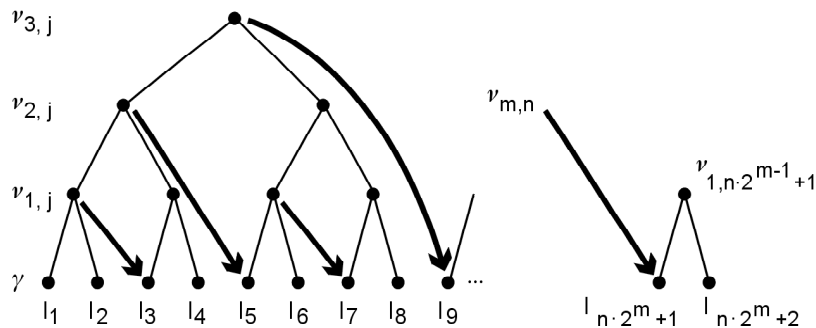


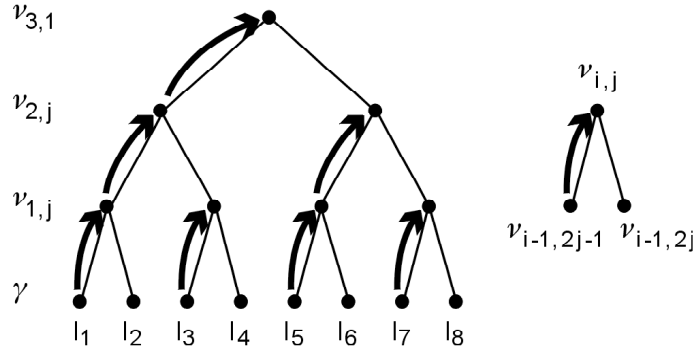
Abb. B-2: Mehrfachgenerierungen durch geschachtelte Suchschleifen

### Fortpflanzung von Mehrfachgenerierungen

**Lemma B.91** Zwischenergebnisse werden so oft mehrfach generiert, wie das erste Vorgängerergebnis mehrfach generiert wird (Abbildung B-3):

$$\begin{aligned} \forall i = 1, 2, \dots, ld(|\mathcal{R}|) : \\ \forall j = 1, 2, \dots, |\mathcal{R}|/2^i : \\ m_{i,j} = m_{i-1,2j-1} \end{aligned} \quad (\text{B} - 156)$$

$m_{0,l}$  ist dabei die Anzahl der Überprüfungen der terminalen Instanz  $\gamma_l$ .



**Abb. B-3:** Fortpflanzung von Mehrfachgenerierungen

Mit  $i' = i - 1$  und  $j' = 2j - 1$  (d. h.  $j'$  ungerade) gilt auch:

$$\begin{aligned} \forall i' = 0, 1, 2, \dots, ld(|\mathcal{R}|) - 1 : \\ \forall j' = 1, 3, 5, \dots, |\mathcal{R}|/2^{i'-1} - 1 : \\ m_{i'+1,(j'+1)/2} = m_{i',j'} \end{aligned} \quad (\text{B} - 157)$$

**Beweis** 1. Teil: Mehrfachgenerierungen auf der 1. Stufe: Jedes Zwischenergebnis  $\nu_{1,j}$  der 1. Stufe wird mit einer geschachtelten Suchschleife so oft generiert, wie die terminale Instanz  $\gamma_{l_{2j-1}}$  überprüft wird.

Die Aussage von Lemma B.91 stimmt also auf der 1. Stufe.

2. Teil: Es wird angenommen, dass die Aussage von Lemma B.91 auf der Stufe  $i$  zutrifft.

Nach Induktionsvoraussetzung ist in Abbildung B-4 die Anzahl der Mehrfachgenerierungen der ersten Zwischenergebnisse  $\nu_{i,l}; l = 1, 2, \dots, i$  von Stufe 1 bis Stufe  $i$  gleich.

Im mittleren rbb-Baum von Abbildung B-4 wird die erste Suchschleife mit  $\gamma_l$  und  $\gamma_{l+1}$  mit  $l = 2^i + 1$  für jedes Ergebnis des linken rbb-Baums  $\nu_{i,1}$  ein Mal durchsucht (Abbildung B-2). Nach Induktionsvoraussetzung wird im mittleren rbb-Baum das Ergebnis  $\nu_{i,2}$  genauso oft generiert, wie  $\gamma_l$  mit  $l = 2^i + 1$  überprüft wird. Es entsteht eine Suchschleife mit den Ergebnissen der zwei rbb-Bäume  $\nu_{i,1}$  und  $\nu_{i,2}$ . Diese Suchschleife generiert das Ergebnis  $\nu_{i+1,1}$  genauso oft wie  $\nu_{i,1}$  generiert wird.

Aus der Annahme, dass die Aussage von Lemma B.91 für die Stufe  $i$  gilt, folgt also, dass die Aussage von Lemma B.91 für die Stufe  $i + 1$  gilt.



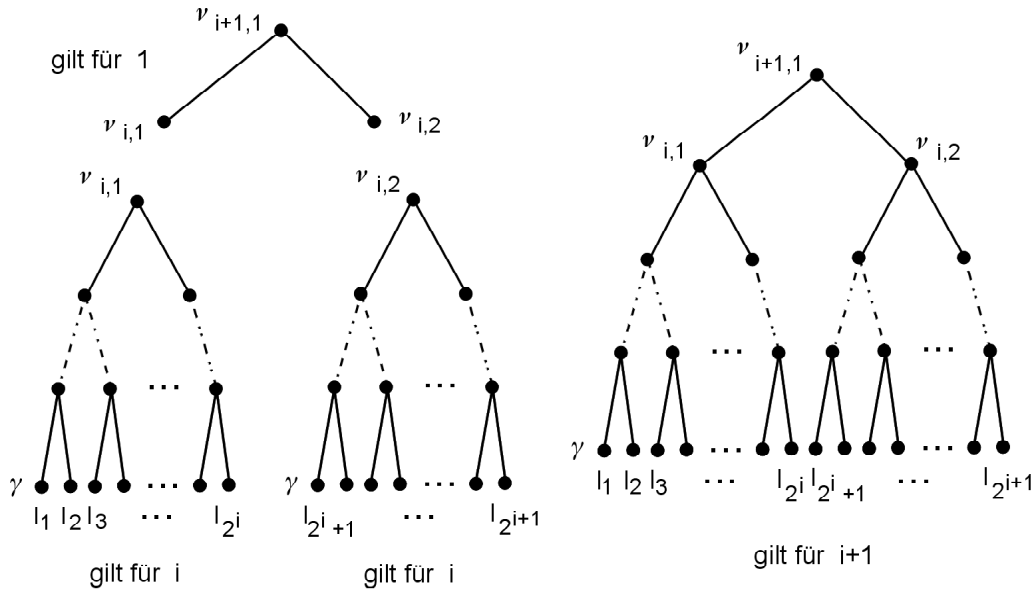


Abb. B-4: Zum Beweis von Lemma B.91

3. Teil: Die Aussage von Lemma B.91 gilt für die erste Stufe mit  $i = 1$  (1. Teil des Beweises). Mit dem 2. Teil des Beweises gilt die Aussage von Lemma B.91 für alle Stufen  $i \in \mathbb{N}$ .  $\square$

**Algorithmus zur Berechnung von Mehrfachgenerierungen für rbb-Ableitungsbäume**

**Lemma B.92** Das 1. Zwischenergebnis jeder Stufe  $i$  wird einfach generiert.

**Beweis** Für ein beliebiges  $i' \in \mathbb{N}$  folgt aus Gleichung B-157  $m_{i',1} = m_{i'+1,(1+1)/2} = m_{i'+1,1}$ . Mit  $m_{1,1} = 1$  (Gl. B-154) gilt also

$$\forall i = 1, 2, \dots, ld(|\mathcal{R}|) : \quad m_{i,1} = 1. \quad (\text{B} - 158)$$

$\square$

**Lemma B.93** Der Nachfolger von  $\nu_{1,(2j-1) \cdot 2^{i-1} + 1}$  auf der Stufe  $k \leq i$  ist  $\nu_{k,(2j-1) \cdot 2^{i-k} + 1}$ .

**Beweis** Sei  $\nu_{k,(2j-1) \cdot 2^{i-k} + 1}$  für ein beliebiges  $k$  Nachfolger von  $\nu_{1,(2j-1) \cdot 2^{i-1} + 1}$  auf der Stufe  $k$ . Dann folgt mit Gleichung B-157, dass  $\nu_{k+1,((2j-1) \cdot 2^{i-k} + 2)/2} = \nu_{k+1,(2j-1) \cdot 2^{i-(k+1)} + 1}$  Nachfolger von  $\nu_{k,(2j-1) \cdot 2^{i-1} + 1}$  ist, so dass die Aussage für  $k + 1$  gilt.

Für  $k = 1$  erhält man das Ausgangsergebnis  $\nu_{1,(2j-1) \cdot 2^{i-1} + 1}$ , dessen Nachfolger gesucht werden.

Die Aussage gilt also für alle Stufen  $k \in \mathbb{N}$  mit  $k \leq i$ .  $\square$

**Lemma B.94** Das  $l$ . Zwischenergebnisse  $\nu_{k,l}$  der Stufe  $k$  mit  $l = (2j - 1) \cdot 2^{i-k} + 1$  wird so oft generiert, wie das die Generierung veranlassende Zwischenergebnis  $\nu_{i,2j-1}$ .

$$\begin{aligned} \forall i = 1, 2, \dots, ld(|\mathcal{R}|) - 1 : \\ \forall j = 1, 2, \dots, |\mathcal{R}|/2^{i+1} : \\ \forall k = 1, 2, \dots, i : \\ t_{i,2j-1}^m = m_{k,(2j-1) \cdot 2^{i-k} + 1} \end{aligned} \quad (\text{B} - 159)$$

**Beweis** Mit Gleichung B-155 gilt  $m_{1,(2j-1) \cdot 2^{i-1} + 1} = t_{i,2j-1}^m$ .

Die Nachfolger  $\nu_{k,(2j-1) \cdot 2^{i-k} + 1}; k \in \mathbb{N}$  von  $\nu_{1,(2j-1) \cdot 2^{i-1} + 1}$  (Lemma B.93) werden genauso oft mehrfach generiert wie  $\nu_{1,(2j-1) \cdot 2^{i-1} + 1}$  (Lemma B.91).  $\square$

**Korollar B.95** Die Anzahl der Mehrfachgenerierungen  $m_{i,2j}$  der Ergebnisse  $\nu_{i,2j}$  mit geraden Indices ist gleich der Anzahl  $t_{i,2j-1}^m$  der Ergebnisse  $\nu_{i,2j-1}$  mit ungeraden Indices:

$$\begin{aligned} \forall i = 1, 2, \dots, ld(|\mathcal{R}|) - 1 : \\ \forall j = 1, 2, \dots, |\mathcal{R}|/2^{i+1} : \\ m_{i,2j} = t_{i,2j-1}^m \end{aligned} \quad (\text{B} - 160)$$

**Beweis** Folgt mit  $k = i$  aus Gleichung B-159:  $m_{i,(2j-1) \cdot 2^{i-i} + 1} = m_{i,2j} = t_{i,2j-1}^m$   $\square$

**Satz B.96** Der folgende Algorithmus berechnet die Anzahl der Mehrfachgenerierungen  $m_{i,j}$  der Zwischenergebnisse  $\nu_{i,j}$  (bzw. das Argument  $M$  der Funktion *Reduziere*) aus der Anzahl  $t_{k,l}$  der jeweils nur ein Mal generierten Zwischenergebnisse  $\mu_{k,l}$ :

- 1: Funktion *Reduziere* ( $i, j, M$ )
- 2: falls  $j$  gleich 1: Ende
- 3: sonst:
- 4: falls  $j$  gerade:
- 5:  $j := j - 1$
- 6:  $M := M \cdot t_{i,j}$
- 7: *Reduziere* ( $i, j, M$ )
- 8: falls  $j$  ungerade:
- 9:  $i := i + 1$  (B - 161)
- 10:  $j := (j + 1)/2$
- 11: *Reduziere* ( $i, j, M$ )
- 12: Ende Funktion *Reduziere*
- 13:  $M := 1$
- 13: lies  $i$  und  $j$  ein
- 14: *Reduziere* ( $i, j, M$ )
- 15:  $M$  ist die Anzahl  $m_{i,j}$  der Mehrfachgenerierungen des Zwischenergebnisses  $\nu_{i,j}$

**Beweis** Zeile 2: Alle Zwischenergebnisse  $\nu_{i,1}, i = 1, 2, \dots$  werden einfach generiert, und ihre Anzahl hängt von keinem anderen Ergebnis ab (Gl. B-158).

Zeile 5 und 6: Alle Zwischenergebnisse  $\nu_{i,2k}$  (mit geradem Index  $2k = j$ ) werden so oft mehrfach generiert, wie es Zwischenergebnisse  $\nu_{i,2k-1}$  gibt (Gl. B-160).

Jeder gerade Index  $j$  wird mit der Anweisung  $j := j - 1$  in Zeile 5 verringert. Sobald der Wert  $j = 1$  erreicht wird, terminiert der Algorithmus in Zeile 2.

Zeilen 9 und 10: Alle Zwischenergebnisse  $\nu_{i,j}$  mit ungeradem Index  $j$  werden so oft mehrfach generiert, wie das Zwischenergebnis  $\nu_{i+1,(j+1)/2}$  mehrfach generiert wird (Gl. B-157).

Jeder Index  $j \geq 3$  wird in den Zeilen 9 und 10 mit der Anweisung  $j := (j+1)/2$  bzw. in der Zeile 5 mit  $j := j-1$  verkleinert. Deshalb wird die Tiefe der Rekursion begrenzt:  $j = 3$  wird zu  $j = 2$  und beim nächsten Aufruf der Funktion in Zeile 5 zu  $j = 1$ , so dass zurück gesprungen wird.  $\square$

**Weitere Aussagen**

**Lemma B.97** Es gilt

$$\sum_{l=1}^{k-1} \frac{a}{2^l} = a - \frac{a}{2^{k-1}} \quad (\text{B} - 162)$$

**Beweis** Für  $k = 2$  gilt  $\sum_{l=1}^1 \frac{a}{2^l} = \frac{a}{2} = a - \frac{a}{2}$ . Sei  $\sum_{l=1}^{k-1} \frac{a}{2^l} = a - \frac{a}{2^{k-1}}$  für ein beliebiges  $k > 1$

wahr. Aus der Aussage für  $k$  folgt, dass die Aussage für  $k+1$  gilt:  $\sum_{l=1}^{(k+1)-1} \frac{a}{2^l} =$

$\sum_{l=1}^{k-1} \frac{a}{2^l} + \frac{a}{2^k} = a - \frac{a}{2^{k-1}} + \frac{a}{2^k} = a - \frac{a}{2^k} = a - \frac{a}{2^{(k+1)-1}}$ . Damit gilt Gleichung B-162 für alle  $k > 1$ .  $\square$

**Lemma B.98** Wenn  $t_{i,2j-1}^m > 1$  ist, gilt

$$1 + \sum_{i=1}^{ld(|\mathcal{R}|-1)} \left( \sum_{j=1}^{|\mathcal{R}|/2^{(i+1)}} t_{i,2j-1}^m \right) > |\mathcal{R}|/2. \quad (\text{B} - 163)$$

**Beweis** Aus  $t_{i,2j-1}^m > 1$  folgt

$$\sum_{i=1}^{ld(|\mathcal{R}|-1)} \left( \sum_{j=1}^{|\mathcal{R}|/2^{(i+1)}} t_{i,2j-1}^m \right) > \sum_{i=1}^{ld(|\mathcal{R}|-1)} \left( \sum_{j=1}^{|\mathcal{R}|/2^{(i+1)}} 1 \right) = \sum_{i=1}^{ld(|\mathcal{R}|-1)} \left( \frac{|\mathcal{R}|}{2^{i+1}} \right).$$

Mit  $k = ld(|\mathcal{R}|)$ ,  $a = |\mathcal{R}|/2$ ,  $|\mathcal{R}| > 2$  (d. h.  $k > 1$ ) erhält man (Lemma B.97)

$$\sum_{i=1}^{ld(|\mathcal{R}|-1)} \left( \frac{|\mathcal{R}|}{2^{i+1}} \right) = |\mathcal{R}|/2 - \frac{|\mathcal{R}|/2}{2^{ld(|\mathcal{R}|-1)}} = |\mathcal{R}|/2 - 1.$$

Daraus folgt Gleichung B-163.  $\square$

**Satz B.99** Wenn die Anzahl  $|\mathcal{T}|$  der zu analysierenden terminalen Instanzen der Menge  $\mathcal{T}$  wesentlich größer ist als die Anzahl  $|\mathcal{R}|$  der zu einem Startsymbol zusammzufassenden terminalen Instanzen der Menge  $\mathcal{R}$  und  $|\mathcal{R}|$  größer als 2 ist, benötigt Algorithmus 3-10 mehr Überprüfungen als Algorithmus 3-19.

**Beweis** Für  $|\mathcal{T}| \gg |\mathcal{R}|$  ist die Anzahl der Überprüfungen mit Algorithmus 3-10

$$n_{tbo} = \sum_{i=1}^{ld(|\mathcal{R}|-1)} \left( \sum_{j=1}^{|\mathcal{R}|/2^{(i+1)}} t_{i,2j}^m \right) + \frac{t_0^2}{2} \cdot \left[ 1 + \sum_{i=1}^{ld(|\mathcal{R}|-1)} \left( \sum_{j=1}^{|\mathcal{R}|/2^{(i+1)}} t_{i,2j-1}^m \right) \right]$$

(Gl. 3-12) und mit Algorithmus 3-19

$$n_{t_{bm}} := \frac{|\mathcal{R}|}{2} \cdot \frac{t_0^2}{2} + \sum_{i=2}^{ld(|\mathcal{R}|)} \left( \sum_{j=1}^{|\mathcal{R}|/2^i} t_{i-1,2j-1} \cdot t_{i-1,2j} \right)$$

(Gl. 3-20).

Für den Vergleich von  $n_{t_{bo}}$  und  $n_{t_{bm}}$  wird zuerst die linke Summe

$$\sum_{i=1}^{ld(|\mathcal{R}|-1)} \left( \sum_{j=1}^{|\mathcal{R}|/2^{(i+1)}} t_{i,2j}^m \right)$$

in Gleichung 3-12 für  $n_{t_{bo}}$  mit der rechten Summe

$$\sum_{i=2}^{ld(|\mathcal{R}|)} \left( \sum_{j=1}^{|\mathcal{R}|/2^i} t_{i-1,2j-1} \cdot t_{i-1,2j} \right)$$

in Gleichung 3-20 für  $n_{t_{bm}}$  verglichen.

Anschließend wird die rechte Summe

$$\frac{t_0^2}{2} \cdot \left[ 1 + \sum_{i=1}^{ld(|\mathcal{R}|-1)} \left( \sum_{j=1}^{|\mathcal{R}|/2^{(i+1)}} t_{i,2j-1}^m \right) \right]$$

in Gleichung 3-12 für  $n_{t_{bo}}$  mit dem linken Term

$$\frac{t_0^2}{2} \cdot \frac{|\mathcal{R}|}{2}$$

in Gleichung 3-20 für  $n_{t_{bm}}$  verglichen.

Für den ersten Vergleich wird mit  $t_{i,2j}^m = t_{i,2j} \cdot m_{i,2j}$  (Gl. 3-13) und  $m_{i,2j} = t_{i,2j-1}^m$  (Gl. B-160) der Faktor  $t_{i,2j}^m$  in Gleichung 3-12 durch  $t_{i,2j} \cdot t_{i,2j-1}^m$  ersetzt. Mit  $m_{i,j} \geq 1$  (Anzahl der Mehrfachgenerierungen) und  $t_{i,2j-1}^m = t_{i,2j-1} \cdot m_{i,2j-1}$  (Gl. 3-13) gilt

$$t_{i,2j}^m = t_{i,2j-1}^m \cdot t_{i,2j} \geq t_{i,2j-1} \cdot t_{i,2j},$$

so dass die (linke) Summe

$$\sum_{i=1}^{ld(|\mathcal{R}|-1)} \left( \sum_{j=1}^{|\mathcal{R}|/2^{(i+1)}} t_{i,2j}^m \right)$$

in Gleichung 3-12 für  $n_{t_{bo}}$  größer gleich der (rechten) Summe

$$\sum_{i=1}^{ld(|\mathcal{R}|-1)} \left( \sum_{j=1}^{|\mathcal{R}|/2^{(i+1)}} t_{i,2j-1} \cdot t_{i,2j} \right) = \sum_{i=2}^{ld(|\mathcal{R}|)} \left( \sum_{j=1}^{|\mathcal{R}|/2^i} t_{i-1,2j-1} \cdot t_{i-1,2j} \right)$$

in Gleichung 3-20 für  $n_{t_{bm}}$  ist.

Für den zweiten Vergleich erhält man mit Lemma B.98 - für das  $ld(|\mathcal{R}|) > 1$ , d. h.  $|\mathcal{R}| > 2$  sein muss - die Aussage

$$1 + \sum_{i=1}^{ld(|\mathcal{R}|-1)} \left( \sum_{j=1}^{|\mathcal{R}|/2^{(i+1)}} t_{i,2j-1}^m \right) > |\mathcal{R}|/2,$$

so dass auch die (rechte) Summe

$$\frac{t_0^2}{2} \left[ 1 + \sum_{i=1}^{ld(|\mathcal{R}|-1)} \left( \sum_{j=1}^{|\mathcal{R}|/2^{(i+1)}} t_{i,2j-1}^m \right) \right]$$

von Gleichung 3-12 für  $n_{tbo}$  größer als der (linke) Term

$$\frac{t_0^2}{2} \cdot \frac{|\mathcal{R}|}{2}$$

von Gleichung 3-20 für  $n_{tbn}$  ist. □

**Lemma B.100** Zwei geschachtelte Schleifen des Typs:

für  $l_1 := 1, t-1$   
 für  $l_2 := l_1 + 1, t$   
 führe Test aus

führen den Test  $\frac{1}{2} \cdot t \cdot (t-1)$  Mal aus.

**Beweis**  $\sum_{l_1=1}^{t-1} \left( \sum_{l_2=l_1+1}^t 1 \right) = \sum_{l_1=1}^{t-1} (t-l_1) = \sum_{l_1=1}^{t-1} t - \sum_{l_1=1}^{t-1} l_1 = t \cdot (t-1) - \frac{1}{2} \cdot t \cdot (t-1) = \frac{1}{2} \cdot t \cdot (t-1)$  □

## B.5 Projektionen von Instanzen, Konfigurationen und Assoziationen

**Definition B.101** Seien  $u := \{1, 2, \dots, |u|\}$  eine Indexmenge und  $\Omega_i, i \in u$  Mengen sowie  $U := \{\Omega_i | i \in u\}$ . Mit der üblichen Tupelschreibweise für Elemente eines kartesischen Produktes von Mengen  $\Omega_i$  wird die Projektion  $\pi_S$  eines Tupels (einer Instanz) auf die Menge  $S \subseteq U$  durch Streichen einzelner Komponenten des Tupels definiert:

$$\left. \begin{array}{l} \pi_i : \prod_{\Omega \in U} \Omega \rightarrow \Omega_i \\ \gamma \mapsto \gamma^{(\Omega_i)} \\ \forall \gamma = (\alpha_1, \alpha_2, \dots, \alpha_{|u|}) \in \Omega_1 \times \Omega_2 \times \dots \times \Omega_{|u|} : \\ \gamma^{(\Omega_i)} := \alpha_i \end{array} \right\}; i \in u \quad (\text{B-164})$$

$$\begin{array}{l} \forall S = \{\Omega'_1, \Omega'_2, \dots, \Omega'_{|S|}\} \in \wp(U) \setminus \emptyset : \\ \pi_S : \prod_{\Omega \in U} \Omega \rightarrow \prod_{\Omega' \in S} \Omega' \\ \gamma \mapsto \gamma^{(S)} \\ \gamma^{(S)} := (\gamma^{(\Omega'_1)}, \gamma^{(\Omega'_2)}, \dots, \gamma^{(\Omega'_{|S|})}) \end{array} \quad (\text{B-165})$$

Aufbauend auf der Projektion  $\pi_S$  eines Tupels werden Projektionen für Mengen von Tupeln (z. B. Konfigurationen)  $\bar{\pi}_S$  und Projektionen für Relationen in Mengen von Tupeln (z. B. Assoziationen)  $\bar{\bar{\pi}}_S$  definiert:

$$\begin{array}{l} \bar{\pi}_S : \wp \left( \prod_{\Omega \in U} \Omega \right) \rightarrow \wp \left( \prod_{\Omega' \in S} \Omega' \right) \\ \mathcal{L} \mapsto \mathcal{L}^{(S)} \\ \mathcal{L}^{(S)} := \{ \gamma^{(S)} | \gamma \in \mathcal{L} \} \end{array} \quad (\text{B-166})$$

$$\bar{\pi}_S : \wp \left[ \wp \left( \prod_{\Omega \in U} \Omega \right) \times \wp \left( \prod_{\Omega \in V} \Omega \right) \right] \rightarrow \wp \left[ \wp \left( \prod_{\Omega' \in S} \Omega' \right) \times \wp \left( \prod_{\Omega' \in S} \Omega' \right) \right] \quad (\text{B} - 167)$$

$$\mathbf{A} \mapsto \mathbf{A}^{(S)}$$

$$\mathbf{A}^{(S)} := \{ (\mathcal{U}^{(S)}, \mathcal{V}^{(S)}) \mid (\mathcal{U}, \mathcal{V}) \in \mathbf{A} \}$$

□

## Stichwortverzeichnis

- A als PDL-Muster 86
- Äquidensiten 108
- A\*-Optimierung 92
- Ablaufbeispiel 12
- Ablaufprinzip 8
- Ableitung 25 36
- Ableitungsgraph 25 26 45 47 51
- Agenda 91
- Aktionsteil 90
- Analyse
  - Ablauf 52
  - Aufwand 52 53
  - Erweiterung 47
  - explorativ 28
  - Fehler 44
  - Funktion 16
  - mit Parallelrechner 28
  - Probleme 46
  - Qualität 43 46
  - Testdaten 48
  - Trainingsdaten 47
  - vollständig ersetzend 39 41
  - Zeit 118
- analysierbarer Zustand 27 167
- Anzahl der Zwischenergebnisse 53
- Approximationslänge 109
- Assoziation 24
  - anlagernd 26 51
  - einfach assoziierend 39
  - graph-differenzierend 25
  - kontextsensitiv 50
  - lokal kontextfrei 26
  - mehrfach assoziierend 26 47 52
  - merkmal-differenzierend 25
- Aufwand 52 53
  - Begrenzung bei der Analyse 75
  - Minimierung bei Zuordnungsautomaten 80
  - Minimierung bei Zuordnungssystemen 78
  - Reduktion 50
- Ausdehnung einer Instanz 37 51
- Auswahlfunktion 27 29 80
- Auswahlstrategie 79
- Ballungsanalyse 95
- Bedingungsteil 90
- Bereich einer Instanz 37
- beschreibendes Merkmal 26 37
- Beschreibungsgraph 25
- Bewertung 9 81
- Bildanalyse 37 41
  - linguistisch 82
  - syntaktisch 16 82
- Bilder 33
  - rekonstruiert 111
  - überlagert 115
- Bildinhalte 43
- BPI-Architektur 28
- CFAR 77
- Chromosomen-Grammatik 82
- CYK 94
- Dämon 6
- demon 6
- dynamische Programmierung 94
- Einbettung 17 25
- einfach assoziierende Assoziation 39
- Entwickler 51
- Erkennung
  - erzeugter Konfigurationen 39
  - kontextfrei und kontextsensitiv 51
  - mit Ableitungen 38
  - mit Konzepten 38
  - zuordnend 36 48
- Erkennungsverfahren 33
- ERNEST 92
- ersetzen
  - von Konfigurationen 18
  - von Zeichenketten 16
- ersetzend erkennen 35
- ersetzende Analyse 39
- Ersetzungsfolgen 46

- erweiterte syntaktische Analyseverfahren 84
- Erzeugung
  - vexierend 47
- Erzeugungsverfahren 31 33
- explorative Analyse 28
- Extraktionsprogramm 46 47
- Fehlende Terminale 47
- Fehler 44
- Flugzeugmessprogramm 107
- FMP 107
- generieren
  - ersetzend 33
  - mit Obermenge 34
- Gesicht 47
- Gestaltpsychologie 1
- Grauwertbild
  - rekonstruiertes 110
- Grauwertgebirge 108 118
- HEARSAY II 90
- Hintergrund 44
  - Trennung vom Objekt 41
- Höhenlinien 108
- Hypothesengenerierung 8
- Hypothesenüberprüfung 9
- Instanz
  - Ausdehnung 37
  - Bereich 37
  - lokal überprüfbar 29 175
  - nicht analysierbar 40
  - nichtterminal 31
  - Obermengen 40
  - terminal 31
  - verschmolzen 47 51
- Instanziierungsfunktion 24
- Konfiguration 17 33
  - erkannte 37
  - Erkennung erzeugter 39
- Konfluenz 15
  - strenge 15
- Konkatenation 16
- Kontext-Objekt 44
- kontextfrei 26 51
- kontextsensitiv 50 51
- Konvergenz 16
- Konzept 24
  - instanziiert 24
  - modifiziert 54 93
  - Repräsentant 24
- Koordinatengrammatik 25 84 87
- level slicing 3 108
- linguistische Bildanalyse 82
- Literaturverweise
  - /Aho72/ 94
  - /Alb78/ 107
  - /And88/ 6 89
  - /ATh89/ 3 108
  - /Ave95/ 14 15 16 17 136 139 140
  - /Bäh95/ 8 100 107
  - /Bal82/ 6 7 91 97 99
  - /Bar81/ 7 90
  - /Bel57/ 94
  - /Ben96/ 7 53 54 70 92 95
  - /Ber80/ 107
  - /Bin82/ 7 91
  - /Boo93/ 14
  - /Bra77/ 90
  - /Bun85/ 7 70 91 92
  - /Bun90/ 7 82 85 86 94
  - /Cae97/ 7 82 89 91
  - /Cho65/ 32
  - /Dör88/ 54
  - /Erm80/ 90
  - /For81/ 91
  - /Füg92/ 8 100 107
  - /Füg94/ 8 100 107
  - /Fu77/ 7 82 83 85
  - /Ger79/ 107
  - /Gib50/ 7
  - /Gon78/ 13 17 82 83 84 85 86
  - /Gon82/ 7
  - /Gra00/ 7 92 93
  - /Han78/ 7
  - /Hrb85/ 6
  - /Hrf85/ 90
  - /HRi78/ 90



- /Jan88/ 13 14 17
- /Kaz80/ 3
- /Koe69/ 1
- /Kum97/ 97
- /Leh85/ 90
- /Lev78/ 90
- /Lüt86/ 8 107 108 111
- /Mar98/ 7 82 85 86
- /Mat90/ 82
- /Mich96/ 25 84
- /Mich98a/ 7 8 25 35 84 86 87 88 89  
92 100 107
- /Mich98b/ 8 100 107
- /Mil72/ 25 84 87
- /Naz83/ 7 90
- /New62/ 7 89
- /Nie89/ 3 7 90 91 92
- /Nie90/ 92
- /Oht85/ 7 90 91
- /Poo98/ 7 53 54 70 92 95
- /Qui97/ 7 53 70 92 93
- /Ros79/ 7
- /Rub21/ 1
- /Rus95/ 53 54
- /Sag85/ 7 53 70 92 93
- /Sag97/ 54 92 93
- /Sal78/ 16 31 83
- /Sch83/ 54
- /Schä98/ 8 100 107
- /Schw98/ 8 100 107
- /Sel58/ 6 89
- /Sha69/ 25 84 87
- /Sha70/ 85
- /Sha79/ 7 90
- /Sim66/ 7 89
- /Ste88/ 31 90
- /Ste98/ 111
- /Sti93/ 8 100 107
- /Sti94/ 8 100 107
- /Sti95a/ 8 100 107
- /Sti95b/ 8 100 107
- /Sti96a/ 8 100 107
- /Sti96b/ 8 100 107
- /Sti97a/ 8 100 107
- /Sti97b/ 8 100 107
- /Sti97c/ 8 100 107
- /Sti98/ 8 100 107
- /Tel79/ 107
- /Ter81/ 107
- /Thu14/ 17 31
- /Win84/ 7 90
- lokal kontextfreie Assoziation 26
- Luftbilder 107
- Mächtigkeit des Merkmalraumes 69
- mehrdeutige Modelle 47
- mehrfach assoziierende Assoziation 26  
47 52 89
- Mehrfachgenerierung
  - bei der Analyse mit Zuordnungsau-  
tomaten 73
  - durch Schachtelung der Suchschlei-  
fen 188
  - ersetzender Systeme 52
  - mit Ableitungsbäumen 75
  - mit Ableitungsketten 75
  - mit Ersetzungsverfahren 188
  - von Objekten 50
  - von Primitivobjekten 74
- Mehrfachgenerierungskoeffizient 60
- Mehrschwellenverfahren 3 108
- Merkmal 3 25
  - beschreibend 26 37
  - Typ-Merkmal 25
- Merkmalraum
  - Einfluss
    - seiner Mächtigkeit auf Aufwand  
69
- Modellierung 45
- Multispektralklassifikation 3
- Navigation
  - bildgestützte 107
- Obermenge
  - von Instanzenmengen 40
- Objekt
  - des Hintergrundes 44
  - Einteilung für eine Analyse 43
  - Erkennung 43
  - Kontext- 44

- nicht modelliert im Hintergrund 47
- schlecht aufgelöst 47
- Trennung vom Hintergrund 41 43
- Verbund- 44
- verdeckt 47
- Verifikation 107
- Ziel- 43
  
- PANDEMONIUM 6 89
- Parallelanalyse 28
- Parameter 118
- PDL 84 85
- picture description language 85
- Primitivobjekt 8
- Probleme 46
- Produktion 24
- Produktionsnetz 4 25
- Produktionsregel 90
- Produktionssystem 90
  
- Qualität einer Analyse 43
  
- rauschen 115
- rbb-Ableitungsbaum 68
- Rechner
  - mehrere Prozessoren 29
  - mehrere verteilt 29
  - Prozessorbelegung 29
- Reduzierung des Analyseaufwandes 94
- Referenzbilder 111
- Referenzobjekt 107
- regelbasiertes System 90
  
- Schlossbild 111
- Schwierigkeitsgrad 78
- Segmentation 3 91
- Segmente mehrdeutig 2
- semantische Netze 90
- Semi-Thue-System 17
- Sensordaten 47
- Sigma 82
- Sprache 32
- Startsymbol 31 33 34 43
  - verwechselt 47
- strukturelle Musteranalyse 82
- STS 17
- Suchbereich 54 60 63 68 93
  
- syntaktische Bildanalyse 16 82
- Szenario 33 51
  - ersetzend erkannt 35
  - ersetzend generiert 33
  - mit Obermenge generiert 34
  - zuordnend erkannt 36
- Szene 33
  
- Tafel 7
- Teil-von-Relation 4
- Testdaten 43 48
- Thue-Systeme 82
- Tiefpassfilterung 107
- Trainingsdaten 43 47
- Transformationssystem 14
  - abstrakt 15
  - Beispiel für 16
  - für Konfigurationen 17
  - für Zeichenketten 16
- Trennung von Objekt und Hintergrund 41
- TS 17
- Typ-Merkmal 25
  
- Überprüfung
  - mehrfach 51
- Überprüfungsauftrag 25 27 162
  - reduziert 179
- Überprüfungsprogramm 9
  
- Vase 47
- Verarbeitungs-Parameter 118
- Verbesserung der Merkmalschätzung 94
- Verbesserung der Modellierung 94
- Verbund-Objekt 44
- Vereinigung
  - Konfigurationen 51
- Verfahrensentwicklung 53
- verschmolzene Instanzen 47
- verwechselte Startsymbole 47
- Vexierbild 1 47
- vexierende Erzeugung 47
- VISION 90
- vollständig ersetzende Analyse 39 41
- Vorverarbeitung 3 108

- Warteschlange 9
- wissensbasiertes System 91
  
- Zeichenketten 16
- Zeilenjitter 115
- Zeilenversatz 115
- Ziel-Objekt 43
- zuordnen 18 36
  - abschließend 19 80
  - asynchron 20
  - kaskadiert 22
  - mit Ableitungen 25
  - mit Konzepten 24
  - parallel 20
  - verteilt 20
  - vollständig 23
- Zuordnungsautomat 26 162
  - mit Ableitungen 179
  - mit Konzepten 162
  - Zustand 27
- Zuordnungssystem 134
  - mit Ableitungen 25 154
  - mit Instanzen 134
  - mit Konzepten 152
- Zustand 27 162
  - analysierbar 27
- Zustandsübergang 27
  - asynchron 29
  - parallel 29
- Zwischenergebnis 43



## Lebenslauf

21.10.1946

geboren in Oldenburg (Oldenburg) als Sohn des Ingenieurs Karl Lütjen und seiner Ehefrau Irmgard, geb. Fricke

1953-1958

Grundschule in Stuttgart und Wiesloch (Baden-Württemberg)

1958-1966

Gymnasium in Wiesloch, Villingen und Heidelberg (Baden-Württemberg)

1967-1968

Dienst bei der Bundeswehr

1969

Praktikant bei der Firma Stotz-Kontakt, Heidelberg

1969-1974

Studium der Elektrotechnik an der Universität Karlsruhe mit Hauptdiplom-Abschluss

seit 1975

Wissenschaftlicher Mitarbeiter und seit 1987 Abteilungsleiter am Forschungsinstitut für Informationsverarbeitung und Mustererkennung (Dir.: Prof. Dr. H. Kazmierczak), dessen Abteilungen seit Juni 1999 Teil des Forschungsinstitutes für Optik und Mustererkennung (Dir.: Prof. Dr. M. Tacke) sind

von Oktober 1998 bis Mai 1999

Kommissarischer Leiter des Forschungsinstitutes für Informationsverarbeitung und Mustererkennung

seit WS 2000/2001

zusammen mit Prof. Dr. M. Tacke Lehrbeauftragter für die Vorlesung "Zweidimensionale Signale und Systeme" an der Fakultät für Elektrotechnik und Informationstechnik der Universität Karlsruhe