

**Universität Karlsruhe (TH)**  
Fakultät für Informatik  
Institut für Logik, Komplexität und  
Deduktionssysteme

**Verantwortlicher Betreuer: Prof. Dr. P. H. Schmitt**  
**Betreuer: Dr. Reiner Hähnle**

**Ein vervollständigungsbasiertes  
Verfahren zur  
Behandlung von Gleichheit im  
Tableaukalkül mit freien Variablen**

**Diplomarbeit**

**Bernhard Beckert**

Überarbeitete Version

# Inhaltsverzeichnis

---

<b>Vorwort</b>	<b>iv</b>
<b>1 Einführung</b>	<b>1</b>
<b>2 Präliminarien</b>	<b>3</b>
2.1 Algebraische Begriffe . . . . .	3
2.2 Syntax und Semantik der Prädikatenlogik . . . . .	5
2.3 Substitutionen, Ersetzungen in Termen . . . . .	7
<b>3 <math>E</math>-Unifikation</b>	<b>13</b>
3.1 Begriffsbestimmung . . . . .	13
3.2 Die Ableitbarkeitsrelation . . . . .	16
3.3 Vollständige Mengen allgemeinsten Unifikatoren . . . . .	20
3.4 Entscheidbarkeit . . . . .	24
3.5 Berechnung von Äquivalenzklassen . . . . .	26
3.6 Vervollständigungs-basierte $E$ -Unifikation . . . . .	29
3.6.1 Grundlagen . . . . .	29
3.6.2 Reduktionsordnungen . . . . .	32
3.6.3 Konfluenz und Grundkonfluenz . . . . .	34
3.6.4 Der Unfailing Knuth-Bendix-Algorithmus . . . . .	35
3.6.5 Das Verfahren von Gallier et al. . . . .	38
3.6.6 Das Verfahren von Goubault . . . . .	41
<b>4 Vervollständigungs-basierte gemischte <math>E</math>-Unifikation</b>	<b>42</b>
4.1 Einführung . . . . .	42
4.2 Ordnungsbedingungen und Constraints . . . . .	44
4.3 Terme und Reduktionsregeln mit Constraint . . . . .	50
4.4 Transformationsregeln und Vervollständigung . . . . .	56
4.4.1 Ziel der Vervollständigung . . . . .	56
4.4.2 Löschung . . . . .	56
4.4.3 Äquivalenzumformung . . . . .	57
4.4.4 Subsumtion . . . . .	57

4.4.5	Kritisches Paar, Kombination, Vereinfachung . . . . .	58
4.4.6	Faire Vervollständigungsverfahren . . . . .	60
4.5	Berechnung von Normalformen . . . . .	62
4.6	Vermischen von Vervollständigung und Normalisierung . . . . .	64
4.7	Korrektheit . . . . .	68
4.8	Vollständigkeit . . . . .	74
4.9	Schwache Konfluenz der Vervollständigung . . . . .	93
4.10	Lösung simultaner $E$ -Unifikationsprobleme . . . . .	94
4.11	Beziehung zu Conditional Rewriting . . . . .	99
<b>5</b>	<b>Der Tableaurechner mit freien Variablen und Gleichheit</b>	<b>100</b>
5.1	Der Tableaurechner mit freien Variablen . . . . .	100
5.1.1	Einführung . . . . .	100
5.1.2	Syntax und Semantik . . . . .	101
5.1.3	Universelle Formeln . . . . .	106
5.2	Der Tableaurechner mit Gleichheit . . . . .	108
5.2.1	Einführung . . . . .	108
5.2.2	Gleichheitsbehandlung durch zusätzliche Tableauregeln . . . . .	109
5.2.3	Gleichheitsbehandlung durch $E$ -Unifikation . . . . .	112
5.2.4	Sukzessiver Abschluß der Äste . . . . .	117
5.2.5	Übertragung auf andere Versionen des Tableaurechners . . . . .	118
<b>6</b>	<b>Implementierung</b>	<b>119</b>
6.1	Implementierung als Teil des Tableaurechners $\mathcal{I}^A\mathcal{P}$ . . . . .	119
6.2	Überprüfen der Erfüllbarkeit von Constraints . . . . .	120
6.3	Die Ordnung auf den Funktions- und Konstantensymbolen . . . . .	123
6.4	Die verwendete Vervollständigungs- und Normalisierungsprozedur . . . . .	124
6.5	Ein- und Ausgabe . . . . .	125
<b>7</b>	<b>Experimente und Beispiele</b>	<b>129</b>
7.1	Testprobleme . . . . .	130
7.1.1	Ein einfaches rein universelles Problem . . . . .	130
7.1.2	Ein einfaches rein starres Problem . . . . .	131
7.1.3	Systeme mit schnell wachsenden Vervollständigungen . . . . .	131
7.1.4	Ein Beispiel für die Berechnung weiterer Unifikatoren . . . . .	133
7.2	Beispiele für rein universelle Probleme . . . . .	134
7.2.1	Gruppentheorie . . . . .	134

---

7.2.2	Mika-Beispiel . . . . .	135
7.2.3	Transitivitäts-Beispiel . . . . .	136
7.2.4	Weitere Beispiele aus [Hsiang & Mzali, 1988] . . . . .	137
7.3	Aussagen über Lukasiewicz Logiken . . . . .	138
7.4	Pelletiers Probleme . . . . .	139
<b>8</b>	<b>Fazit und Ausblick</b>	<b>142</b>
8.1	Fazit . . . . .	142
8.2	Mögliche Verbesserungen . . . . .	142
	<b>Literaturverzeichnis</b>	<b>144</b>
	<b>Symbolverzeichnis</b>	<b>147</b>

# Vorwort

---

## Thema der Arbeit

Thema dieser Diplomarbeit ist die Erweiterung des Tableakalküls mit freien Variablen und universellen Formeln um die Behandlung von Gleichheit. Dabei war das Ziel nicht nur der theoretische Entwurf eines Verfahrens, sondern es sollte auch — als Teil des tableaubasierten Theorembeweisers  $\mathcal{3TAP}$  — implementiert werden.

Dabei konnte von dem in [Beckert, 1991] entwickelten und als effizient nachgewiesenen Konzept ausgegangen werden, die Anwendung von Gleichungen nur zum Abschluß eines Tableaus einzusetzen und nicht zu seinem Aufbau. Das erlaubt es, die Gleichheitsbehandlung im Tableauekalkül auf ein allgemeineres, im folgenden als „gemischte  $E$ -Unifikation“ bezeichnetes Problem zu reduzieren, dessen Lösung — vor allem mit vervollständigungsbasierten Methoden — auch unabhängig von der Gleichheitsbehandlung im Tableauekalkül von Interesse ist.

Die Aufgabenstellung dieser Arbeit und erste Ergebnisse sind in [Beckert, 1993] zusammengefaßt.

## Aufbau der Arbeit

Durch diese Arbeit ziehen sich zwei rote Fäden: die Lösung gemischter  $E$ -Unifikationsprobleme und die Konstruktion eines effizienten Verfahrens zur Gleichheitsbehandlung im Tableauekalkül. Diese roten Fäden laufen in Kapitel 5 zusammen. Aber auch beim Lesen der Kapitel 1 bis 4 sollte man stets beide Ziele vor Augen haben.

Nach einer kurzen Einführung in Kapitel 1 werden in Kapitel 2 einige grundlegende Definitionen gegeben und Schreibweisen eingeführt. Kapitel 3 enthält eine Beschreibung der gemischten  $E$ -Unifikation, die Unterschiede zu in der Literatur üblichen Definitionen werden aufgezeigt, und die bekannten Verfahren für  $E$ -Unifikation dargestellt. In Kapitel 4 ist das neue vervollständigungsbasierte Verfahren für gemischte  $E$ -Unifikation beschrieben; seine Eigenschaften, Vor- und Nachteile werden diskutiert. Kapitel 5 gibt zunächst eine kurze Einführung in den Tableauekalkül mit freien Variablen und universellen Formeln; anschließend wird beschrieben, wie dieser Kalkül mit Hilfe der gemischten  $E$ -Unifikation um die Behandlung von Gleichheit erweitert werden kann. Die Implementierung des Verfahrens ist in Kapitel 6 beschrieben. In Kapitel 7 sind Experimente mit dieser Implementierung dargestellt, und die Implementierung wird mit anderen Systemen verglichen. Kapitel 8 enthält eine Zusammenfassung der Ergebnisse dieser Arbeit, ein Fazit und einen Ausblick auf mögliche Erweiterungen und Verbesserungen. Die Arbeit schließt mit dem Literaturverzeichnis und einer Liste der verwendeten Symbole und Schreibweisen.

## **Danksagung**

Viele haben mich während der Entstehung dieser Arbeit unterstützt und ermutigt. Ihnen allen möchte ich von Herzen danken. Mein ganz besonderer Dank gilt

- Prof. Peter Schmitt, der der verantwortliche Betreuer dieser Arbeit ist, und mir manch wertvollen Hinweis und Ratschlag gab.
- Dr. Reiner Hähnle, meinem Betreuer, der es mir ermöglicht hat, dieses Thema für meine Diplomarbeit zu wählen. Seine Tür stand mir — und meinen Fragen und Problemen — jederzeit offen; zugleich ließ er mir immer den nötigen Freiraum.
- Wilfred Chen, Jean Goubault, Eric de Kogel, Prof. Paliath Narendran und Dr. Uwe Petermann für Hinweise und interessante Diskussionen.

## **Erklärung**

Ich erkläre hiermit, diese Arbeit selbständig verfaßt und keine anderen als die angegebenen Quellen und Hilfsmittel verwendet zu haben.

Karlsruhe, den 10. Juli 1993

Bernhard Beckert

## **Überarbeitete Version**

Dies ist eine leicht überarbeitete Version der Arbeit, die nicht mit der ersten, „offiziellen“ Version übereinstimmt. Eine Reihe kleinerer Fehler — auf die meisten hat mich mein Betreuer, Reiner Hähnle, aufmerksam gemacht — sind korrigiert. Im wesentlichen ist die Arbeit aber unverändert geblieben.

Karlsruhe, den 31. Juli 1993

Bernhard Beckert

# 1 Einführung

---

*Logicians have but ill defined  
As rational the human kind.  
Logic, they say, belongs to man,  
But let them prove it if they can.*

— OLIVER GOLDSMITH

Eines der Hauptziele des automatischen Beweisens ist die effiziente Behandlung der Prädikatenlogik mit Gleichheit. Aber seit die ersten Versuche auf diesem Gebiet unternommen wurden, ist bekannt, daß es fast unausweichlich zu einem viel zu großen Suchraum führt, die Axiomatisierung der Gleichheit zur Datenbasis hinzuzufügen.<sup>1</sup> Selbst einfachste Theoreme der Prädikatenlogik können nur mit solchen Verfahren bewiesen werden, bei denen die Gleichheitsbehandlung in irgendeiner Form Teil der Inferenzregeln ist. Aber auch dann bleibt die typische Eigenschaft der Gleichheit, viele verschiedene Ableitungen zu ermöglichen, problematisch; und es ist unerlässlich, Methoden zu finden, den entstehenden Suchraum einzuschränken.

Für resolutionsbasierte Theorembeweiser sind solche Verfahren — das wichtigste ist die Paramodulation — schon seit Ende der sechziger Jahre bekannt und finden breiten Einsatz (obwohl auch hier das Problem einer effizienten Kontrolle, die die Ableitung redundanter Formeln verhindert, noch lange nicht gelöst ist).

Genauso lange gibt es auch Methoden zur Gleichheitsbehandlung für Gentzen-artige Kalküle, wie den Tableauekalkül und die Konnektionsmethode [Jeffrey, 1967, Popplestone, 1967]. Diese sind aber wenig angewendet worden und im Vergleich zur Resolution mit Paramodulation ineffizient. In den letzten Jahren jedoch hat die Behandlung von Gleichheit in der Konnektionsmethode [Gallier *et al.*, 1992, Petermann, 1993] und im Tableauekalkül [Reeves, 1987, Fitting, 1990, Beckert & Hähnle, 1992] wieder verstärkt Beachtung gefunden.

Es hat sich dabei gezeigt, daß die Gleichheitsbehandlung auf ein abstrakteres Problem, die  $E$ -Unifikation zurückgeführt werden kann. Allerdings genügt es nicht, das klassische  $E$ -Unifikationsproblem<sup>2</sup>, das im folgenden als universelle  $E$ -Unifikation bezeichnet wird, zu lösen, sondern zur Gleichheitsbehandlung im Tableauekalkül mit *freien Variablen* und in der Konnektionsmethode muß ein als *starre*  $E$ -Unifikation bezeichnetes Problem gelöst werden. Für den Tableauekalkül mit universellen Formeln [Beckert & Hähnle, 1992] genügt jedoch auch das nicht, sondern es muß eine

---

<sup>1</sup> Eine Diskussion der Problematik und eine Übersicht über verschiedene Verfahren enthält [Fitting, 1990].

<sup>2</sup> Zu diesem existiert eine umfangreiche Literatur. Eine Übersicht gibt [Siekman, 1989].

Kombination beider Versionen der  $E$ -Unifikation, die im folgenden als gemischt  $E$ -Unifikation bezeichnet wird, betrachtet werden.

Die Symmetrie der Gleichheit ist ein wesentlicher Grund für die Größe des Suchraums. Er kann deutlich eingeschränkt werden, wenn man die Gleichungen orientiert. In ihrem berühmten Artikel haben D. Knuth und P. Bendix [1970] gezeigt, daß es möglich ist, die Menge der entstehenden *Regeln* so zu vervollständigen, daß auch mit ihrer Hilfe alle jeweils gültigen Gleichungen bewiesen werden können. Weiterentwicklungen dieser Methode haben sich überall da, wo Gleichheit eine Rolle spielt, — nicht nur beim automatischen Beweisen — als die erfolgreichsten erwiesen.

Es ist darum wünschenswert, auch für die Lösung gemischter  $E$ -Unifikationsprobleme ein vervollständigungsbasiertes Verfahren zur Verfügung zu haben — und damit für die Gleichheitsbehandlung im Tableauekalkül mit freien Variablen und universellen Formeln. Ein solches Verfahren wird in Kapitel 4 vorgestellt.



# 2 Präliminarien

---

*I hate definitions.*

— BENJAMIN DISRAELI, Vivian Grey (1826)

## 2.1 Algebraische Begriffe

In diesem Abschnitt sind die wichtigsten der im weiteren verwendeten algebraischen Strukturen definiert. Wie diese werden auch alle anderen algebraischen Begriffe (wie „Äquivalenzrelation“, „Komposition von Relationen“ und „Verband“) in der üblichen Weise verwendet.<sup>1</sup>

**Definition 2.1.1** (Ordnungsrelation, wohlfundierte Relation)

Eine binäre Relation  $\rightarrow$  auf einer Menge  $B$  heißt

- (partielle) Ordnungsrelation (auf  $B$ ), wenn sie reflexiv, transitiv und antisymmetrisch ist,
- strikte (partielle) Ordnungsrelation (auf  $B$ ), wenn sie irreflexiv und transitiv ist,
- total, wenn für  $b_1, b_2 \in B$ ,  $b_1 \neq b_2$  stets  $b_1 \rightarrow b_2$  oder  $b_2 \rightarrow b_1$  gilt,
- wohlfundiert (noethersch, terminierend), wenn es keine unendlich absteigende Folge  $(b_n)_{n>0} \subset B$  gibt mit

$$b_1 \rightarrow b_2 \rightarrow b_3 \rightarrow \dots$$

**Definition 2.1.2** (Schreibweise für die Abschlüsse einer Relation)

Sei  $\rightarrow$  eine binäre Relation. Dann bezeichnet  $\xrightarrow{+}$  ihren transitiven,  $\xrightarrow{*}$  ihren reflexiven und transitiven,  $\leftrightarrow$  ihren symmetrischen,  $\overset{*}{\leftrightarrow}$  ihren transitiven, reflexiven und symmetrischen Abschluß und  $\leftarrow$  ihre Inverse.

**Definition 2.1.3** (Church–Rosser–Eigenschaft, [lokale] Konfluenz)

Sei  $\rightarrow \subset B \times B$  eine binäre Relation.

$\rightarrow$  hat die Church–Rosser–Eigenschaft, wenn für beliebige  $b_1, b_2 \in B$  aus

$$b_1 \overset{*}{\leftrightarrow} b_2$$

---

<sup>1</sup> Im Zweifel seien sie wie in [Lipson, 1981] definiert.

folgt, daß es ein  $b' \in B$  gibt mit

$$b_1 \xrightarrow{*} b' \quad \text{und} \quad b_2 \xrightarrow{*} b' .$$

$\rightarrow$  heißt konfluent, wenn für beliebige  $b, b_1, b_2 \in B$  aus

$$b \xrightarrow{*} b_1 \quad \text{und} \quad b \xrightarrow{*} b_2$$

folgt, daß es ein  $b' \in B$  gibt mit

$$b_1 \xrightarrow{*} b' \quad \text{und} \quad b_2 \xrightarrow{*} b' .$$

Folgt das gleiche zumindest aus

$$b \rightarrow b_1 \quad \text{und} \quad b \rightarrow b_2 ,$$

so heißt  $\rightarrow$  lokal konfluent.

Diese Definition der Begriffe „Church–Rosser–Eigenschaft“ und „konfluent“ entspricht ihrer intuitiven Bedeutung. Kürzer und eleganter lassen sie sich definieren durch:

- Die Relation  $\rightarrow$  hat die Church–Rosser–Eigenschaft, wenn  $\leftrightarrow^*$  Teilrelation von  $\xrightarrow{*} \circ \xleftarrow{*}$  ist.
- Die Relation  $\rightarrow$  ist konfluent, wenn  $\xleftarrow{*} \circ \xrightarrow{*}$  Teilrelation von  $\xrightarrow{*} \circ \xleftarrow{*}$  ist.
- Die Relation  $\rightarrow$  ist lokal konfluent, wenn  $\leftarrow \circ \rightarrow$  Teilrelation von  $\xrightarrow{*} \circ \xleftarrow{*}$  ist.

Dabei bezeichnet  $\circ$  die Komposition von Relationen.

Die folgenden beiden Lemmata enthalten die wichtigsten Eigenschaften konfluenter Relationen.<sup>2</sup>

#### Lemma 2.1.4

*Eine binäre Relation hat genau dann die Church–Rosser–Eigenschaft, wenn sie konfluent ist.*

Die Äquivalenz der Begriffe Church–Rosser–Eigenschaft und Konfluenz ist wesentlich für die vervollständigungsbasierte Gleichheitsbehandlung (siehe Abschnitt 3.6). Sie ist aber auch der Grund dafür, daß im folgenden (fast) nur noch von Konfluenz die Rede sein wird.

#### Lemma 2.1.5

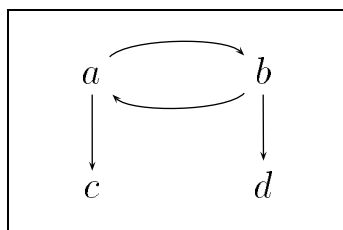
*Ist eine binäre Relation lokal konfluent und wohlfundiert, dann ist sie konfluent.*

#### Beispiel 2.1.6

Abbildung 2.1 zeigt ein Beispiel für eine Relation, die zwar — wie man leicht nachvollzieht — lokal konfluent ist, nicht aber konfluent: Es gilt  $a \xrightarrow{*} c$  und  $a \xrightarrow{*} d$ , aber es gibt keinen Term  $t$  mit  $c \xrightarrow{*} t$  und  $d \xrightarrow{*} t$ .

Da die Relation einen Zyklus enthält, ist sie nicht wohlfundiert, und Lemma 2.1.5 auf sie nicht anwendbar.  $\square$

<sup>2</sup> Eine ausführliche Darstellung der Eigenschaften konfluenter Relationen und Angaben zum Beweis der Lemmata findet sich in [Dershowitz & Jouannaud, 1990].



**Abbildung 2.1:** Relation, die zwar lokal konfluent nicht aber konfluent ist (Beispiel 2.1.6).

## 2.2 Syntax und Semantik der Prädikatenlogik

Eine ausführliche Darstellung der Syntax und der Semantik der Prädikatenlogik, insbesondere auch der mit Gleichheit, findet sich in [Fitting, 1990]. Wie dort seien mit den logischen Operatoren  $\neg$ ,  $\wedge$ ,  $\vee$ ,  $\supset$ , den Quantoren  $\forall$  und  $\exists$  und einer Menge  $\mathcal{V}$  von Variablen über einer Signatur  $\mathcal{S}$  — bestehend aus einer Menge  $\mathcal{P}$  von Prädikaten-,  $\mathcal{F}$  von Funktions- und  $\mathcal{K}$  von Konstantensymbolen — die Mengen der Terme **Term** und der Formeln **Form** der Prädikatenlogik (mit und ohne Gleichheit) in der üblichen Weise definiert. Das gleiche gelte für die aus einer Grundmenge  $\mathcal{D}$  und einer Interpretation  $\mathcal{I}$  bestehenden Modelle  $\mathcal{M}$  der Prädikatenlogik.

Auf syntaktischer Ebene wird die Gleichheit durch das Prädikatensymbol  $\approx$  dargestellt, um Verwechslungen mit der Meta-Gleichheit  $=$  zu vermeiden.<sup>3</sup>

**Definition 2.2.1** (Mengen  $\text{Var}(G)$ ,  $\text{Free}(G)$ ,  $\text{Bound}(G)$ )

Sei  $G$  ein Term, eine Formel, eine Menge von Formeln oder ein anderes syntaktisches Objekt. Dann bezeichne

- $\text{Var}(G)$  die Menge der in  $G$  auftretenden Variablen,
- $\text{Free}(G)$  die Menge der in  $G$  frei auftretenden Variablen,
- $\text{Bound}(G)$  die Menge der in  $G$  gebunden auftretenden Variablen.

Im folgenden wird o. B. d. A. davon ausgegangen, daß stets

$$\text{Free}(G) \cap \text{Bound}(G) = \emptyset ,$$

was ggf. durch Umbenennung der gebundenen Variablen erreicht wird.

**Definition 2.2.2** (Normales Modell)

Ein Modell  $\mathcal{M} = \langle \mathcal{D}, \mathcal{I} \rangle$  heie *normal*, wenn die Interpretation des Prädikatensymbols  $\mathcal{I}(\approx)$  die Gleichheitsrelation auf der Grundmenge  $\mathcal{D}$  ist.

**Definition 2.2.3** (Kanonisches Modell)

Ein normales Modell  $\mathcal{M} = \langle \mathcal{D}, \mathcal{I} \rangle$  heie *kanonisch*, wenn für jedes  $d \in \mathcal{D}$  ein Term  $t$  existiert, so daß  $\mathcal{I}(t) = d$  ist.

<sup>3</sup>  $t = s$  bedeutet also, daß die Terme  $t$  und  $s$  identisch (d. h. syntaktisch gleich) sind, während  $t \approx s$  eine Formel der Objektebene ist.

**Definition 2.2.4** (Erfüllbarkeit, Folgerbarkeit, Tautologie)

Ist eine Formel  $F \in \mathbf{Form}$  wahr in einem Modell  $\mathcal{M}$ , so heie  $F$  erfllbar und  $\mathcal{M}$  ein Modell von  $F$ .  $\mathcal{M}$  heie Modell einer Formelmenge  $\Phi \subset \mathbf{Form}$  und die Menge  $\Phi$  erfllbar, wenn  $\mathcal{M}$  Modell aller Formeln in  $\Phi$  ist.

Ist  $F$  wahr in allen [normalen] Modellen einer Formelmenge  $\Phi$ , so heie  $F$  logisch folgerbar aus  $\Phi$  [in der Prdikatenlogik mit Gleichheit].

Ist  $F$  wahr in allen [normalen] Modellen, heie  $F$  Tautologie [der Prdikatenlogik mit Gleichheit].

Schreibweise fr

- $\mathcal{M}$  ein Modell von  $F$ :  $\mathcal{M} \models F$ ,
- $\mathcal{M}$  ein Modell von  $F$ ,  $\mathcal{M}$  zudem normal:  $\mathcal{M} \models^{\cong} F$ ,
- $F$  folgerbar aus  $\Phi$ :  $\Phi \models F$ ,
- $F$  folgerbar aus  $\Phi$  in der Prdikatenlogik mit Gleichheit:  $\Phi \models^{\cong} F$ ,
- $F$  eine Tautologie:  $\models F$ ,
- $F$  eine Tautologie der Prdikatenlogik mit Gleichheit:  $\models^{\cong} F$ .

Treten in  $\Phi$  und  $F$  freie Variablen auf, so knnen bei der Fragestellung, ob  $\Phi \models F$  (bzw.  $\Phi \models^{\cong} F$ ) gilt, diese freien Variablen als implizit all-quantifiziert angesehen und zum Beweis beliebig instantiiert werden. Es wird aber im weiteren hufig von Interesse sein, ob  $F$  auch dann noch aus  $\Phi$  folgerbar ist, wenn die freien Variablen nicht instantiiert werden drfen, sondern „starr“ sind.<sup>4</sup> Es ist darum sinnvoll, eine Schreibweise fr diesen Sachverhalt zu definieren:

**Definition 2.2.5** (Starre Folgerbarkeit)

Seien  $F$  eine Formel und  $\Phi$  eine Formelmenge ber der Signatur  $\mathcal{S}$ .  $\mathcal{S}$  und die Menge der Variablen<sup>5</sup>  $\mathcal{V}$  werden in der Weise verndert, da die in  $F$  und  $\Phi$  frei auftretenden Variablen zu Konstanten werden, d. h., es sei

$$\mathcal{V}' = \mathcal{V} \setminus (\text{Free}(F) \cup \text{Free}(\Phi))$$

die neue Menge der Variablen, und ist  $\mathcal{K}$  die Menge der Konstanten von  $\mathcal{S}$ , dann sei

$$\mathcal{K}' = \mathcal{K} \cup \text{Free}(F) \cup \text{Free}(\Phi)$$

die Menge der Konstanten der neuen Signatur  $\mathcal{S}'$ ; ansonsten stimme  $\mathcal{S}'$  mit  $\mathcal{S}$  berein.

Es gelte

$$\Phi \models^{\cong} F$$

<sup>4</sup> Dieser Folgerbarkeitsbegriff wurde zuerst in [Gallier *et al.*, 1987] definiert (allerdings eingeschrnkt auf Gleichungen): “. . .  $E \models s \approx t$  where the Variables in the problem are held rigid.” Die Bezeichnung „starre Folgerbarkeit“ ist an diese Formulierung angelehnt.

<sup>5</sup> Die Menge der Variablen ist nicht Teil der Signatur.

in der Prädikatenlogik über der Signatur  $\mathcal{S}$  und der Menge der Variablen  $\mathcal{V}$  genau dann, wenn  $\Phi \models F$  in der Prädikatenlogik über der Signatur  $\mathcal{S}'$  und der Menge der Variablen  $\mathcal{V}'$  gilt, und es gelte

$$\Phi \models F$$

in der Prädikatenlogik mit Gleichheit über der Signatur  $\mathcal{S}$  und der Menge der Variablen  $\mathcal{V}$  genau dann, wenn  $\Phi \models^= F$  in der Prädikatenlogik mit Gleichheit über der Signatur  $\mathcal{S}'$  und der Menge der Variablen  $\mathcal{V}'$  gilt.

$\models$  und  $\models^=$  sind nur als Relationen zwischen Formelmengen und Formeln definiert und nicht wie  $\models$  auch als Relationen zwischen Modellen und Formeln.

Wie man leicht nachvollzieht, besteht folgende Beziehung zwischen  $\models$  und  $\models^=$  (bzw.  $\models^{\approx}$  und  $\models^{\approx=}$ ):

### Lemma 2.2.6

Seien  $F$  eine Formel und  $\Phi$  eine Formelmenge.  $F_{\mathcal{V}}$  und  $\Phi_{\mathcal{V}}$  bezeichnen deren All-Abschluß. Dann gilt:

$$\Phi \models F \quad \text{gdw.} \quad \Phi_{\mathcal{V}} \models F_{\mathcal{V}} ,$$

und

$$\Phi \models^= F \quad \text{gdw.} \quad \Phi_{\mathcal{V}} \models^= F_{\mathcal{V}} .$$

Wie der folgende Satz zeigt, bilden die kanonische Modelle das Analogon zu den Herbrand-Modellen der Prädikatenlogik ohne Gleichheit.

### Satz 2.2.7

Gibt es ein normales Modell, in dem die Menge geschlossener Formeln  $\Phi$  erfüllt ist, dann es gibt auch ein kanonisches Modell, in dem  $\Phi$  erfüllt ist.

#### Beweis:

Beweis zu Theorem 8.6.3 in [Fitting, 1990]. ■

Aus diesem Satz folgt, daß es genügt, sich bei der Betrachtung der Erfüllbarkeit oder Allgemeingültigkeit einer Formel auf kanonische Modelle zu beschränken.

## 2.3 Substitutionen, Ersetzungen in Termen

Da Substitutionen und Ersetzungen in Termen in den folgenden Kapiteln eine wichtige Rolle spielen, ist ihrer Definition und den Formalismen für ihre Handhabung ein eigener Abschnitt gewidmet.

**Definition 2.3.1** (Substitution, Definitions- und Wertebereich, Einschränkung einer Substitution, Grundsubstitution, Komposition von Substitutionen)

**Subst** sei die Menge der Substitutionen, d.h. der Abbildungen  $\sigma : \mathcal{V} \rightarrow \mathbf{Term}$  der Variablen auf die Terme.

Es seien

$$\text{Dom}(\sigma) = \{x \in \mathcal{V} : \sigma(x) \neq x\}$$

und

$$\text{Range}(\sigma) = \{\sigma(x) : x \in \text{Dom}(\sigma)\} .$$

Ist  $\text{Dom}(\sigma) = \{x_1, \dots, x_n\}$  endlich, so kann  $\sigma$  auch in der Form

$$\sigma = \{x_1/t_1, \dots, x_n/t_n\}$$

dargestellt werden. Dabei bezeichnet  $t_i$  ( $1 \leq i \leq n$ ) das Bild der Variablen  $x_i$  unter  $\sigma$ . Eine Substitution  $\sigma$ , so daß  $\text{Dom}(\sigma) = \mathcal{V}$ , und alle Terme in  $\text{Range}(\sigma)$  Grundterme sind, heie Grundsubstitution.

Die Einschränkung  $\sigma|_V$  einer Substitution  $\sigma$  auf eine Menge  $V \subset \mathcal{V}$  von Variablen ist die Substitution, die fur  $x \in \mathcal{V}$  definiert ist durch

$$\sigma|_V(x) = \begin{cases} \sigma(x) & \text{falls } x \in V \\ x & \text{sonst} \end{cases}$$

Die Komposition  $\sigma \circ \tau$  zweier Substitutionen  $\sigma$  und  $\tau$  ist die Substitution, die fur  $x \in \mathcal{V}$  definiert ist durch

$$(\sigma \circ \tau)(x) = \sigma(\tau(x)) .$$

Im folgenden werden auch solche Substitutionen als Grundsubstitutionen bezeichnet, die zwar nicht alle, aber doch alle in einem betrachteten Problem auftretenden Variablen mit Grundtermen belegen.

Die leere Substitution wird mit  $id$  bezeichnet. Fur sie gilt  $\text{Dom}(id) = \emptyset$ .

**Definition 2.3.2** (Idempotente Substitution)

Eine Substitution  $\sigma$  heie idempotent, wenn  $\sigma = \sigma \circ \sigma$ .

**Subst\***  $\subset$  **Subst** sei die Menge der idempotenten Substitutionen.

Bei der Anwendung einer idempotenten Substitution mussen die Einzelsubstitutionen nicht parallel sondern konnen auch nacheinander ausgefuhrt werden.

**Beispiel 2.3.3**

Idempotent sind beispielsweise die Substitutionen

- $\{x/y, z/y\}$
- $\{x/a, y/f(b)\}$ .

Nicht idempotent sind

- $\{x/y, y/a\}$ .
- $\{x/f(x)\}$ .

□

**Definition 2.3.4** (Anwendung einer Substitution, Instanz)

Die Anwendung  $t\sigma$  einer Substitution  $\sigma$  auf einen Term  $t$  erhält man, indem man alle in  $t$  frei auftretenden Variablen  $x$  (simultan) durch  $\sigma(x)$  ersetzt.

$t\sigma$  heißt eine Instanz von  $t$ .

Die Anwendung einer Substitution wird entsprechend auf aus Termen aufgebauten Objekten definiert.<sup>6</sup>

Würden durch die Ersetzung zusätzliche Variablenbindungen entstehen, bleibt die Anwendung der Substitution undefiniert.

**Beispiel 2.3.5**

Sei  $\sigma_1 = \{x/f(x,y), y/h(a), z/a\}$ . Dann ist

$$h(g(x), y)\sigma_1 = h(g(f(x,y)), h(a)).$$

Sei  $\sigma_2 = \{x/a\}$ . Dann ist, da  $x$  in der Formel gebunden ist, und also nicht ersetzt wird,

$$((\forall x)p(x,y))\sigma_2 = (\forall x)p(x,y) .$$

Die Anwendung der Substitution  $\{y/x\}$  auf die gleiche Formel ist unzulässig (bzw. undefiniert), weil eine zusätzliche Variablenbindung auftreten würde.  $\square$

**Definition 2.3.6** (Spezialisierungsrelation  $\leq$ )

$\leq$  sei die Spezialisierungsrelationen auf den Substitutionen, d. h.,

$$\sigma \leq \tau$$

genau dann, wenn  $\tau$  eine Spezialisierung von  $\sigma$  ist ( $\sigma$  allgemeiner ist als  $\tau$ ), wenn es also eine Substitution  $\sigma'$  gibt mit

$$\tau = \sigma' \circ \sigma .$$

**Beispiel 2.3.7**

Es gilt  $\{x/f(y)\} \leq \{x/f(a), y/a\}$ , aber  $\{x/f(y)\} \not\leq \{x/f(a)\}$ .  $\square$

Die leere Substitution  $id$  ist die allgemeinste Substitution, es gilt also  $id \leq \sigma$  für alle  $\sigma \in \mathbf{Subst}$ .

Die Relation  $\leq$  ist eine Halbordnung. Durch Übergang zu Äquivalenzklassen kann man sie zur Ordnungsrelation machen:

**Definition 2.3.8** (Äquivalenzrelation von Substitutionen)

Die Äquivalenzrelation  $\sim \subset \mathbf{Subst} \times \mathbf{Subst}$  sei gegeben durch

$$\sigma \sim \tau$$

genau dann, wenn

$$\sigma \leq \tau \quad \text{und} \quad \tau \leq \sigma .$$

<sup>6</sup> Dabei ist jedoch stets zu beachten, daß nur frei auftretende Variablen ersetzt werden.

Eine äquivalente Definition der Relation  $\sim$  ist, daß  $\sigma \sim \tau$  genau dann gelte, wenn  $\sigma$  aus  $\tau$  durch Variablenumbenennung hervorgeht.

**Beispiel 2.3.9**

Es gilt  $\{x/y, z/y\} \sim \{y/x, z/x\}$ , aber  $\{x/a\} \not\sim \{y/a\}$ . □

Überträgt man die Ordnungsrelation  $\leq$  in naheliegender Weise auf die Äquivalenzklassen bezüglich  $\sim$ , und nimmt man ein zusätzliches Element  $\top$  hinzu, so wird die Menge der Substitutionen mit endlichem Definitionsbereich modulo  $\sim$  zum Verband:

**Definition 2.3.10** (Substitutionen modulo  $\sim$ )

$[\sigma]$  bezeichne die Äquivalenzklasse einer Substitution  $\sigma \in \mathbf{Subst}$  bezüglich der Relation  $\sim$  (Def. 2.3.8).

Auf der Menge

$$\mathbf{Subst}_{\sim} = \{[\sigma] : \sigma \in \mathbf{Subst}, \text{Dom}(\sigma) \text{ endlich}\} \cup \{\top\}$$

sei die Ordnungsrelation  $<_{\sim} \in \mathbf{Subst}_{\sim} \times \mathbf{Subst}_{\sim}$  definiert durch

- $[\sigma] <_{\sim} [\tau]$  gdw.  $\sigma \leq \tau$ ,  $[\sigma] \neq [\tau]$ ,
- $\top \not<_{\sim} [\sigma]$ ,  $[\sigma] <_{\sim} \top$

für alle  $\sigma, \tau \in \mathbf{Subst}$  mit  $\text{Dom}(\sigma)$  und  $\text{Dom}(\tau)$  endlich.

**Satz 2.3.11**

Die Menge  $\mathbf{Subst}_{\sim}$  ist mit der auf ihr definierten Ordnungsrelation  $<_{\sim}$  ein Vereinigungshalbverband.<sup>7</sup>

**Beweis:**

Substitutionen  $\sigma$  und  $\tau$ , die überhaupt eine gemeinsame Spezialisierung besitzen, besitzen eine bis auf Variablenumbenennung eindeutig bestimmte allgemeinste gemeinsame Spezialisierung [Martelli & Montanari, 1982],<sup>8</sup> und also besitzen in diesem Fall  $[\sigma]$  und  $[\tau]$  ein Supremum. Besitzen  $\sigma$  und  $\tau$  keine Spezialisierung, so ist  $\top$  die einzige obere Schranke von  $[\sigma]$  und  $[\tau]$  und also auch ein Supremum. ■

Dieser Satz ist von Interesse, weil er die Handhabung von Substitutionen und die Notation sehr vereinfacht. Insbesondere wird das Supremum  $[\sigma] \sqcup [\tau]$  im folgenden eine wichtige Rolle spielen, denn es bezeichnet die allgemeinste gemeinsame Spezialisierung von  $\sigma$  und  $\tau$  (modulo Variablenumbenennung) — sofern eine solche existiert. Besitzen  $\sigma$  und  $\tau$  keine gemeinsame Spezialisierung, so ist  $[\sigma] \sqcup [\tau] = \top$ .

Aus der Tatsache, daß  $\mathbf{Subst}_{\sim}$  ein Verband ist, lassen sich sofort wichtige Eigenschaften des Operators  $\sqcup$  ableiten, so etwa seine Kommutativität, Assoziativität

<sup>7</sup>  $\mathbf{Subst}_{\sim}$  ist sogar ein Verband (mit dem kleinsten Element  $[id]$ ); und die Ordnung  $<_{\sim}$  ist wohlfundiert. Das ist aber nicht ganz einfach zu beweisen und spielt für alles weitere keine Rolle.

<sup>8</sup> Nämlich den allgemeinsten Unifikator der als simultanes Unifikationsproblem aufgefaßten Vereinigung der Substitutionen.



und Idempotenz und, was besonders wichtig ist, daß  $[\sigma] \sqcup [\tau]$  stets eindeutig bestimmt ist.

Die folgende Definition dient nur der Vereinfachung der Notation. Sie ermöglicht es, die allgemeinste gemeinsame Spezialisierung von Substitutionen zu bilden, ohne explizit den Umweg über die Äquivalenzklassen zu nehmen:

**Definition 2.3.12** (Vereinigung von Substitutionen)

Seien  $\sigma, \tau$  Substitutionen. Dann heie

$$\sigma \sqcup \tau = \begin{cases} \text{ein Representant aus } [\sigma] \sqcup [\tau] & \text{falls } [\sigma] \sqcup [\tau] \neq \top \\ \top & \text{falls } [\sigma] \sqcup [\tau] = \top \end{cases}$$

die Vereinigung von  $\sigma$  und  $\tau$ .

Dabei sei eine idempotente Substitution aus  $[\sigma] \sqcup [\tau]$  gewhlt, falls eine solche existiert.<sup>9</sup>

**Beispiel 2.3.13**

In Tabelle 2.1 sind einige Beispiele fur die Vereinigung von Substitutionen angegeben. □

$\sigma$	$\tau$	$\sigma \sqcup \tau$
$\{x/a\}$	$\{y/b\}$	$\{x/a, y/b\}$
$\{x/a\}$	$\{x/a\}$	$\{x/a\}$
$\{x/f(y)\}$	$\{y/a\}$	$\{x/f(a), y/a\}$
$\{x/f(y)\}$	$\{x/f(a)\}$	$\{x/f(a), y/a\}$
$\{x/a\}$	$\{x/b\}$	$\top$
$\{x/f(y)\}$	$\{x/f(a), y/b\}$	$\top$

**Tabelle 2.1:** Beispiele fur die Vereinigung von Substitutionen.

**Definition 2.3.14** (Syntaktische Unifikation)

Terme  $s$  und  $t$  heien syntaktisch<sup>10</sup> unifizierbar, wenn es eine Substitution  $\mu$  mit  $s\mu = t\mu$  gibt. In diesem Fall heit  $\mu$  ein (syntaktischer) Unifikator von  $s$  und  $t$ .

Gilt zudem fur jeden syntaktischen Unifikator  $\nu$  von  $s$  und  $t$

$$\mu \leq \nu ,$$

dann heit  $\mu$  ein allgemeinsten (syntaktische) Unifikator (syntaktischer MGU) von  $s$  und  $t$ .

<sup>9</sup> Das ist insbesondere immer dann der Fall, wenn  $\sigma$  und  $\tau$  idempotent sind, und eine Vereinigung besitzen. Abgesehen von der Bedingung, da eine idempotente Substitution zu wahlen ist, kann ein beliebiger aber eindeutig bestimmter Representant gewhlt werden. Die einzige Anforderung an eine Implementierung ist, da die Auswahl deterministisch erfolgt.

<sup>10</sup> Der Zusatz „syntaktisch“ wird verwendet, um die gewohnliche Unifikation von der  $E$ -Unifikation zu unterscheiden.

Bekanntlich (beispielweise [Gallier *et al.*, 1992]) gilt das folgende Lemma:

**Lemma 2.3.15**

*Unifizierbare Terme  $s$  und  $t$  besitzen stets einen idempotenten allgemeinsten Unifikator.*

**Definition 2.3.16** (Position, Ersetzung an einer Position)

*Eine Position (in einem Term) ist eine endliche Folge natürlicher Zahlen  $\langle p_1, \dots, p_n \rangle$  ( $n \geq 0$ ). Die leere Position wird mit  $\langle \rangle$  bezeichnet.*

*Sei  $t$  ein Term und  $p = \langle p_1, \dots, p_n \rangle$  eine Position. Dann bezeichnet  $t|_p$  einen Unterterm von  $t$  nach folgender Vorschrift:*

$$t|_p = \begin{cases} t & \text{wenn } p = \langle \rangle \\ (t_{p_1})|_{\langle p_2, \dots, p_n \rangle} & \text{wenn } t = f(t_1, \dots, t_k) \text{ und } p_1 \leq k \\ \text{undefiniert} & \text{sonst} \end{cases}$$

*Ist  $t|_p$  definiert, und  $p' = \langle p_1, \dots, p_m \rangle$  ( $m \leq n$ ) ein Anfangsstück von  $p$ , dann heie  $p$  eine Position „im“ Unterterm  $r = t|_{p'}$  von  $t$ .*

*Insbesondere heie  $p$  eine Position in  $t$ , wenn  $t|_p$  definiert ist. In diesem Fall bezeichnet  $t[p/s]$  denjenigen Term,<sup>11</sup> der dadurch entsteht, da  $t|_p$  in  $t$  (an der einen durch  $p$  bezeichneten Stelle) durch den Term  $s$  ersetzt wird.*

*$p \circ p'$  bezeichnet die Verkettung der Positionen  $p$  und  $p'$ , d. h., falls  $p' = \langle p'_1, \dots, p'_m \rangle$ , dann ist  $p \circ p' = \langle p_1, \dots, p_n, p'_1, \dots, p'_m \rangle$ .*

**Beispiel 2.3.17**

Sei  $t = f(a, g(c, d))$ . Dann ist  $t|_{\langle \rangle} = f(a, g(c, d))$ ,  $t|_{\langle 2 \rangle} = g(c, d)$ ,  $t|_{\langle 2, 1 \rangle} = c$ ,  $t|_{\langle 2, 1, 1 \rangle}$  undefiniert. □

**Definition 2.3.18** (Die Relation  $\triangleright$ )

*Fr Terme  $s$  und  $t$  gelte*

$$s \triangleright t$$

*genau dann, wenn*

1. *ein Unterterm von  $s$  Instanz von  $t$  ist, d. h., wenn es eine Position  $p$  in  $s$  und eine Substitution  $\sigma$  gibt, so da  $s|_p = t\sigma$ ,*
2. *das Umgekehrte aber nicht gilt.*

**Beispiel 2.3.19**

Es gilt  $f(a) \triangleright x$  und  $f(a) \triangleright f(x)$ , aber *nicht*  $a \triangleright a$  und *nicht*  $f(x) \triangleright f(y)$ . □

<sup>11</sup> Im folgenden wird auch  $t[s]$  als Kurzschreibweise fr  $t[p/s]$  verwendet, wenn aus dem Kontext hervorgeht, welche Position gemeint ist.

# 3 E-Unifikation

---

```
P = NP = _6568
      yes
— QUINTUS PROLOG, VERSION 3.1.3
auf die Anfrage
| ?- P = NP
```

## 3.1 Begriffsbestimmung

Eine gute und ausführliche Übersicht über die verschiedenen Aspekte der klassischen  $E$ -Unifikation (die im folgenden als rein universelle  $E$ -Unifikation bezeichnet wird), über Probleme und Lösungsmöglichkeiten gibt [Siekmann, 1989].<sup>1</sup>

Folgendermaßen beschreibt J. Siekman das Problem „Unifikation“ in seiner allgemeinen Form:

Gegeben seien zwei Terme  $s$  und  $t$ , die in einer festgelegten Weise Strukturen darstellen und freie Variablen enthalten. Man sagt,  $s$  und  $t$  seien unifizierbar, genau dann, wenn es Substitutionen (d. h. Terme, die die freien Variablen in  $s$  und  $t$  ersetzen) gibt, so daß die beiden Terme in einem bestimmten wohldefinierten Sinne gleich werden.

Bei der  $E$ -Unifikation ist die dargestellte Struktur durch eine Menge  $E$  prädikatenlogischer Gleichungen festgelegt; und Terme sind gleich, wenn ihre Gleichheit aus  $E$  logisch folgerbar ist. Formal läßt sich das  $E$ -Unifikationsproblem folgendermaßen fassen:

**Definition 3.1.1** (Gleichung)

*Eine prädikatenlogische Formel der Form*

$$(\forall x_1) \dots (\forall x_n)(l \approx r) ,$$

*die all-quantifiziert sein kann bezüglich einiger oder aller der Variablen, die sie enthält, heie Gleichung.*

*Im folgenden wird auch die Kurzschreibweise*

$$(\forall \bar{x})(l \approx r)$$

*verwendet.*

---

<sup>1</sup> [Siekmann, 1989] enthlt auch eine ausfhrliche Bibliographie. Dieser und andere Artikel aus dem *Journal of Symbolic Computation* zu verschiedenen Bereichen der Unifikationstheorie sind in [Kirchner, 1990] zusammengefat.

**Definition 3.1.2** (Gemischte  $E$ -Unifikation)

Ein gemischtes  $E$ -Unifikationsproblem  $\langle E, s, t \rangle$  besteht aus einer endlichen<sup>2</sup> Menge  $E$  von Gleichungen und zwei Termen  $s$  und  $t$ .<sup>3</sup>

Eine Substitution  $\sigma$  heißt Lösung des Problems (ein Unifikator), wenn

$$E\sigma \models (s\sigma \approx t\sigma) .$$

Ist  $\sigma$  eine Grundsubstitution, heie sie ein Grund-Unifikator.

Diese Formulierung des Problems unterscheidet sich in einigen wesentlichen Punkten von der in der — sehr umfangreichen — Literatur zur  $E$ -Unifikation blichen.<sup>4</sup>

1. Dort werden die Gleichungen in  $E$  nicht mit Quantoren versehen; statt dessen werden alle Variablen in  $E$  als implizit all-quantifiziert aufgefat.
2. Dementsprechend wird die „normale“ logische Folgerbarkeit  $\models$  verwendet statt der starren Folgerbarkeit  $\models$ .
3. Die Substitution  $\sigma$  wird, anders als in Definition 3.1.2, nicht auf die Menge  $E$  erstreckt, sondern nur auf die Terme  $s$  und  $t$ .

Der Vorteil der — zugegebenermaen komplizierteren — Formulierung des Problems in Definition 3.1.2 ist, da sie das  $E$ -Unifikationsproblem verallgemeinert. Es hat sich erwiesen, da neben der blichen, in verschiedenen Gebieten eingesetzten  $E$ -Unifikation auch ein etwas anderer Unifikationsbegriff von Interesse ist. Um nmlich Gentzen-artige Kalkle, wie den Tableaunkalkl und die Konnektionsmethode, um die Behandlung von Gleichheit zu erweitern, bentigt man auch solche Unifikatoren, die

1. Terme bezglich der starren Folgerbarkeit gleich machen, d. h., es ermglichen, deren Gleichheit zu beweisen, ohne freie Variablen zu instantiiieren, und
2. Terme unter der Voraussetzung gleich machen, da der Unifikator auch auf die Menge der Gleichungen angewendet wird.

Der Unifikationsbegriff aus Definition 3.1.2 stellt nun eine Verknpfung dieser, von Gallier et al. [1992] als „starre  $E$ -Unifikation“<sup>5</sup> bezeichneten, und der in der Literatur blichen nicht-starren Version der  $E$ -Unifikation dar. Beide — sie sind im folgenden „rein starre“ und „rein universelle“  $E$ -Unifikation genannt — knnen als Spezialflle der gemischten  $E$ -Unifikation angesehen werden:

**Definition 3.1.3** (Rein universelle, rein starre  $E$ -Unifikation)

Ein  $E$ -Unifikationsproblem  $\langle E, s, t \rangle$  heie

<sup>2</sup> Die meisten im folgenden gemachten Aussagen gelten schon, wenn die Menge  $E$  zumindest aufzhlbar ist, was jedoch im Einzelfall zu beweisen wre.

<sup>3</sup> Es gelte (ohne Beschrnkung der Allgemeinheit)  $\text{Bound}(\langle E, s, t \rangle) \cap \text{Free}(\langle E, s, t \rangle) = \emptyset$ .

<sup>4</sup> Wie beispielsweise in [Siekmann, 1989].

<sup>5</sup> „Rigid  $E$ -unification“.

- *rein universell, wenn  $E$  keine freien Variablen enthält.*
- *rein starr, wenn  $E$  keine gebundenen Variablen enthält.*

*In diesem Fall heie auch die Menge  $E$  rein universell bzw. rein starr.*

### Beispiel 3.1.4

Tabelle 3.1 zeigt einfache Beispiele fr gemischte, rein universelle und rein starre  $E$ -Unifikationsprobleme  $\langle E, s, t \rangle$  und ihre Lsungen. Es sind jeweils vollstndige Mengen allgemeinsten Unifikatoren angegeben.  $\square$

$E$	$s$	$t$	Unifikatoren	Typ
$\{f(x) \approx x\}$	$f(a)$	$a$	$\{x/a\}$	rein starr
$\{f(a) \approx a\}$	$f(x)$	$a$	$\{x/a\}^a$ $\{x/f(a)\}$ $\{x/f(f(a))\}$ ...	rein starr/ universell <sup>b</sup>
$\{(\forall x)(f(x) \approx x)\}$	$g(f(a), f(b))$	$g(a, b)$	$id$	rein universell
$\{f(x) \approx x\}$	$g(f(a), f(b))$	$g(a, b)$	— <sup>c</sup>	rein starr
$\{(\forall x)(f(x, y) \approx f(y, x))\}$	$f(a, b)$	$f(b, a)$	$\{y/b\}$	gemischt

<sup>a</sup> Bezglich der Subsumtionsrelation  $\leq_E$  ist schon  $\{\{x/a\}\}$  eine vollstndige Menge allgemeinsten Unifikatoren. Bezglich  $\leq$  ist die vollstndige Menge allgemeinsten Unifikatoren dagegen unendlich.

<sup>b</sup> Da  $E$  keine Variablen enthlt, ist das Problem sowohl rein universell als auch rein starr.

<sup>c</sup> Nicht unifizierbar.

**Tabelle 3.1:** Beispiele fr gemischte, rein starre und rein universelle  $E$ -Unifikationsprobleme (Bsp. 3.1.4)

Abbildung 3.1 zeigt, welche Versionen der  $E$ -Unifikation Spezialflle anderer Versionen sind. Da der Unifikationsbegriff aus Definition 3.1.2 in dieser Hierarchie der allgemeinste ist, ist die Bezeichnung als „gemischte“  $E$ -Unifikation eigentlich unntig. Weil die Definition aber unblich ist, und um Verwechslungen zu vermeiden, wird der Zusatz doch beibehalten.

Im folgenden wird es auch von groem Interesse sein, solche Substitutionen zu finden, die mehrere Unifikationsprobleme simultan lsen:

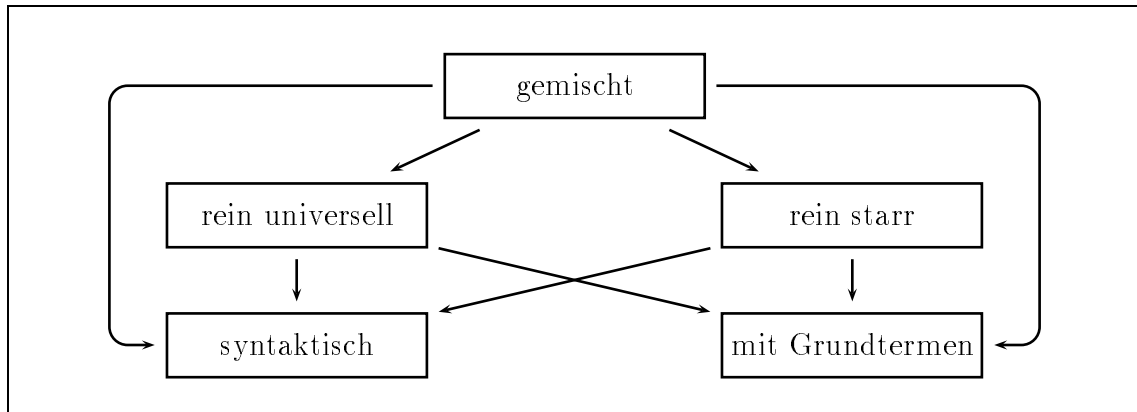
### Definition 3.1.5 (Simultanes $E$ -Unifikationsproblem)

*Ein simultanes  $E$ -Unifikationsproblem ist eine endliche Menge*

$$\{\langle E_1, s_1, t_1 \rangle, \dots, \langle E_n, s_n, t_n \rangle\} \quad (n \geq 1)$$

*gemischter  $E$ -Unifikationsprobleme.*

*Eine Substitution  $\sigma$  heit Lsung dieses simultanen Problems, wenn sie Lsung aller einfachen Probleme  $\langle E_k, s_k, t_k \rangle$  ( $1 \leq k \leq n$ ) ist.*



**Abbildung 3.1:** Dieses Diagramm zeigt die Hierarchie der verschiedenen Versionen der  $E$ -Unifikation. Die Pfeile können als „beinhaltet als Spezialfall“ gelesen werden; die Einschränkungen nehmen also nach unten zu.

## 3.2 Die Ableitbarkeitsrelation

Nahezu alle Verfahren zur Lösung von  $E$ -Unifikationsproblemen beruhen auf die ein oder andere Weise auf der folgenden Ableitbarkeitsrelation. Sie ist auf den Termen definiert, also auf syntaktischer Ebene, und eignet sich darum für Implementierungen.

**Definition 3.2.1** (Ableitbarkeitsrelation  $\leftrightarrow_E$ , Rechtfertigung, Beweis für  $s \overset{*}{\leftrightarrow}_E t$ )  
 Sei  $E$  eine Menge von Gleichungen. Die Relation  $\leftrightarrow_E$  ist auf der Menge der Terme definiert durch:

Für zwei Terme  $s, t \in \mathbf{Term}$  gilt

$$s \leftrightarrow_E t$$

genau dann, wenn es eine Gleichung  $G \in E$ ,

$$G = (\forall x_1) \dots (\forall x_n)(l \approx r) \quad \text{oder} \quad G = (\forall x_1) \dots (\forall x_n)(r \approx l) ,$$

eine Position  $p$  in  $s$  und eine Substitution  $\sigma$  gibt, so daß

1.  $s|_p = l\sigma$ ,
2.  $t = (s[p/r])\sigma$ ,
3.  $\text{Dom}(\sigma) \cap \text{Free}(G, s) = \emptyset$ , d.h.,  $\sigma$  belegt keine freien Variablen in der Gleichung  $G$  oder in  $s$ .<sup>6</sup>

In diesem Fall heißt das Tripel  $\langle G, p, \sigma \rangle$  eine Rechtfertigung für  $s \leftrightarrow_E t$ .

Eine endliche Folge  $\langle t_0, \dots, t_n \rangle$  ( $n \geq 0$ ) von Termen mit

$$t_0 \leftrightarrow_E t_1 \leftrightarrow_E t_2 \leftrightarrow_E \dots \leftrightarrow_E t_n$$

heißt ein  $E$ -Beweis für

$$t_0 \overset{*}{\leftrightarrow}_E t_n .$$

<sup>6</sup> Hieraus folgt  $t = (s[p/r])\sigma = s[p/r\sigma]$ .

Die Relation  $\leftrightarrow_E$  ist offenbar symmetrisch.

G. Birkhoff zeigte schon in den dreißiger Jahren, daß man auf die Ableitbarkeitsrelation aus Definition 3.2.1 ein korrektes und vollständiges Verfahren für die *rein universelle*  $E$ -Unifikation stützen kann. Wie der folgende Satz zeigt, behält dieses Ergebnis — die Relation  $\overset{*}{\leftrightarrow}_E$  ist gerade so definiert — auch für die gemischte  $E$ -Unifikation Gültigkeit:

**Satz 3.2.2**

Seien  $E$  eine Menge von Gleichungen,  $s$  und  $t$  Terme und  $\sigma$  eine Substitution. Es gilt

$$E \models (s \approx t) \quad \text{gdw.} \quad s \overset{*}{\leftrightarrow}_E t .$$

**Beweis:**

Es gilt  $E \models (s \approx t)$  über einer Signatur  $\mathcal{S}$  genau dann, wenn  $E \models (s \approx t)$  über der Signatur  $\mathcal{S}'$  gilt, in der die in  $\langle E, s, t \rangle$  frei auftretenden Variablen Konstanten sind (Def. 2.2.5). G. Birkhoff zeigte, daß dies genau dann gilt, wenn  $s$  aus  $t$  (bzw.  $t$  aus  $s$ ) ableitbar ist bezüglich der in der Literatur üblichen (beispielsweise in [Dershowitz, 1989]) Ableitbarkeitsrelation für rein universelle Gleichungsmengen  $E$ . Diese unterscheidet sich von der Ableitbarkeitsrelation  $\overset{*}{\leftrightarrow}_E$  (Def. 3.2.1) nur dadurch, daß die Einschränkung entfällt, daß die Substitution  $\sigma$  keine freien Variablen belegen darf. Diese Einschränkung wird aber in der Prädikatenlogik über  $\mathcal{S}'$  genau dadurch simuliert, daß die in  $E$  freien Variablen als Konstanten aufgefaßt werden (die Variablen in  $s$  und  $t$  werden ohnehin nicht belegt). Also ist  $s$  aus  $t$  bezüglich dieser Relation über der Signatur  $\mathcal{S}'$  genau dann ableitbar, wenn  $s \overset{*}{\leftrightarrow}_E t$  über  $\mathcal{S}$  gilt. ■

**Korollar 3.2.3**

Seien  $\langle E, s, t \rangle$  ein gemischtes  $E$ -Unifikationsproblem und  $\sigma$  eine Substitution.  $\sigma$  ist genau dann eine Lösung des Problems, wenn

$$s\sigma \overset{*}{\leftrightarrow}_{E\sigma} t\sigma$$

gilt.

**Beweis:**

Die Behauptung folgt mit Definition 3.1.2 sofort aus Satz 3.2.2. ■

Ein  $E$ -Unifikationsproblem  $\langle E, s, t \rangle$  kann also dadurch gelöst werden, daß man nach Substitutionen  $\sigma$  sucht, so daß es einen  $E\sigma$ -Beweis für  $s\sigma \overset{*}{\leftrightarrow}_{E\sigma} t\sigma$  gibt. Der Suchraum nach einer Lösung ist jedoch ungeheuer groß, denn:

1. Die Suche nach einem  $E\sigma$ -Beweis für  $s\sigma \overset{*}{\leftrightarrow}_{E\sigma} t\sigma$  ist, insbesondere wegen der Symmetrie von  $\overset{*}{\leftrightarrow}_{E\sigma}$ , ein äußerst indeterministischer Prozeß.
2. Korollar 3.2.3 liefert nur einen Test. Wie eine Substitution  $\sigma$ , die eine Lösung sein könnte, zu finden ist, bleibt zunächst unklar.

Diese beiden Probleme zu lösen, den Suchraum einzuschränken und so eine effiziente Implementierung zu ermöglichen, ist das Ziel aller im weiteren beschriebenen Verfahren.

Die Intention bei der Unterscheidung rein universeller, rein starrer und gemischter  $E$ -Unifikationsprobleme ist, wie schon in Abschnitt 3.1 kurz angedeutet, daß die Gleichungen in  $E$  in verschiedener Weise in einem  $E\sigma$ -Beweis für  $s\sigma \overset{*}{\leftrightarrow}_{E\sigma} t\sigma$  verwendet werden können, d. h. in einem Beweis dafür, daß  $\sigma$  ein (gemischter)  $E$ -Unifikator von  $s$  und  $t$  ist:

1. Im rein universellen Fall mehrfach, mit verschiedenen Instantiierungen für die Variablen, die sie enthalten.
2. Im rein starren Fall zwar auch mehrfach aber mit nur jeweils *einer* Instantiierung einer Variablen, die für alle Gleichungen und alle Anwendungen dieselbe ist.
3. Im gemischten Fall soll es beide Möglichkeiten geben.

Tatsächlich kommen diese Forderungen in der Definition der Ableitbarkeitsrelation  $\overset{*}{\leftrightarrow}_{E\sigma}$  zum Ausdruck, denn:

1. Enthält  $E$  eine rein universelle Gleichung  $G$ , so ist diese auch in  $E\sigma$  enthalten, denn alle Variablen in  $G$  sind gebunden, und daher durch die Substitution nicht belegt worden. Da in der Definition der Ableitbarkeitsrelation  $\leftrightarrow_E$  (Def. 3.2.1, Punkt 3) nur die Belegung freier Variablen eingeschränkt ist, kann die Gleichung  $G$  in einem  $E\sigma$ -Beweis mit beliebigen Instantiierungen verwendet werden.
2. Ist  $G$  dagegen eine rein starre Gleichung, enthält  $E\sigma$  die Gleichung  $G\sigma$ . Sie kann bei jedem Beweisschritt nur in genau dieser Form, mit den durch  $\sigma$  bestimmten Variablenbelegungen verwendet werden, denn auch die von  $\sigma$  nicht belegten, in  $G\sigma$  noch frei auftretenden Variablen dürfen wegen der Einschränkung in der Definition von  $\leftrightarrow_E$  (Def. 3.2.1, Punkt 3) nicht instantiiert werden.
3. Der Absicht, beide Möglichkeiten zugleich zur Verfügung zu haben, entspricht, daß nicht nur rein universelle und rein starre Gleichungen in einer (gemischten) Menge  $E$  liegen können, sondern daß sogar einzelne Gleichung  $G$  sowohl freie als auch universell gebundene Variablen enthalten und damit teilweise starr und teilweise universell sein können.

Daraus folgt mit Korollar 3.2.3 auch, daß die Definition der gemischten  $E$ -Unifikation der Intention entspricht.

Da die Gleichungen nicht beliebig instantiiert werden können, sind bei der gemischten  $E$ -Unifikation die Ressourcen, die zum Beweis zur Verfügung stehen, gegenüber der rein universellen  $E$ -Unifikation eingeschränkt, was in dem folgenden Satz zum Ausdruck kommt:



**Satz 3.2.4**

Sei  $\langle E, s, t \rangle$  ein gemischtes  $E$ -Unifikationsproblem. Enthält  $E_{\forall}$  den All-Abschluß aller Gleichungen in  $E$ , dann ist jede Substitution  $\sigma$ , die Lösung ist von  $\langle E, s, t \rangle$ , auch Lösung des rein universellen Problems  $\langle E_{\forall}, s, t \rangle$ .

**Beweis:**

Es gilt  $\leftrightarrow_E \subset \leftrightarrow_{E_{\forall}}$ , denn wie man anhand von Definition 3.2.1 leicht nachvollzieht, ist  $\langle G_{\forall}, p, \sigma \rangle$  eine Rechtfertigung für  $s \leftrightarrow_{E_{\forall}} t$ , wenn  $\langle G, p, \sigma \rangle$  eine Rechtfertigung für  $s \leftrightarrow_E t$  ist und  $G_{\forall} \in E_{\forall}$  der All-Abschluß der Gleichung  $G \in E$ .

Sei nun  $\sigma$  ein beliebiger Unifikator von  $\langle E, s, t \rangle$ . Dann folgt aus Korollar 3.2.3  $s\sigma \overset{*}{\leftrightarrow}_{E\sigma} t\sigma$  und also  $s\sigma \overset{*}{\leftrightarrow}_{E_{\forall}\sigma} t\sigma$  und daraus (wiederum mit Korollar 3.2.3), daß  $\sigma$  auch  $E$ -Unifikator von  $\langle E_{\forall}, s, t \rangle$  ist. ■

Zum Abschluß wird bewiesen, daß einige einfache, für rein universelle und rein starre  $E$ -Unifikationsprobleme bekanntermaßen geltende Aussagen<sup>7</sup> auf die gemischte  $E$ -Unifikation übertragen werden können.

**Lemma 3.2.5**

Sei  $\langle E, s, t \rangle$  ein gemischtes  $E$ -Unifikationsproblem mit

$$s \overset{*}{\leftrightarrow}_E t .$$

Dann gilt für jede Substitution  $\tau$

$$s\tau \overset{*}{\leftrightarrow}_{E\tau} t\tau .$$

**Beweis:**

Gelte zunächst  $s \overset{*}{\leftrightarrow}_E t$  mit einer Rechtfertigung  $\langle G, p, \sigma \rangle$ , wobei  $G = (\forall \bar{x})(l \approx r)$ . Sei  $\tau' = \tau|_{\text{Free}(G)}$  die Einschränkung von  $\tau$  auf die in  $G$  frei auftretenden Variablen.

Dann gilt  $s\tau \overset{*}{\leftrightarrow}_{E\tau} t\tau$  mit der Rechtfertigung  $\langle G\tau, p, \sigma \circ \tau' \rangle$ , denn:

1.  $G\tau = (\forall \bar{x})(l\tau' \approx r\tau')$ ;
2. da  $s$  keine der in  $G$  gebunden auftretenden Variablen enthält (siehe Fußnote zu Definition 3.1.2), und  $\sigma$  keine in  $G$  frei auftretenden Variablen belegt, folgt aus  $s|_p = l\sigma$ , daß  $s\tau|_p = s\tau'|_p = l\sigma\tau' = l\tau'\sigma$  gilt.
3. In gleicher Weise folgt aus  $t = (s[p/r])\sigma$ , daß  $t\tau = s[p/r\sigma] = (s[p/r])\tau'\sigma$ .
4. Schließlich gilt auch  $\text{Dom}(\sigma \circ \tau') \cap \text{Free}(G\tau) = \emptyset$ .

Sei nun

$$s = t_0 \leftrightarrow_E t_1 \leftrightarrow_E \cdots \leftrightarrow_E t_n = t$$

ein Beweis für  $s \overset{*}{\leftrightarrow}_E t$ . Aus dem bisher bewiesenen folgt, daß dann

$$s\tau = t_0\tau \leftrightarrow_{E\tau} t_1\tau \cdots \leftrightarrow_{E\tau} t_n\tau = t\tau$$

ein Beweis für  $s\tau \overset{*}{\leftrightarrow}_{E\tau} t\tau$  ist. ■

<sup>7</sup> Für die rein universellen  $E$ -Unifikationsprobleme sind sie in [Siekmann, 1989], für die rein starren Probleme in [Gallier *et al.*, 1992] enthalten.

Aus dem Lemma folgt sofort:

**Satz 3.2.6**

Die Substitution  $\sigma$  sei Lösung eines gemischten  $E$ -Unifikationsproblems  $\langle E, s, t \rangle$ . Dann ist jede Spezialisierung  $\tau$  von  $\sigma$  (d. h.  $\sigma \leq \tau$ ) ebenfalls Lösung von  $\langle E, s, t \rangle$ .

**Beweis:**

Sei  $\tau = \sigma' \circ \sigma$ . Auch  $\langle E\sigma, s\sigma, t\sigma \rangle$  ist ein gemischtes  $E$ -Unifikationsproblem und erfüllt gemäß Korollar 3.2.3 die Voraussetzungen von Satz 3.2.5, und es folgt

$$s\sigma\sigma' \stackrel{*}{\leftrightarrow}_{E\sigma\sigma'} t\sigma\sigma'$$

und also

$$s\tau \stackrel{*}{\leftrightarrow}_{E\tau} t\tau .$$

Daraus folgt (wieder mit Korollar 3.2.3), daß  $\tau$  eine Lösung des Problems  $\langle E, s, t \rangle$  ist. ■

**Satz 3.2.7**

Sei  $E$  eine Menge von Gleichungen,  $t, l, r$  seien Terme und  $p$  sei eine Position in  $t$ . Gilt

$$l \stackrel{*}{\leftrightarrow}_E r ,$$

dann auch

$$s[p/l] \stackrel{*}{\leftrightarrow}_E s[p/r] ,$$

d. h.,  $\stackrel{*}{\leftrightarrow}_E$  ist monoton bezüglich der Termstruktur.

**Beweis:**

Sei  $\mathcal{S}$  die verwendete Signatur, und  $\mathcal{S}'$  — wie in Definition 2.2.5 — die Signatur, in der die auftretenden freien Variablen Konstantensymbole sind.

Aus  $l \stackrel{*}{\leftrightarrow}_E r$  folgt zunächst  $E \stackrel{\cong}{\models} (l \approx r)$  und also  $\mathcal{M} \stackrel{\cong}{\models} (l \approx r)$  für jedes kanonische Modell  $\mathcal{M}$  von  $E$  über der Signatur  $\mathcal{S}'$ . Wegen der Monotonie der Gleichheit bezüglich der Termstruktur, und weil  $\mathcal{M}$  ein *kanonisches* Modell ist, gilt dann auch  $\mathcal{M} \stackrel{\cong}{\models} (s[l] \approx s[r])$ . Nun kann man die Schlußkette umkehren: Da  $\mathcal{M}$  beliebig gewählt werden kann, folgt  $E \stackrel{\cong}{\models} (s[l] \approx s[r])$ , und daraus  $s[l] \stackrel{*}{\leftrightarrow}_E s[r]$ . ■

### 3.3 Vollständige Mengen allgemeinsten Unifikatoren

Oft reicht es nicht aus, nur (irgend-)einen Unifikator für ein gegebenes Problem zu bestimmen. Alle Lösungen benötigt man in der Regel aber auch nicht, sondern es genügt eine in gewissem — im folgenden präzisierten — Sinne vollständige Menge von Unifikatoren.

Eine naheliegende und häufig angewendete Möglichkeit, die Zahl der Unifikatoren, die in einer solch vollständigen Menge von Unifikatoren liegen müssen, und damit auch den Suchraum einzuschränken, ist, nur *allgemeinste* Unifikatoren<sup>8</sup> zu berechnen. Allgemeinste Unifikatoren sind solche Lösungen eines Unifikationsproblems,

<sup>8</sup> Sie werden im folgenden auch als MGUs (*most general unifiers*) bezeichnet.

die minimal sind bezüglich einer auf den Substitutionen definierten Subsumtionsrelation  $\leq$ .<sup>9</sup> Diese Relation muß sinnvoller Weise zumindest die folgenden Eigenschaften haben:

- (S1) Sie muß eine Halbordnung sein, d. h. transitiv und reflexiv.
- (S2) Sie muß abgeschlossen sein bezüglich der Menge der Unifikatoren. Ist also  $\sigma$  Lösung eines gegebenen Problems, und gilt  $\sigma \leq \tau$ , so muß auch  $\tau$  Lösung sein.

Für eine Implementierung sind außerdem folgende Eigenschaften wichtig:

- (S3) Die Relation selbst sollte einfach berechenbar, d. h.  $\sigma \leq \tau$  für gegebene Substitutionen  $\sigma$  und  $\tau$  schnell zu überprüfen sein.
- (S4) Es sollte entscheidbar sein, ob zwei gegebene Substitutionen  $\sigma_1, \sigma_2$  eine gemeinsame Spezialisierung, d. h. eine Substitution  $\tau$  mit

$$\sigma_1 \leq \tau \quad \text{und} \quad \sigma_2 \leq \tau ,$$

besitzen, und eine solche sollte ggf. möglichst einfach zu berechnen sein.

Unter Verwendung einer Relation  $\leq$  mit diesen Eigenschaften ist nun eine vollständige Menge allgemeinsten Unifikatoren in folgender Weise definiert (beispielsweise [Siekman, 1989]):

**Definition 3.3.1** (Vollständige Menge allgemeinsten Unifikatoren)

Ist  $\mathcal{L}$  die Menge aller Lösungen eines  $E$ -Unifikationsproblems, dann heie eine Menge  $\mathcal{U} \subset \mathbf{Subst}$ , die die Bedingung

- (1)  $\mathcal{U} \subset \mathcal{L}$  (Korrektheit)

erfüllt, eine vollständige Menge allgemeinsten Unifikatoren des Problems, falls:

- (2a) Für jedes  $\tau \in \mathcal{L}$  gibt es ein  $\sigma \in \mathcal{U}$  mit  $\sigma \leq \tau$  (Vollständigkeit).

Sie heie grund-vollständige Menge allgemeinsten Unifikatoren, falls:

- (2b) Für jede Grundsubstitution  $\tau \in \mathcal{L}$  gibt es ein  $\sigma \in \mathcal{U}$  mit  $\sigma \leq \tau$  (Grund-Vollständigkeit).

$\mathcal{U}$  heit eine minimale vollständige (bzw. grund-vollständige) Menge allgemeinsten Unifikatoren, falls außerdem gilt:

- (3) Es gibt keine Unifikatoren  $\sigma_1, \sigma_2 \in \mathcal{U}$ ,  $\sigma_1 \neq \sigma_2$ , mit  $\sigma_1 \leq \sigma_2$  (Minimalität).

---

<sup>9</sup> Die Orientierung des Symbols  $\leq$  widerspricht der Intuition:  $\sigma \leq \tau$  bedeutet, daß die „kleinere“ Substitution  $\sigma$  die „größere“ Substitution  $\tau$  subsumiert.

Eine zu der in Definition 3.3.1 äquivalente Formulierung der Minimalität ist die, daß eine Menge allgemeinsten Unifikatoren unvollständig wird, wenn man auch nur einen Unifikator aus ihr entfernt. Die Forderung nach Minimalität kann u. U. eingeschränkt werden, wenn das Entfernen überflüssiger Unifikatoren einen zu großen Aufwand bedeutet.

Für praktisch alle Anwendungen, insbesondere auch für das automatische Beweisen, ist es völlig ausreichend, statt der Menge  $\mathcal{L}$  aller Lösungen eines Unifikationsproblems, eine vollständige Menge  $\mathcal{U}$  allgemeinsten Unifikatoren zu berechnen. Auch genügt es für viele Fragestellungen, nur eine grund-vollständige Menge von Unifikatoren anzugeben, beispielsweise, wenn es darum geht, die Unifizierbarkeit von Termen zu überprüfen.

Normalerweise wird für die Definition allgemeinsten Unifikatoren die gewöhnliche Spezialisierungsrelation  $\leq$  (Def. 2.3.6) verwendet. Sie hat die Eigenschaften (S1) bis (S4). Bei der gemischten  $E$ -Unifikation ist aber auch die folgende Subsumtionsrelation  $\leq_E$  sinnvoll:<sup>10</sup>

**Definition 3.3.2** (Die Relationen  $\sqsubseteq_E$  und  $\leq_E$ )  
*Sei  $E$  eine Menge von Gleichungen. Dann sind die Relationen*

$$\sqsubseteq_E, \leq_E \in \mathbf{Subst} \times \mathbf{Subst}$$

*auf den Substitutionen definiert durch:*

$$\sigma \sqsubseteq_E \tau$$

*genau dann, wenn*

$$E\tau \stackrel{\approx}{=} \sigma(x) \approx \tau(x) \quad \text{für alle } x \in \mathcal{V} \text{ ,}$$

*und*

$$\sigma \leq_E \tau$$

*genau dann, wenn es eine Substitution  $\sigma'$  gibt, so daß*

$$(\sigma' \circ \sigma) \sqsubseteq_E \tau \text{ .}$$

Die intuitive Bedeutung von  $\sigma \leq_E \tau$  ist, daß es eine Spezialisierung von  $\sigma$  gibt, die mit Hilfe von  $E\tau$  aus  $\tau$  abgeleitet werden kann. Damit hängt die Relation  $\leq_E$  — anders als die Spezialisierungsrelation  $\leq$  — von der jeweiligen Menge  $E$  von Gleichungen ab.

### Beispiel 3.3.3

Sei  $E_1 = \{f(a) \approx a\}$ . Dann gilt  $\{x/f(a)\} \sqsubseteq_{E_1} \{x/a\}$  und  $\{x/f(y)\} \leq_{E_1} \{x/a\}$ .

Mit  $E_2 = \{x \approx a\}$  gilt  $\{x/a\} \sqsubseteq_{E_2} \{x/b\}$ , aber *nicht*  $\{x/b\} \sqsubseteq_{E_2} \{x/a\}$ . Die Relationen  $\sqsubseteq_E$  und  $\leq_E$  sind also nicht symmetrisch.  $\square$

<sup>10</sup> Sie ist in ähnlicher Weise in [Gallier *et al.*, 1992] definiert — dort allerdings nur für die rein starre  $E$ -Unifikation. Auch ihre Einschränkung auf rein universelle Mengen  $E$  wird in der Literatur verwendet. Für diese, insbesondere für  $E = \emptyset$ , hat  $\leq_E$  die Eigenschaften (S1) und (S2), was es rechtfertigt, — wie in der Literatur zur rein universellen  $E$ -Unifikation üblich — die notwendigen Eigenschaften von  $\leq$  nicht explizit aufzuzählen, sondern als gegeben hinzunehmen.

Für  $E = \emptyset$  stimmt  $\leq_E$  mit  $\leq$  überein:

**Lemma 3.3.4**

*Es gilt*

$$\leq_{\emptyset} = \leq .$$

**Beweis:**

Seien  $\sigma$  und  $\tau$  beliebige Substitutionen. Wie aus Definition 3.3.2 folgt, gilt  $\sigma \sqsubseteq_{\emptyset} \tau$  genau dann, wenn  $\sigma = \tau$ , also  $\sigma \leq_{\emptyset} \tau$  genau dann, wenn es eine Substitution  $\sigma'$  gibt, so daß  $(\sigma' \circ \sigma) = \tau$ . Dies wiederum ist gleichbedeutend mit  $\sigma \leq \tau$ . ■

Die Relation  $\leq_E$  hat die für eine Subsumtionsrelation auf den Substitutionen notwendigen Eigenschaften (S1) und (S2):

**Lemma 3.3.5**

*Sei  $\langle E, s, t \rangle$  ein gemischtes  $E$ -Unifikationsproblem. Dann gilt:*

(S1)  $\leq_E$  ist eine Halbordnung.

(S2) Ist  $\sigma$  ein Unifikator, und gilt  $\sigma \leq_E \tau$ , dann ist auch  $\tau$  ein Unifikator.

**Beweis:**

(S1): Die Reflexivität der Relation  $\leq_E$  folgt direkt aus ihrer Definition. Zu zeigen bleibt, daß  $\leq_E$  auch transitiv ist. Seien also  $\sigma, \tau, \rho$  beliebige Substitutionen mit

$$\sigma \leq_E \tau \leq_E \rho .$$

Es gibt also Spezialisierungen  $\sigma' = (\sigma'' \circ \sigma)$  und  $\tau' = (\tau'' \circ \tau)$ , so daß

$$\sigma' \sqsubseteq_E \tau \quad \text{und} \quad \tau' \sqsubseteq_E \rho$$

und also

$$E\tau \stackrel{\cong}{=} (\sigma'(x) \approx \tau(x)) \quad \text{und} \quad E\rho \stackrel{\cong}{=} (\tau'(x) \approx \rho(x))$$

für jede Variable  $x$ . Daraus kann man wegen der Monotonie der Gleichheit bezüglich der Termstruktur

$$E\tau' \stackrel{\cong}{=} ((\tau'' \circ \sigma')(x) \approx \tau'(x))$$

und für alle  $G \in E\tau'$

$$E\rho \stackrel{\cong}{=} G$$

folgern, und daraus wiederum

$$E\rho \stackrel{\cong}{=} (\tau'' \circ \sigma')(x) \approx \rho(x) .$$

Dann gilt aber, da dies für jede Variable  $x$  gilt, und  $(\tau'' \circ \sigma')$  eine Spezialisierung von  $\sigma$  ist, auch  $\sigma \leq_E \rho$ .

(S2): Seien  $\sigma$  und  $\tau$  Substitutionen mit  $\sigma \sqsubseteq_E \tau$ , und  $\sigma$  sei ein Unifikator von  $\langle E, s, t \rangle$ . Es gilt also  $E\sigma \stackrel{\cong}{=} (s\sigma \approx t\sigma)$  und  $E\tau \stackrel{\cong}{=} (\sigma(x) \approx \tau(x))$  für alle Variablen  $x$ . Wegen der Monotonie der Gleichheit folgt  $E\tau \stackrel{\cong}{=} G$  für alle  $G \in E\sigma$ , daraus  $E\tau \stackrel{\cong}{=} (s\sigma \approx t\sigma)$  und wieder wegen der Monotonie der Gleichheit  $E\tau \stackrel{\cong}{=} (s\tau \approx t\tau)$ .

Gelte  $\sigma \leq_E \tau$ . Dann gibt es eine Spezialisierung  $\sigma'$  von  $\sigma$  mit  $\sigma' \sqsubseteq_E \tau$ .  $\sigma'$  ist ebenfalls ein Unifikator (Satz 3.2.6) und also, wie aus der schon bewiesenen Abgeschlossenheit von  $\sqsubseteq_E$  folgt, auch  $\tau$ . ■

Die Eigenschaft (S3) hat die Relation  $\sqsubseteq_E$  jedoch nicht, denn für beliebige  $E$  ist bei gegebenen  $\sigma$  und  $\tau$  die zu überprüfende Bedingung

$$E\tau \stackrel{\approx}{\equiv} \sigma(x) \approx \tau(x)$$

nicht entscheidbar (wohl aber für rein starre  $E$ ). Ein noch größeres Problem stellt die — eigentlich wünschenswerte — Eigenschaft (S4) dar. Da die Relation  $\sqsubseteq_E$  nicht symmetrisch ist (Beispiel 3.3.3), ist es schwierig, bei gegebener Substitution  $\sigma$  solche Substitutionen  $\tau$  zu bestimmen, für die  $\sigma \sqsubseteq_E \tau$  gilt. Das ist aber notwendig, um eine Vereinigung zweier Substitutionen  $\sigma_1$  und  $\sigma_2$  zu berechnen.

Trotzdem überwiegen die Vorteile, wenn man die Relation  $\leq_E$  statt der Spezialisierungsrelation  $\leq$  als Subsumtionsrelation verwendet, da vollständige Mengen allgemeinster Unifikatoren mit ihr sehr viel weniger Elemente haben.

### 3.4 Entscheidbarkeit

Bevor im nächsten Abschnitt auf verschiedene Verfahren zur  $E$ -Unifikation näher eingegangen wird, soll hier zunächst geklärt werden, in welchen Fällen die Existenz einer Lösung überhaupt entscheidbar ist, und welche Eigenschaften (endlich, entscheidbar oder aufzählbar) die Mengen allgemeinster Unifikatoren haben. Tabelle 3.2 gibt eine Übersicht.

Problem	Entscheidbarkeit	MGUs, vollständig <sup>a</sup>	Simultanes Problem	Verfahren
Syntaktische Unif., $E = \emptyset$	entscheidbar	einelementig (bzw. leer)	entscheidbar	Robinson, Martelli-Montanari
Grundterme u. -gleichungen, $\text{Var}(\langle E, s, t \rangle) = \emptyset$	entscheidbar	einelementig (bzw. leer)	entscheidbar	[Shostak, 1978], [Nelson & Oppen, 1980]
Rein universell, $\text{Free}(E) = \emptyset$	unentscheidbar	aufzählbar	unentscheidbar	Knuth-Bendix-Verf., Unfailing KBA
Rein starr, $\text{Bound}(E) = \emptyset$	entscheidbar	endlich/aufzählbar <sup>b</sup>	entscheidbar (?) <sup>c</sup>	[Gallier <i>et al.</i> , 1992], [Goubault, 1993]
Gemischt, $E$ beliebig	unentscheidbar	aufzählbar	unentscheidbar	[Beckert, 1991], Verfahren in Kapitel 4

<sup>a</sup> Vollständige Menge allgemeinster Unifikatoren.

<sup>b</sup> Endlich bezüglich der Relation  $\leq_E$  auf den Substitutionen, aufzählbar bezüglich  $\leq$ .

<sup>c</sup> [Gallier *et al.*, 1992] enthält zwar ein entsprechendes Theorem; an dessen Korrektheit bestehen jedoch gewisse Zweifel (siehe Abschnitt 3.6.5).

**Tabelle 3.2:** Entscheidbarkeit der verschiedenen Versionen der  $E$ -Unifikation.

#### Gemischte $E$ -Unifikation

Die gemischte  $E$ -Unifikation ist, wie schon ihr rein universeller Spezialfall (s. u.), unentscheidbar. Eine vollständige Menge allgemeinster Unifikatoren ist jedoch — gestützt auf Satz 3.2.2 und sein Korollar — in jedem Falle aufzählbar.<sup>11</sup>

<sup>11</sup> Wie dies in effizienter Weise geschehen kann, ist, neben der Anwendung auf den Tableauekalkül, die wesentliche Fragestellung dieser Arbeit.

### Syntaktische Unifikation

Die syntaktische Unifikation, d. h. die  $E$ -Unifikation für  $E = \emptyset$ , — ohne sie kommt praktisch kein Kalkül für die Prädikatenlogik aus — ist bekanntlich entscheidbar. Mengen allgemeinsten Unifikatoren sind einelementig (vorausgesetzt es existiert ein Unifikator).

Neben dem Standardverfahren von J. A. Robinson gibt es eine Reihe sehr effizienter Methoden, insbesondere auch für simultane Probleme (beispielsweise das von Martelli und Montanari [1982]).

### $E$ -Unifikation mit Grundtermen und -gleichungen

Unifikationsprobleme, die ausschließlich aus Grundtermen bestehen, sind entscheidbar. Entsprechende Verfahren bilden dazu einen Kongruenzabschluß bezüglich der Ableitbarkeitsrelation (siehe Abschnitt 3.5).

Da für ein solches Problem stets entweder alle Substitutionen Lösungen sind oder keine, besteht eine vollständige Menge allgemeinsten Unifikatoren nur aus der bezüglich der verwendeten Subsumtionsrelationen allgemeinsten Substitution *id*.

### Rein universelle $E$ -Unifikation

Es ist bekannt, daß schon die rein universelle  $E$ -Unifikation — und damit erst recht auch die gemischte — im allgemeinen unentscheidbar ist. Eine große Rolle spielt darum die Frage, für welche speziellen Gleichungsmengen  $E$  sie dennoch entscheidbar ist. Diese Frage ist für eine große Zahl von  $E$  untersucht worden ([Siekman, 1989] gibt eine Übersicht).

#### Beispiel 3.4.1

Unentscheidbar ist die rein universelle  $E$ -Unifikation beispielsweise für jede Menge  $E$  von Gleichungen, die das Assoziativitätsaxiom

$$As = (\forall x)(\forall y)(\forall z)(f(f(x, y), z) \approx f(x, f(y, z)))$$

und die Distributivitätsaxiome

$$\begin{aligned} Dis_R &= (\forall x)(\forall y)(\forall z)(f(x, f(y, z)) \approx f(f(x, y), f(x, z))) \\ Dis_L &= (\forall x)(\forall y)(\forall z)(f(f(x, y), z) \approx f(f(x, y), f(x, z))) \end{aligned}$$

für auch nur ein Funktionssymbol  $f$  enthält. □

Entscheidbar ist die rein universelle  $E$ -Unifikation insbesondere immer dann, wenn ein zu  $E$  äquivalentes vollständiges Reduktionssystem (siehe Abschnitt 3.6) existiert. Auf die Konstruktion und Anwendung solcher Reduktionssysteme stützen sich die vervollständigungsbasierten Verfahren zur rein universellen  $E$ -Unifikation, insbesondere der Knuth-Bendix-Algorithmus (siehe Abschnitt 3.6.4).

### Rein starre $E$ -Unifikation

Die rein starre  $E$ -Unifikation ist NP-vollständig und also entscheidbar [Gallier *et al.*, 1992]. Dieses Ergebnis beruht ganz wesentlich darauf, daß Mengen allgemeinsten Unifikatoren für rein starre Probleme als endlich und berechenbar bewiesen werden, und zwar bezüglich der Subsumtionsrelation  $\leq_E$ . Bezüglich der Relation  $\leq$  sind allgemeinste Unifikatoren auch für rein starre Probleme nur aufzählbar (siehe Abschnitt 3.6.5).

### Simultane Probleme

Da schon die einfache gemischte  $E$ -Unifikation unentscheidbar ist, gilt das erst recht für simultane Probleme. In wie weit die — für die verschiedenen Spezialfälle der gemischten  $E$ -Unifikation unterschiedlichen — Eigenschaften (Endlichkeit, Entscheidbarkeit) vollständiger Mengen allgemeinsten Unifikatoren erhalten bleiben, hängt ganz wesentlich von der verwendeten Subsumtionsrelation auf den Substitutionen ab; insbesondere davon, ob diese Relation neben den notwendigen Eigenschaften (S1) und (S2) auch die Eigenschaft (S4) hat. Denn um die Lösungen eines simultanen Problems aus den Lösungen der Einzelprobleme zu berechnen, ist es notwendig, die Vereinigung von Lösungen, d. h. gemeinsame Spezialisierungen von Unifikatoren, zu bilden.

Im Gegensatz zu  $\leq$  hat die Relation  $\leq_E$  die Eigenschaft (S4) — zumindest für beliebige Mengen  $E$  — nicht. Ob sie sie für rein starre  $E$  hat, ist zumindest fraglich (siehe Abschnitt 3.6.5).

Die *Aufzählbarkeit* der Lösungen eines gemischten  $E$ -Unifikationsproblems, wie auch vollständiger Mengen allgemeinsten Unifikatoren, bleibt in jedem Fall erhalten, weil die Schnittmenge endlich vieler aufzählbarer Mengen wieder aufzählbar ist.

Sind bei einem simultanen Problem die Mengen der Gleichungen identisch, hat es also die Form

$$\{\langle E, s_1, t_1 \rangle, \dots, \langle E, s_n, t_n \rangle\} ,$$

so ist es äquivalent zu dem einfachen Problem

$$\langle E, f(s_1, \dots, s_n), f(t_1, \dots, t_n) \rangle$$

(dabei ist  $f$  ein beliebiges, im ursprünglichen Problem nicht auftretendes Funktionssymbol). Solche simultanen Probleme haben also die gleichen Eigenschaften wie einfache Probleme und können mit entsprechenden Verfahren gelöst werden.

## 3.5 Berechnung von Äquivalenzklassen

Eine Möglichkeit,  $E$ -Unifikationsprobleme zu lösen, besteht darin, die Äquivalenzklassen der zu unifizierenden Terme bezüglich der Ableitbarkeitsrelation  $\overset{*}{\leftrightarrow}_E$  zu berechnen.

**Definition 3.5.1** (Äquivalenzklasse  $[t]_E$ )

Sei  $t$  ein Term und  $E$  eine Menge von Gleichungen. Dann bezeichnet  $[t]_E$  die Äquivalenzklasse von  $t$  bezüglich  $\overset{*}{\leftrightarrow}_E$ .



Aus Korollar 3.2.3 folgt sofort:

**Satz 3.5.2**

Eine Substitution  $\sigma$  ist genau dann eine Lösung des  $E$ -Unifikationsproblems  $\langle E, s, t \rangle$ , wenn

$$[s\sigma]_{E\sigma} = [t\sigma]_{E\sigma} ,$$

d. h. genau dann, wenn es einen Term

$$t' \in [s\sigma]_{E\sigma} \cap [t\sigma]_{E\sigma}$$

gibt.

Es bleiben jedoch zwei Schwierigkeiten zu lösen: Zum einen liefert Satz 3.5.2 nur einen Test und macht keine Aussage darüber, wie ein Unifikator  $\sigma$  zu bestimmen ist; und zum zweiten ist unklar, wie man bei der Berechnung der Äquivalenzklassen von  $s$  und  $t$  Redundanzen vermeiden und möglichst schnell einen Term  $t'$  finden kann, der beweist, daß die Klassen gleich sind.

Beide Probleme spielen keine Rolle, wenn ein  $E$ -Unifikationsproblem nur Grundterme enthält. Bei einem solchen sind entweder alle Substitutionen Unifikatoren oder aber keine. Es ist also nur von Interesse, ob es lösbar ist. Diese Frage ist, wie aus dem folgenden Satz [Shostak, 1978] folgt, entscheidbar:

**Satz 3.5.3**

Sei  $\langle E, s, t \rangle$  ein  $E$ -Unifikationsproblem mit  $\text{Var}(\langle E, s, t \rangle) = \emptyset$ .

Ist das Problem lösbar, dann gibt es einen Beweis

$$\langle s, t_1, \dots, t_n, t \rangle \quad (n \geq 0)$$

für  $s \xrightarrow{*}_E t$ , so daß die Terme  $t_1, \dots, t_n$  sämtlich Terme in oder Unterterme von Termen in  $\langle E, s, t \rangle$  sind.

Um ein nur aus Grundgleichungen und -termen bestehendes  $E$ -Unifikationsproblem zu lösen, genügt es also, den Kongruenzabschluß<sup>12</sup> derjenigen Terme und Unterterme zu berechnen, die in  $\langle E, s, t \rangle$  auftreten. Verfahren, die auf diesem Kongruenzabschluß beruhen, sind darum sehr effizient.<sup>13</sup>

Auch bei rein universellen  $E$ -Unifikationsproblemen kann man das Problem, daß Satz 3.5.2 nur einen Test liefert umgehen: Für rein universelle Gleichungsmengen  $E$  hängt die Äquivalenzklasse  $[t]_{E\sigma}$  eines Terms  $t$  wegen  $E\sigma = E$  nicht von der Substitution  $\sigma$  ab. Mit Hilfe des folgenden Satzes kann man es außerdem vermeiden,  $[t\sigma]_E$  für verschiedene  $\sigma$  berechnen zu müssen, indem man ein rein universelles Problem  $\langle E, s, t \rangle$  auf ein anderes rein universelles Problem  $\langle E', s', t' \rangle$  reduziert, bei dem  $s'$  und  $t'$  Grundterme sind:

<sup>12</sup> [Goubault, 1993] enthält eine elegante Fixpunktdefinition des Kongruenzabschlusses (wobei die Menge der Klasseneinteilungen als Verband aufgefaßt wird), aus der sofort die Berechenbarkeit des Abschlusses (und sogar ein Verfahren dafür) ableitbar ist.

<sup>13</sup> Ein sehr effizientes und ausgefeiltes Verfahren ist beispielsweise in [Nelson & Oppen, 1980] angegeben; es verwendet u. a. Hashing und Methoden der Graphentheorie.

**Satz 3.5.4**

Ein rein universelles  $E$ -Unifikationsproblem  $\langle E, s, t \rangle$  besitzt genau dann eine Lösung, wenn das rein universelle Problem

$$\langle E \cup \{(\forall x)(eq(x, x) \approx \text{true}, (\forall y_1) \dots (\forall y_n)(eq(s, t) \approx \text{false})\}, \text{true}, \text{false} \rangle$$

eine Lösung besitzt. Dabei sind  $y_1, \dots, y_n$  die in  $s$  und  $t$  auftretenden Variablen,  $\text{true}$  und  $\text{false}$  sind neue Konstanten,  $eq$  ist ein neues Prädikatensymbol und  $x$  eine neue Variable.

Allerdings erlaubt es dieser Satz nicht — zumindest nicht ohne weiteres — Unifikatoren für das ursprüngliche Problem  $\langle E, s, t \rangle$  zu bestimmen, wenn man festgestellt hat, daß solche existieren müssen, weil  $\langle E', s', t' \rangle$  lösbar ist.

Auch gilt Satz 3.5.3 für rein universelle Probleme nicht — das würde ihrer Unentscheidbarkeit widersprechen. Darum bleibt das Problem, die Äquivalenzklassen der Terme  $\text{true}$  und  $\text{false}$  in effizienter Weise zu bestimmen, schwierig. Als Lösung bieten sich vervollständigungsbasierte Verfahren an (siehe Abschnitt 3.6).

Bei allen  $E$ -Unifikationsproblemen, die nicht rein universell sind, also bei der allgemeinen gemischten  $E$ -Unifikation, aber auch schon bei rein starren Problemen, kommt das Problem voll zum Tragen, daß man eigentlich einen möglichen Unifikator  $\sigma$  schon kennen muß, bevor man daran gehen kann, die Äquivalenzklassen zu berechnen; denn für nicht rein universelle  $E$  hängt die Relation  $\overset{*}{\leftrightarrow}_{E\sigma}$  und damit die Äquivalenzklasse  $[t]_{E\sigma}$  eines Termes von  $\sigma$  ab.

Es ist viel zu aufwendig, alle Substitutionen der Reihe nach auszuprobieren. Selbst wenn man dabei Heuristiken anwendet, um Redundanzen zu vermeiden, und nur möglichst allgemeine Substitutionen verwendet und nur solche, die zu unterschiedlichen Äquivalenzklassen führen, wird doch viel zu viel redundante Information erzeugt; der Suchraum ist für eine Implementierung viel zu groß.

Allerdings muß man berücksichtigen, daß in der Regel auch für verschiedene  $\sigma$  sich unterscheidende Äquivalenzklassen  $[t]_{E\sigma}$  zumindest doch in ihrer Struktur sehr ähnlich sind.

Das in [Beckert, 1991, Beckert & Hähnle, 1992] beschriebene Verfahren nutzt diese Tatsache dadurch aus, daß alle verschiedenen Äquivalenzklassen eines Terms durch nur eine Menge  $\langle t \rangle_E$  dargestellt werden:

**Definition 3.5.5** (Die Menge  $\langle t \rangle_E$ )

Sie  $t$  ein Term und  $E$  eine Menge von Gleichungen. Dann ist

$$\langle t \rangle_E = \{s_\sigma : \sigma \text{ ist eine allgemeinste Substitution (bzgl. } \leq), \text{ so daß } s\sigma \in [t\sigma]_{E\sigma} \} .$$

Die an ein Element  $s_\sigma \in \langle t \rangle_E$  angeheftete (allgemeinste) Substitution  $\sigma$  kann als notwendige Bedingungen dafür aufgefaßt werden, daß  $s$  (bzw.  $s\sigma$ ) aus  $t$  (bzw.  $t\sigma$ ) ableitbar ist. Die Mengen  $\langle t \rangle_E$  enthalten nur wenig Redundanz, da ihre Definition eine Beschränkung auf allgemeinste Substitutionen enthält, und, was in verschiedenen Äquivalenzklassen gleich ist, in ihnen nur einmal vorkommt.

In [Beckert, 1991] ist beschrieben, wie die Mengen  $\langle t \rangle_E$  recht effizient berechnet werden können. Das Verfahren beruht darauf, durch Anwendung der Gleichungen in  $E$  induktiv Mengen  $\langle t \rangle_E^0, \langle t \rangle_E^1, \langle t \rangle_E^2, \dots$  zu konstruieren, die, ausgehend von  $\langle t \rangle_E^0 = \{t_{id}\}$ , die Menge  $\langle t \rangle_E$  annähern.

Darauf aufbauend kann man für jedes beliebige gemischte  $E$ -Unifikationsproblem  $\langle E, s, t \rangle$  eine Menge allgemeinsten Unifikatoren bestimmen:

**Satz 3.5.6**

*Sei  $\langle E, s, t \rangle$  ein gemischtes  $E$ -Unifikationsproblem. Dann ist die Menge*

$$\mathcal{U} = \{ \sigma_1 \sqcup \sigma_2 \sqcup \sigma_3 : \text{es gibt } t'_{\sigma_1} \in [t]_E, s'_{\sigma_s} \in [s]_E, \text{ so daß } s', t' \text{ (syntaktisch) unifizierbar sind mit MGU } \sigma_3 \}$$

*eine bezüglich der Subsumtionsrelation  $\leq$  auf den Substitutionen vollständige Menge allgemeinsten Unifikatoren von  $\langle E, s, t \rangle$ .*

**Beweis:**

Beweis zu Satz 4.7.13 in [Beckert, 1991]. ■

Bei der Anwendung dieses Verfahrens ist es nicht nötig und wäre eher von Nachteil, den „Trick“ aus Satz 3.5.4 (bzw. seine starre Version, Satz 3.6.18) anzuwenden.

Dieses Verfahren ist das bisher einzige zur Lösung allgemeiner gemischter  $E$ -Unifikationsprobleme. Es krankt jedoch an der Möglichkeit die Gleichungen in  $E$  bei der Konstruktion von  $\langle t \rangle_E$  symmetrisch, in beide Richtungen anzuwenden, was trotz der Verbesserungen und zusätzlich angewandeter Heuristiken zu einem großen Suchraum führt. Zudem ist die  $E$ -Unifizierbarkeit von Termen — auch für rein starre Probleme — mit dieser Methode nicht entscheidbar.<sup>14</sup> Diese Probleme zu lösen und die genannten Vorteile mit denen der im nächsten Abschnitt beschriebenen vervollständigungs-basierten Ansätzen zu verbinden, ist das Ziel des neuen, in Kapitel 4 beschriebenen Verfahrens.

## 3.6 Vervollständigungs-basierte $E$ -Unifikation

### 3.6.1 Grundlagen

Das Hauptproblem bei der Berechnung von  $E$ -Unifikatoren ist, wie schon gesagt, die Symmetrie der Gleichheit und die daraus resultierende Symmetrie der Ableitbarkeitsrelation  $\overset{*}{\leftrightarrow}_E$ .

Das Problem kann man lösen, indem man zu Reduktionsregeln — im Prinzip orientierte Gleichungen — übergeht, die eine nicht symmetrische Ableitbarkeitsrelation  $\overset{*}{\rightarrow}_R$  induzieren. Ist die Relation  $\overset{*}{\rightarrow}_R$  konfluent und wohlfundiert, definiert sie für jeden Term eine Normalform, die mit Hilfe der Reduktionsregeln relativ einfach berechenbar ist. Ist zudem der symmetrische Abschluß von  $\overset{*}{\rightarrow}_R$  gleich  $\overset{*}{\leftrightarrow}_E$ , läßt sich überprüfen, ob  $s \overset{*}{\leftrightarrow}_E t$  für beliebige Terme  $s$  und  $t$  gilt, denn dann haben  $s$  und  $t$  dieselbe Normalform.

<sup>14</sup> Die Unifikatoren sind nur aufzählbar.

Die Idee nun ist, ein gegebenes Reduktionssystem  $R$ , das die Anforderung  $\leftrightarrow_R^* = \leftrightarrow_E^*$  schon erfüllt, so zu transformieren — zu vervollständigen —, daß es konfluent wird. Dazu werden die Regeln in  $R$  in bestimmter Weise auf  $R$  selbst angewendet, und dabei auch neue Regeln erzeugt. Die Forderung, daß  $\rightarrow_R^*$  wohlfundiert sei, kann dadurch eingehalten werden, daß die Orientierung der Reduktionsregeln einer wohlfundierten Reduktionsordnung  $\succ$  (Def. 3.6.6) auf den Termen entspricht, so daß  $\rightarrow_R^*$  eine Teilrelation von  $\succ$  ist. Die verwendete Reduktionsordnung  $\succ$  hat damit entscheidenden Einfluß auf den Vervollständigungsprozeß; auf die Schwierigkeit,  $\succ$  günstig zu wählen, wird in Abschnitt 3.6.2 eingegangen.

Die Vervollständigung eines Reduktionssystem spiegelt sich in einer Transformation der mit seiner Hilfe durchführbaren Gleichheitsbeweise wieder. Durch die Vervollständigung können nämlich Peaks in diesen Beweisen, d. h. Teilbeweise der Form  $t_i \leftarrow_R t_{i+1} \rightarrow_R t_{i+2}$  beseitigt werden. Gleichheitsbeweise mit einem konfluenten Reduktionssystem enthalten keine Peaks mehr, sind also von der Form

$$t_0 \rightarrow_R \cdots \rightarrow_R t_k \leftarrow_R \cdots \leftarrow_R t_n .$$

Die Konstruktion von Reduktionssystemen zu *rein universellen* Gleichungssystemen ist eine häufig angewendete Methode. Eine kompakte Darstellung aller damit verbundenen Fragen und eine Übersicht über die umfangreiche Literatur ist in [Dershowitz & Jouannaud, 1990] enthalten. Kürzere Einführungen, insbesondere auch eine Beschreibung des wichtigsten Verfahrens, des *Unfailing Knuth–Bendix–Algorithmus* (siehe Abschnitt 3.6.4), geben [Dershowitz, 1989] und [Bachmair *et al.*, 1989].

Zunächst werden die in der Literatur zur Vervollständigung rein universeller Gleichungssysteme üblichen Begriffe in analoger Weise auf gemischte Gleichungssysteme übertragen:<sup>15</sup>

**Definition 3.6.1** (Reduktionsregel, Reduktionssystem)

Eine prädikatenlogische Formel der Form

$$(\forall x_1) \dots (\forall x_n)(s \rightsquigarrow t) ,$$

die all-quantifiziert sein kann bezüglich einiger oder aller der Variablen, die sie enthält, heie Reduktionsregel.

Eine Menge  $R$  von Reduktionsregeln heie Reduktionssystem.<sup>16</sup>

Das Prädikatensymbol  $\rightsquigarrow$  bezeichnet die Ableitbarkeitsrelation auf der Objektebene. Es steht in etwa dem gleichen Verhältnis zum Symbol  $\rightarrow$  der Meta-Ebene wie das Symbol  $\approx$  zur Meta-Gleichheit  $=$ .

**Definition 3.6.2** (Rein universelles, rein starres Reduktionssystem)

Ein Reduktionssystem  $R$ , d. h. eine Menge von Reduktionsregeln, heie

- *rein universell*, wenn es keine freien Variablen enthält,

<sup>15</sup> Im Zweifel seien alle Begriff, deren Bedeutung nicht explizit angegeben ist, wie in [Dershowitz & Jouannaud, 1990] definiert.

<sup>16</sup> Es gelte (ohne Beschränkung der Allgemeinheit)  $\text{Bound}(R) \cap \text{Free}(R) = \emptyset$ .

- *rein starr, wenn es keine gebundenen Variablen enthält.*

Für Reduktionsregeln gilt in entsprechender Weise alles das, was im vorigen Abschnitt über die Bezeichnungen „rein universell“ und „rein starr“ und über die Unterschiede zu der in der Literatur üblichen Schreibweise ohne Quantoren gesagt wurde. Wie die gemischten Gleichungssysteme stellen gemischte Reduktionssysteme eine Verallgemeinerung dar, die rein universelle und rein starre Systeme einschließt.

In der gleichen Weise, wie eine Menge  $E$  von Gleichungen eine symmetrische Ableitbarkeitsrelation  $\leftrightarrow_E$  induziert (Def. 3.2.1), induziert ein Reduktionssystem  $R$  eine nicht-symmetrische Ableitbarkeitsrelation  $\rightarrow_R$  (der einzige Unterschied ist, daß eine Regel nur von links nach rechts und nicht wie eine Gleichung in beide Richtungen angewendet werden kann):

**Definition 3.6.3** (Ableitbarkeitsrelation  $\rightarrow_R$ , Rechtfertigung, Beweis für  $s \xrightarrow{*}_R t$ )  
 Sei  $R$  eine Menge von Reduktionsregeln. Die Relation  $\rightarrow_R$  ist auf der Menge der Terme definiert durch:

Für zwei Terme  $s, t \in \mathbf{Term}$  gilt

$$s \rightarrow_R t$$

genau dann, wenn es eine Reduktionsregel

$$(\forall x_1) \dots (\forall x_n)(l \rightsquigarrow r) \in R ,$$

eine Position  $p$  in  $s$  und eine Substitution  $\sigma$  gibt, so daß

1.  $s|_p = l\sigma$ ,
2.  $t = (s[p/r])\sigma$ ,
3.  $\sigma$  belegt keine freien Variablen in  $s$  oder der Reduktionsregel.

In diesem Fall heißt das Tripel  $\langle (\forall x_1) \dots (\forall x_n)(l \rightsquigarrow r), p, \sigma \rangle$  eine Rechtfertigung für  $s \rightarrow_R t$ .

Eine endliche Folge  $\langle t_0, \dots, t_n \rangle$  ( $n \geq 0$ ) von Termen mit

$$t_0 \rightarrow_R t_1 \rightarrow_R t_2 \rightarrow_R \dots \rightarrow_R t_n$$

heiße ein  $R$ -Beweis für

$$t_0 \xrightarrow{*}_R t_n .$$

Die Eigenschaften der Relation  $\rightarrow_R$  werden auf das Reduktionssystem  $R$  übertragen:

**Definition 3.6.4** (Konfluenz, Church-Rosser-Eigenschaft, Wohlfundiertheit eines Reduktionssystems)

Ein Reduktionssystem  $R$

- *heiße wohlfundiert, konfluent, lokal konfluent,*

- habe die Church–Rosser–Eigenschaft,

wenn die Relation  $\xrightarrow{*}_R$  die entsprechende Eigenschaft hat.

Wie schon in der Einführung zu diesem Abschnitt angedeutet, kann ein  $E$ -Unifikationsproblem im Prinzip dann mit Hilfe eines Reduktionssystems  $R$  gelöst werden, wenn

1.  $R$  die gleiche *symmetrische* Ableitbarkeitsrelation wie  $E$  induziert, d. h., wenn  $\xrightarrow{*}_R = \xrightarrow{*}_E$ ,
2.  $R$  endlich, wohlfundiert und konfluent ist.

Zunächst kann man mit einem Reduktionssystem  $R$ , das diese Eigenschaften hat, aber nur entscheiden, ob zwei Terme gleich sind, und ob eine gegebene Substitution  $\sigma$  ein  $E$ -Unifikator ist. Es gibt aber auch Möglichkeiten, mit Hilfe eines wohlfundierten konfluenten Reduktionssystems  $E$ -Unifikatoren zu *berechnen*. Hierzu gehört insbesondere die Methode des *Narrowing*, die die Anwendung der Regeln in  $R$  mit syntaktischer Unifikation verbindet [Nutt *et al.*, 1989].

Das Berechnen vollständiger Mengen allgemeinsten Unifikatoren für gemischte  $E$ -Unifikationsprobleme ist aber auch damit zumindest problematisch. Man muß nämlich beachten, daß die Gleichungen in  $E\sigma$  für unterschiedliche Substitutionen  $\sigma$  unterschiedlich orientiert werden müssen, und es darum im allgemeinen kein Reduktionssystem  $R$  gibt, so daß  $R\sigma$  für alle  $\sigma$  eine Vervollständigung von  $E\sigma$  ist:

### Beispiel 3.6.5

Sei die Ordnung auf den Termen so gewählt, daß  $a \succ b$ . Dann muß die Gleichung  $x \approx y$  nach Anwendung der Substitution  $\sigma_1 = \{x/a, y/b\}$  von links nach rechts orientiert werden, mit  $\sigma_2 = \{x/b, y/a\}$  aber von rechts nach links.  $\square$

## 3.6.2 Reduktionsordnungen

Die übliche Methode sicherzustellen, daß die Ableitbarkeitsrelation  $\xrightarrow{*}_R$  wohlfundiert ist, ist zu zeigen, daß sie Teilrelation einer wohlfundierten (partiellen) Ordnung  $\succ$  auf den Termen ist. Dies ist zumindest immer dann der Fall, wenn

1. die linke Seite jeder Regel in  $R$  größer als ihre rechte Seite ist, d. h., wenn  $l \succ r$  für alle  $(\forall \bar{x})(l \rightsquigarrow r) \in R$  gilt,
2. die Ordnungsrelation  $\succ$  monoton ist bezüglich der Termstruktur und bezüglich Instantiierung.

Es genügt also für ein spezielles Reduktionssystem  $R$  den ersten Punkt zu überprüfen, wenn die Monotonie eine bekannte Eigenschaft der Ordnungsrelation  $\succ$  ist:

### Definition 3.6.6 ([Totale] Reduktionsordnung)

Eine strikte (partielle) Ordnung  $\succ$  auf den Termen heißt *Reduktionsordnung*, wenn sie wohlfundiert ist und monoton bezüglich der Termstruktur und Instantiierung, d. h., wenn für beliebige Terme  $s, t, u$ , Positionen  $p$  in  $u$  und Substitutionen  $\sigma$  gilt:

1. Wenn  $s \succ t$ , dann  $u[p/s] \succ u[p/t]$ .
2. Wenn  $s \succ t$ , dann  $s\sigma \succ t\sigma$ .

Eine Reduktionsordnung heißt *total*, wenn ihre Einschränkung auf die Menge der Grundtermen eine Totalordnung ist.

Eine Reduktionsordnung heißt also schon dann total, wenn sie eine Totalordnung auf den Grundtermen ist.

Ein wichtiges Beispiel für Reduktionsordnungen sind die lexikographischen Pfadordnungen [Dershowitz, 1987].

**Definition 3.6.7** (Lexikographische Pfadordnung)

$>_F$  sei eine strikte Ordnung auf den Funktionssymbolen. Dann ist die (von der Ordnung  $>_F$  induzierte) lexikographische Pfadordnung (LPO)

$$\succ_{\text{LPO}} \in \mathbf{Term} \times \mathbf{Term}$$

definiert durch:

Für beliebige Terme<sup>17</sup>

$$s = f(s_1, \dots, s_m) \quad \text{und} \quad t = g(t_1, \dots, t_n)$$

gilt  $s \succ_{\text{LPO}} t$  genau dann, wenn entweder

1. es ein  $1 \leq i \leq m$  gibt, so daß  $s_i \succ_{\text{LPO}} t$  oder  $s_i = t$ ,
2.  $f >_F g$  und  $s \succ_{\text{LPO}} t_j$  für alle  $1 \leq j \leq n$ , oder
3.  $f = g$ ,  $(s_1, \dots, s_n) \succ_{\text{LPO}} (t_1, \dots, t_n)$  und  $s \succ_{\text{LPO}} t_i$  für alle  $1 \leq i \leq n$ .

Dabei bezeichnet  $\succ_{\text{LPO}}$  die durch  $\succ_{\text{LPO}}$  lexikographisch induzierte Ordnung<sup>18</sup> auf Tupeln von Termen, d.h.,

$$\langle s_1, \dots, s_n \rangle \succ_{\text{LPO}} \langle t_1, \dots, t_n \rangle$$

gilt genau dann, wenn es ein  $1 \leq i \leq n$  gibt, so daß  $s_j = t_j$  für  $1 \leq j < i$  und  $s_i \succ_{\text{LPO}} t_i$ .

Diejenige LPO, die durch die alphabetische Ordnung auf den Konstanten- und Funktionssymbolen definiert wird, heie alphabetische LPO.

### Beispiel 3.6.8

Sei  $\succ_{\text{LPO}}$  die alphabetische LPO. Dann gilt:

$$\begin{array}{l} b \succ_{\text{LPO}} a \\ g(a) \succ_{\text{LPO}} f(b) \\ f(g(a)) \succ_{\text{LPO}} g(a) \\ h(b, a) \succ_{\text{LPO}} h(a, b) \end{array}$$

□

<sup>17</sup> Konstanten werden als nullstellige Funktionssymbole aufgefat.

<sup>18</sup> Das heit  $f >_F c >_F b >_F a$ ,  $\text{true} >_F \text{false}$  usw.

**Lemma 3.6.9**

Ist  $>_F$  eine Totalordnung, dann ist  $>_{\text{LPO}}$  eine totale Reduktionsordnung.

Die lexikographischen Pfadordnungen sind ein mächtiges Werkzeug. Mit ihrer Hilfe kann die Wohlfundiertheit vieler Reduktionssysteme bewiesen werden. Außerdem haben sie eine für das in Kapitel 4 beschriebene Verfahren wichtige Entscheidbarkeitseigenschaft (Satz 4.2.14), und sind darum auch für die Implementierung verwendet worden.

Eine Übersicht über andere Reduktionsordnungen und ihre Eigenschaften gibt [Dershowitz & Jouannaud, 1990].

**3.6.3 Konfluenz und Grundkonfluenz**

In vielen Fällen ist es nicht möglich, zu einem gegebenen Gleichungssystem  $E$  ein äquivalentes Reduktionssystem anzugeben, das konfluent ist. Es gibt Gleichungssysteme (auch rein universelle), für die kein solches Reduktionssystem existiert. Für andere ist es mit den vorhandenen Methoden nicht zu bestimmen. Man kann aber stets ein (u. U. unendliches) *grundkonfluentes* Reduktionssystem berechnen, das zu  $E$  äquivalent ist [Bachmair *et al.*, 1989].

**Definition 3.6.10** (Grundkonfluenz)

Sei  $\overset{*}{\rightarrow}_R$  die durch ein Reduktionssystem  $R$  induzierte Relation auf den Grundtermen.

Die Relation  $\overset{*}{\rightarrow}_R$  und das System  $R$  heißen *grundkonfluent*, wenn die Einschränkung von  $\overset{*}{\rightarrow}_R$  auf die Menge der Grundterme konfluent ist.

Es kann auch sein, daß zu einem Gleichungssystem  $E$  nur ein *unendliches* konfluentes Reduktionssystem existiert, es aber ein *endliches* grundkonfluentes System gibt. Eine ausführliche Darstellung der Gemeinsamkeiten und Unterschiede konfluenter und grundkonfluenter Reduktionssysteme ist in [Göbel, 1987] enthalten.

Wie der folgende Satz zeigt, genügt es für das automatische Beweisen völlig, wenn man sich auf grundkonfluente Reduktionssysteme beschränkt, da die verwendeten Signaturen in der Regel unendlich viele Konstantensymbole enthalten (nämlich die Skolem-Konstanten).

**Satz 3.6.11**

$R$  sei ein Reduktionssystem über der Signatur  $\mathcal{S}$ , und  $\mathcal{S}$  enthalte unendlich viele Konstantensymbole. Dann gilt:

$$R \text{ ist konfluent} \quad \text{gdw.} \quad R \text{ ist grundkonfluent} .$$

**Beweis:**

Siehe [Peterson, 1990]. ■



Auch für viele andere Anwendungen, wie beispielsweise Induktionsbeweise und Programmsynthese, ist Grundkonfluenz ausreichend.

Als Veranschaulichung für den Unterschied zwischen Grundkonfluenz und Konfluenz mag dienen, daß unter Umständen für den Übergang zwischen beiden ein Induktionsschluß bzw.  $\omega$ -Grenzübergang notwendig sein kann, um aus „für alle Grundterme  $t$  gilt  $f(t) \approx g(t)$ “ auf „es gilt  $(\forall x)(f(x) \approx g(x))$ “ zu schließen.

### Beispiel 3.6.12

Hier ein Beispiel für ein grundkonfluentes aber nicht konfluentes Reduktionssystem:<sup>19</sup>

- (1)  $(\forall y)(add(0, y) \rightsquigarrow y)$
- (2)  $(\forall x)(\forall y)(add(s(x), y) \rightsquigarrow s(add(x, y)))$
- (3)  $(\forall u)(\forall v)(\forall w)(add(u, add(v, w)) \rightsquigarrow add(add(u, v), w))$

Mit Hilfe dieses Reduktionssystems können alle Grundterme auf eine Normalform  $s^j(0)$  ( $j \geq 0$ ) gebracht werden. Die Umformung beruht auf einer „Induktion“ über das erste Argument eines  $add$ . Steht dort jedoch kein Grundterm, kann die 0 nicht erreicht werden.

Das Reduktionssystem ist beispielsweise nicht konfluent für den Term

$$add(u, add(s(v), s(w))) ,$$

denn

$$\begin{aligned} add(u, add(s(v), s(w))) &\rightarrow_{(2)} add(u, s(add(v, s(w)))) \\ add(u, add(s(v), s(w))) &\rightarrow_{(3)} add(add(u, s(v)), w) . \end{aligned}$$

Auf beide neuen Terme lassen sich keine Regeln mehr anwenden. □

### 3.6.4 Der Unfailing Knuth–Bendix–Algorithmus

Der Unfailing Knuth–Bendix–Algorithmus (UKBA) ist eine Erweiterung des von D. E. Knuth und P. B. Bendix [1970] entwickelten Verfahrens zur Vervollständigung rein universeller Reduktionssysteme. In Kombination mit Narrowing ist es das wichtigste und ein häufig implementiertes Verfahren für die rein universelle  $E$ -Unifikation. Eine Einführung und eine Übersicht über die umfangreiche Literatur zu diesem Verfahren gibt [Bachmair *et al.*, 1989].

Die Konstruktion eines konfluenten und wohlfundierten Reduktionssystems zu einer Gleichungsmenge  $E$  mit Hilfe des klassischen Knuth–Bendix–Algorithmus kann scheitern, obwohl ein solches Reduktionssystem existiert, — etwa weil die Reduktionsordnung ungünstig gewählt ist. Dieses Problem wird durch die Weiterentwicklung zum UKBA gelöst. Mit seiner Hilfe kann zu jedem Gleichungssystem ein grundkonfluentes (Def. 3.6.10), jedoch nicht notwendig konfluentes Reduktionssystem berechnet werden — sofern eines existiert.

<sup>19</sup> Das Beispiel stammt ursprünglich von L. Fribourg. Es ist zitiert in [Göbel, 1987].

Die im folgenden verwendete — inzwischen in der Literatur übliche — Transformationsregel-Schreibweise<sup>20</sup> für Vervollständigungsverfahren geht auf [Dershowitz, 1987] zurück.

**Definition 3.6.13** (Unfailing Knuth–Bendix–Algorithmus)

Seien  $\langle E, R \rangle$  und  $\langle E', R' \rangle$  Paare rein universeller Mengen von Gleichungen und rein universeller Reduktionssysteme.  $\succ$  sei eine totale Reduktionsordnung.<sup>21</sup> Es gilt

$$\langle E, R \rangle \mid_{\text{UKBA}} \langle E', R' \rangle$$

genau dann, wenn  $\langle E', R' \rangle$  aus  $\langle E, R \rangle$  hervorgeht durch Anwendung einer der in Tabelle 3.3 dargestellten Transformationsregeln des Unfailing Knuth–Bendix–Algorithmus (Orient, Delete, Deduce, Simplify 1, Simplify 2, Compose 1, Compose 2, Collapse 1, Collapse 2).<sup>22</sup> Dabei bedeutet  $s \rightarrow_{E, \succ} t$ , daß es eine Rechtfertigung  $\langle (\forall \bar{x})(l \approx r), p, \sigma \rangle$  für  $s \rightarrow_E t$  gibt, wobei zwar die Gleichung  $l \approx r$  nicht unbedingt orientierbar ist, wohl aber ihre Instanz  $(l \approx r)\sigma$  — nämlich zu  $l\sigma \succ r\sigma$ , was  $s \succ t$  impliziert.

Die Menge  $EP_{\succ}(\langle E, R \rangle)$  der erweiterten kritischen Paare in Tabelle 3.3 ist definiert als die Menge aller Gleichungen  $(\forall \bar{x})(\forall \bar{y})(t\sigma \approx (s[p/r]\sigma)$ , so daß

- $(\forall \bar{x})(s \rightsquigarrow t) \in R, (\forall \bar{x})(s \approx t) \in E$  oder  $(\forall \bar{x})(t \approx s) \in E$ ;
- $(\forall \bar{y})(l \rightsquigarrow r) \in R, (\forall \bar{y})(l \approx r) \in E$  oder  $(\forall \bar{y})(r \approx l) \in E$ ;
- $p$  ist eine Position in  $s$  und  $s|_p$  ist keine Variable;
- $s|_p$  und  $l$  sind unifizierbar mit MGU  $\sigma$ ;
- $s\sigma \not\approx t\sigma$ ;
- $l\sigma \not\approx r\sigma$ .

Die wichtigste Regel ist die Deduktionsregel. Sie erzeugt aus einem kritischen Paar, d. h. einem Paar von Regeln, die dazu führen können, daß die Ableitbarkeitsrelation nicht konfluent ist, eine neue Regel, die die lokale Konfluenz und damit wegen der Wohlfundiertheit die Konfluenz wiederherstellt.

**Beispiel 3.6.14**

Ist  $R = \{a \rightsquigarrow b, a \rightsquigarrow c\}$  und  $E$  leer, dann gilt  $a \rightarrow_{\langle E, R \rangle} b$  und  $a \rightarrow_{\langle E, R \rangle} c$ , aber es gibt keinen Term  $t$ , so daß  $b \xrightarrow{\ast}_{\langle E, R \rangle} t$  und  $c \xrightarrow{\ast}_{\langle E, R \rangle} t$ .<sup>23</sup>

Da aber die beiden Regeln ein kritisches Paar bilden, wird eine neue Gleichung  $b \approx c$  erzeugt, und mit  $\langle E', R' \rangle = \langle E \cup \{b \approx c\}, R \rangle$  gilt  $b \rightarrow_{\langle E', R' \rangle} c$ .  $\square$

<sup>20</sup> Dabei ist die unterhalb der Linie stehende Struktur aus der oberhalb stehenden ableitbar, falls die rechts stehenden Bedingungen erfüllt sind.

<sup>21</sup> Es genügt auch, wenn  $\succ$  zu einer totalen Reduktionsordnung erweitert werden kann.

<sup>22</sup> Vor der Anwendung einer Regel sind alle auftretenden Variablen durch gebundene Umbenennung durch neue zu ersetzen.

<sup>23</sup>  $\rightarrow_{\langle E, R \rangle}$  ist die Vereinigung von  $\rightarrow_E$  und  $\rightarrow_R$ .

(Ori)	$\frac{\langle E \cup \{(\forall \bar{x})(s \approx t)\}, R \rangle}{\langle E, R \cup \{(\forall \bar{x})(s \rightsquigarrow t)\}\rangle}$	$s \succ t$
(Del)	$\frac{\langle E \cup \{(\forall \bar{x})(s \approx s)\}, R \rangle}{\langle E, R \rangle}$	
(Ded)	$\frac{\langle E, R \rangle}{\langle E \cup \{(\forall \bar{x})(s \approx t)\}, R \rangle}$	$(\forall \bar{x})(s \approx t) \in EP_{\succ}(\langle E, R \rangle)$
(Si1)	$\frac{\langle E \cup \{(\forall \bar{x})(s \approx t)\}, R \rangle}{\langle E \cup \{(\forall \bar{x})(\forall \bar{y})(u \approx t)\}, R \rangle}$	$s \rightarrow_R u$ mit $(\forall \bar{y})(l \rightsquigarrow r) \in R$
(Si2)	$\frac{\langle E \cup \{(\forall \bar{x})(s \approx t)\}, R \rangle}{\langle E \cup \{(\forall \bar{x})(\forall \bar{y})(u \approx t)\}, R \rangle}$	$s \rightarrow_{E_{\succ}} u$ mit $(\forall \bar{y})(l \approx r) \in E_{\succ}$ , wobei $s \triangleright l$
(Cm1)	$\frac{\langle E, R \cup \{(\forall \bar{x})(s \rightsquigarrow t)\}\rangle}{\langle E, R \cup \{(\forall \bar{x})(\forall \bar{y})(s \rightsquigarrow u)\}\rangle}$	$t \rightarrow_R u$ mit $(\forall \bar{y})(l \rightsquigarrow r) \in R$
(Cm2)	$\frac{\langle E, R \cup \{(\forall \bar{x})(s \rightsquigarrow t)\}\rangle}{\langle E, R \cup \{(\forall \bar{x})(\forall \bar{y})(s \rightsquigarrow u)\}\rangle}$	$t \rightarrow_{E_{\succ}} u$ mit $(\forall \bar{y})(l \approx r) \in E_{\succ}$
(Cl1)	$\frac{\langle E, R \cup \{(\forall \bar{x})(s \rightsquigarrow t)\}\rangle}{\langle E \cup \{(\forall \bar{x})(\forall \bar{y})(v \approx t)\}, R \rangle}$	$s \rightarrow_R v$ mit $(\forall \bar{y})(l \rightsquigarrow r) \in R$ , wobei $s \triangleright l$
(Cl2)	$\frac{\langle E, R \cup \{(\forall \bar{x})(s \rightsquigarrow t)\}\rangle}{\langle E \cup \{(\forall \bar{x})(\forall \bar{y})(v \approx t)\}, R \rangle}$	$s \rightarrow_{E_{\succ}} v$ mit $(\forall \bar{y})(l \approx r) \in E_{\succ}$ , wobei $s \triangleright l$

**Tabelle 3.3:** Die Transformationsregeln des Unfailing Knuth–Bendix–Algorithmus (Def. 3.6.13).

Als das Resultat der Vervollständigung durch den UKBA, das heißt der iterativen Anwendung der Transformationsregeln aus Definition 3.6.13 auf initiales Paar  $\langle E^0, R^0 \rangle$  definiert man nun das Paar  $\langle E^\infty, R^\infty \rangle$  in folgender Weise:

**Definition 3.6.15** (Vervollständigung)

Für eine (endliche oder unendliche) Folge  $(\langle E^i, R^i \rangle)_{i \geq 0}$  von Paaren von Gleichungssystemen  $E_i$  und Reduktionssystemen  $R_i$  gelte

$$\langle E^0, R^0 \rangle \vdash_{\text{UKBA}} \langle E^1, R^1 \rangle \vdash_{\text{UKBA}} \langle E^2, R^2 \rangle \vdash_{\text{UKBA}} \cdots$$

Dann heiÙe das Paar  $\langle E^\infty, R^\infty \rangle$ , das definiert ist durch

$$E^\infty = \begin{cases} E^m & \text{falls die Folge der } \langle E^i, R^i \rangle \text{ von der Länge } m \text{ ist} \\ \bigcup_{k \geq 0} \bigcap_{m \geq k} E^m & \text{falls die Folge unendlich ist} \end{cases}$$

und

$$R^\infty = \begin{cases} R^m & \text{falls die Folge der } \langle E^i, R^i \rangle \text{ von der Länge } m \text{ ist} \\ \bigcup_{k \geq 0} \bigcap_{m \geq k} R^m & \text{falls die Folge unendlich ist} \end{cases}$$

die Vervollständigung von  $\langle E^0, R^0 \rangle$ .

Falls die Folge der  $\langle E^i, R^i \rangle$  unendlich ist, enthält  $\langle E^\infty, R^\infty \rangle$  also die *persistenten* Gleichungen und Reduktionsregeln, die in allen bis auf endlich vielen der  $\langle E^i, R^i \rangle$  enthalten sind.

Den zusätzliche Freiheitsgrad, den man durch die Formulierung mit Transformationsregeln hat, nämlich diese Regeln in beliebiger Reihenfolge anzuwenden, um  $\langle E^\infty, R^\infty \rangle$  zu erzeugen, muß man nun wieder etwas einschränken. Die Anwendung muß nämlich in der Weise fair sein, daß früher oder später auf jedes persistente kritische Paar die Regel *Deduce* angewendet wird:

**Definition 3.6.16** (Fairneß für den UKBA)

*Eine Ableitungsfolge*

$$\langle E^0, R^0 \rangle \vdash_{\text{UKBA}} \langle E^1, R^1 \rangle \vdash_{\text{UKBA}} \langle E^2, R^2 \rangle \vdash_{\text{UKBA}} \dots$$

heißt fair, wenn

$$EP_{\succ}(\langle E^\infty, R^\infty \rangle) \subset \bigcup_{i \geq 0} E^i .$$

Wird diese Fairneßbedingung eingehalten, ist der UKBA in folgendem Sinne korrekt und vollständig [Bachmair *et al.*, 1989]:

**Satz 3.6.17**

*Ist  $E$  ein rein universelles Gleichungssystem,  $\langle E^0, R^0 \rangle = \langle E, \emptyset \rangle$ , und die Ableitungsfolge*

$$\langle E^0, R^0 \rangle \vdash_{\text{UKBA}} \langle E^1, R^1 \rangle \vdash_{\text{UKBA}} \langle E^2, R^2 \rangle \vdash_{\text{UKBA}} \dots$$

*fair im Sinne von Definition 3.6.16, dann gilt:*

Korrektheit:  $\xrightarrow{*}_{\langle E^\infty, R^\infty \rangle} = \xleftrightarrow{*}_{E^\infty} \cup \xrightarrow{*}_{R^\infty}$  ist gleich  $\xleftrightarrow{*}_{E^0}$  und grundkonfluent.

Vollständigkeit: *Unter der Voraussetzung, daß zu  $E$  ein grundkonfluentes und wohlfundiertes Reduktionssystem  $R$  mit  $\xrightarrow{*}_R = \xleftrightarrow{*}_E$  existiert, ist auch  $R^\infty$  grundkonfluent und wohlfundiert, und es ist  $E^\infty = \emptyset$ .*

Ein Beispiel für eine leistungsfähige Implementierung des UKBA ist der Theorembeweiser SbREVE [Hsiang & Mzali, 1988]. Eine Übersicht über dieses und andere den UKBA verwendende Systeme gibt [Herrman *et al.*, 1991]. Mit solchen Systemen sind Vervollständigungen vieler spezieller Gleichungssysteme  $E$  bestimmt worden.

### 3.6.5 Das Verfahren von Gallier et al.

In [Gallier *et al.*, 1992] ist ein vervollständigungsbasiertes Verfahren für die rein starre  $E$ -Unifikation (*rigid  $E$ -unification*) beschrieben. Als Motivation für die Betrachtung des Problems dient dort die Erweiterung der Konnektionsmethode (*method of matings*) um die Behandlung von Gleichheit. Ähnlich wie die Gleichheitsbehandlung im Tableaurekalkül mit freien Variablen läßt sie sich auf das Problem der rein starren<sup>24</sup>  $E$ -Unifikation zurückführen (siehe Kapitel 5).

<sup>24</sup> Für die Gleichheitsbehandlung im Tableaurekalkül mit freien Variablen und universellen Formeln (siehe Abschnitt 5.1.3) müssen simultane gemischte  $E$ -Unifikationsprobleme gelöst werden; beim Tableaurekalkül ohne universelle Formeln genügt — wie bei der Konnektionsmethode — die Lösung simultaner rein starrer Probleme.

[Gallier *et al.*, 1992] schließt eine Reihe von Veröffentlichungen der Autoren zum Thema starre  $E$ -Unifikation ab und stellt die darin enthaltenen Ergebnisse ausführlich dar. Zum ersten Mal formuliert wurde das Problem der rein starren  $E$ -Unifikation in [Gallier *et al.*, 1987] und die Vermutung geäußert, es sei im Gegensatz zur rein universellen  $E$ -Unifikation entscheidbar. Bewiesen wurde die Entscheidbarkeit der rein starren  $E$ -Unifikation — genauer gesagt: ihre NP-Vollständigkeit<sup>25</sup> — in [Gallier *et al.*, 1988]. Ausführlicher ist der Beweis in [Gallier *et al.*, 1990] dargestellt. Dieser Beweis ist konstruktiv, d. h. es wird tatsächlich ein (nicht-deterministisches) Verfahren angegeben, mit dessen Hilfe eine bezüglich  $\leq_E$  vollständige Menge von Lösungen eines rein starren Problems  $\langle E, s, t \rangle$  berechnet werden kann.

Dazu wird das Problem zunächst durch ein äquivalentes Problem  $\langle E', s', t' \rangle$  ersetzt, bei dem  $s'$  und  $t'$  Grundterme sind (der folgende Satz ist das Analogon für rein starre Probleme zu Satz 3.5.4):

**Satz 3.6.18**

*Eine Substitution  $\sigma$  ist genau dann Lösung eines rein starren  $E$ -Unifikationsproblems  $\langle E, s, t \rangle$ , wenn sie Lösung des rein starren Problems*

$$\langle E', s', t' \rangle = \langle E \cup \{eq(x, x) \approx \text{true}, eq(s, t) \approx \text{false}\}, \text{true}, \text{false} \rangle$$

*ist. Dabei sind true und false neue Konstanten, eq ist ein neues Prädikatensymbol und x eine neue Variable.*

Anschließend werden, ausgehend von  $E'$  iterativ folgende Schritte ausgeführt (die hier allerdings nur vereinfacht beschrieben sind):

1. Eine *order assignment* genannte Ordnung auf den in  $E'$  auftretenden Termen wird gewählt, die es erlaubt,  $E'$  unter der Voraussetzung zu vervollständigen, daß man die auftretenden Variablen als Konstanten, also als unveränderlich bzw. *starr* auffaßt. Es ist ein wesentlicher Punkt des Beweises von Gallier *et al.*, daß eine solche Ordnung auf den Termen immer gewählt werden kann. Gibt es mehrere geeignete *order assignments*, wird indeterministisch eine beliebige ausgewählt.
2. Dann wird eine Vervollständigung  $R'$  für dieses starre System  $E'$  konstruiert. Dafür wird ein effizientes, auf dem Kongruenzabschluß basierendes Verfahren zur Vervollständigung von Systeme ohne Variablen verwendet.
3. Enthält die Vervollständigung  $R'$  — als nicht starres Problem betrachtet — noch kritische Paare, wird ein solches kritisches Paar (indeterministisch) ausgewählt, und der Unifikator  $\mu$ , der bei der Bildung des kritischen Paares auftritt, wird auf die gesamte Vervollständigung  $R'$  angewendet. Dann wird das Gleichungssystem  $E'$  durch  $R'\mu$  (einschließlich der durch das kritische Paar neu entstandenen Gleichung) ersetzt und wieder mit dem ersten Schritt begonnen. Das Verfahren terminiert, wenn  $R'$  kein kritisches Paar mehr enthält.

<sup>25</sup> Es wird auch bewiesen, daß das Problem NP-hart ist.

Terminiert das Verfahren, enthält  $R'$  also keine kritischen Paare mehr, und gilt  $\text{false} \stackrel{*}{\leftrightarrow}_R \text{true}$ , dann ist die Vereinigung der bis dahin angewendeten Unifikatoren  $\mu$  eine Lösung des ursprünglichen Unifikationsproblems. Das Verfahren muß terminieren, weil das System nach jeder Bildung eines kritischen Paares wenigstens eine Variable weniger enthält.

Die Entscheidbarkeit der rein starren  $E$ -Unifikation ist ein interessantes und sogar etwas überraschendes Ergebnis, weil der Unterschied zur rein universellen  $E$ -Unifikation klein zu sein scheint, insbesondere die Einschränkung der Ressourcen, die man bei Gleichheitsbeweisen zur Verfügung hat, erscheint zunächst unwesentlich, ist aber, wie der Satz von Gallier et al. zeigt, entscheidend.

[Gallier et al., 1992] enthält auch den Satz, daß das *simultane* rein starre  $E$ -Unifikationsproblem entscheidbar sei. Der Beweis dieses Satzes ist aber etwas kurz, und beruht im wesentlichen auf Analogieschlüssen zum Einzelproblem. Es sind auch verschiedentlich Zweifel an der Korrektheit des Beweises und der Gültigkeit des Satzes geäußert worden.<sup>26</sup>

Die Frage der Entscheidbarkeit simultaner Probleme ist aber für die Frage, in wie weit sich das Verfahren auf die *gemischte*  $E$ -Unifikation übertragen läßt unerheblich, da diese ohnehin unentscheidbar ist.

Der wesentlich Nachteil des Verfahrens ist sein Indeterminismus. Es bleibt völlig unklar, wie dieser anders als durch Backtracking zu beseitigen wäre. Das ist wegen der großen Zahl von *choice points* äußerst ungünstig, und das Verfahren erscheint — solange keine guten Heuristiken für die Auswahl kritischer Paare zur Verfügung stehen — für eine Implementierung ungeeignet (und ist tatsächlich nie implementiert worden<sup>27</sup>).

Dennoch sind einige der dem Verfahren zu Grunde liegenden Ideen für das neue, in Kapitel 4 beschriebene Verfahren übernommen worden.

### Beispiel 3.6.19

Hier ein Beispiel für die Anwendung des Verfahrens von Gallier et al.: Gegeben sei das rein starre  $E$ -Unifikationsproblem

$$\langle E, s, t \rangle = \langle \{f(a) \approx a, g(g(x)) \approx f(a)\}, g(g(g(x))), x \rangle .$$

Dann ist

$$E' = \{f(a) \approx a, g(g(x)) \approx f(a), eq(g(g(g(x))), x) \approx \text{false}, eq(y, y) \approx \text{true}\} .$$

Nach Wahl eines geeigneten *order assignments* und Durchführung des ersten Vervollständigungs-schrittes erhält man

$$R'_1 = \{f(a) \rightsquigarrow a, g(g(x)) \rightsquigarrow a, eq(g(a), x) \rightsquigarrow \text{false}, eq(y, y) \rightsquigarrow \text{true}\} .$$

$R'_1$  enthält das kritische Paar

$$eq(g(a), x) \rightsquigarrow \text{false} \quad \text{und} \quad eq(y, y) \rightsquigarrow \text{true} .$$

<sup>26</sup> Unter anderem von U. Petermann und J. Goubault. Auch P. Narendran räumte ein, daß man den Beweis zumindest noch einmal überprüfen mußte.

<sup>27</sup> Nach Auskunft von P. Narendran.

Der bei der Bildung des Paares auftretende Unifikator ist  $\mu = \{x/g(a), y/g(a)\}$ . Wendet man  $\mu$  auf  $R'$  an, fügt die neue Gleichung  $\text{true} \approx \text{false}$  hinzu und führt wieder einen Vervollständigungsschritt aus, ergibt dies

$$R'_2 = \{f(a) \rightsquigarrow a, g(g(g(a))) \rightsquigarrow a, \text{eq}(g(a), g(a)) \rightsquigarrow \text{true}, \text{false} \rightsquigarrow \text{true}\} .$$

Da  $R_2$  kein kritisches Paar mehr enthält, terminiert das Verfahren.  $\text{false} \rightsquigarrow \text{true}$  ist in  $R_2$  enthalten, und man kann schließen, daß die Vereinigung aller bisher angewendeten Substitutionen — also  $\mu = \{x/g(a), y/g(a)\}$  — ein  $E$ -Unifikator von  $\langle E, s, t \rangle$  ist.  $\square$

### 3.6.6 Das Verfahren von Goubault

Ein weiteres Verfahren für die rein starre  $E$ -Unifikation ist in [Goubault, 1993] beschrieben. Es ist nicht wie das in [Gallier *et al.*, 1992] vervollständigungs-basiert, sondern es basiert völlig auf einem „trickreichen“ Kongruenzabschluß. Auch dieses Verfahren ist indeterministisch und darum schwer zu implementieren.

In [Goubault, 1993] heißt es, die Methode sei effizienter als die von Gallier *et al.* (was jedoch nicht belegt wird), auch sei sie mächtiger, da man beliebige Auswahlheuristiken hinzufügen kann, die die Suche steuern, und es u. a. eine Auswahlheuristik gibt, die dazu führt, daß das Verfahren die Vervollständigung nach Gallier *et al.* simuliert.

Wie bei dem Verfahren von Gallier *et al.* ist aber auch bei dem von Goubault völlig unklar, wie man es für die gemischten  $E$ -Unifikation erweitern könnte.

# 4 Ein vervollständigungsbasiertes Verfahren für gemischte E–Unifikation

---

*It is obvious that the sign =  
is really the wrong sign for such relations,  
because it suggests symmetry,  
and there is no such symmetry.*

— N. G. DE BRUIJN, 1958

## 4.1 Einführung

In diesem Kapitel wird ein neues Verfahren für die gemischte  $E$ -Unifikation entwickelt. Es vereinigt in sich die Vorteile anderer Methoden zur Lösung von  $E$ -Unifikationsproblemen:

- Das Verfahren ist vervollständigungsbasiert. Es kann als eine Erweiterung des (nur für die rein universelle  $E$ -Unifikation geeigneten) Unfailing Knuth–Bendix–Algorithmus mit Narrowing gesehen werden. Allerdings sind tiefgreifende Änderungen notwendig, um gemischte Probleme behandeln zu können.
- Es ist — wie das Verfahren in [Beckert, 1991] — in dem Sinne deterministisch, daß bei der Suche nach einem Unifikator niemals Backtracking notwendig wird. Neue Reduktionsregeln und abgeleitete Terme müssen nie wieder verworfen und einmal angewendete Substitutionen nicht wieder zurückgenommen werden.
- Betrachtet man einen Suchbaum, dessen Pfade den verschiedenen Ableitungen zulassenden Substitutionen entsprechen, und in dem immer dann eine Verzweigung auftritt, wenn aus einer Reduktionsregel oder einem Term mit verschiedenen Substitutionen verschiedene neue Regeln oder Terme abgeleitet werden können, dann führt das Verfahren in diesem Suchbaum eine Breitensuche aus. Die Erfahrung lehrt, und in [Beckert & Hähnle, 1992] ist das auch näher begründet, daß eine Breitensuche bezüglich der Substitutionen einer Tiefensuche fast immer überlegen und für eine effiziente Implementierung unerlässlich ist.
- Die zu unifizierenden Terme gehen im Gegensatz zu dem Verfahren aus [Gallier *et al.*, 1992] *nicht* in die Berechnung der Vervollständigung ein. Das ist für die



Gleichheitsbehandlung im Tableaukalkül wichtig, denn dort treten in der Regel viele  $E$ -Unifikationsprobleme mit der selben Gleichungsmenge  $E$  auf, die sich nur in den zu unifizierenden Termen unterscheiden (siehe Abschnitt 5.2.3).

Die grundlegende Idee des neuen Verfahrens ist, bei gegebener Gleichungsmenge  $E$  die für verschiedene Substitutionen  $\sigma$  unterschiedlichen Mengen  $E\sigma$  durch ein Reduktionssystem darzustellen, wobei diese Darstellung möglichst frei von Redundanzen sein soll. Dazu werden an die Reduktionsregeln *Constraints* (Bedingungen) angeheftet, die beschreiben, für welche  $\sigma$  die Regeln Gültigkeit haben, d. h. für welche  $\sigma$  sie in der Vervollständigung von  $E\sigma$  enthalten sind; diese Constraints werden auch an abgeleitete Terme angeheftet und geben dann an, für welche  $\sigma$  er aus einem gegebenen Term ableitbar ist.

Auch schon bei der Methode aus [Beckert, 1991] werden abgeleitete Terme mit Bedingungen versehen, nämlich in Form der an die Elemente  $t'_\sigma$  der Mengen  $\langle t \rangle_E$  angehefteten Substitutionen  $\sigma$ .

Bei Reduktionsregeln genügt es aber nicht, Substitutionen als Constraints zu verwenden, sondern auch über die durch die Reduktionsordnung gegebene Relation zwischen den Termen, die für die freien Variablen eingesetzt werden, muß man in den Constraints Aussagen machen können.

#### Beispiel 4.1.1

Sei beispielsweise  $E = \{x \approx y\}$ . Zu einem zu  $E\sigma$  äquivalenten Reduktionssystem wird — je nachdem welche der Variablen  $x$  und  $y$  durch  $\sigma$  mit dem (bzgl. der Reduktionsordnung) größeren Term belegt wird — entweder die Regel  $(x \rightsquigarrow y)\sigma$  oder  $(y \rightsquigarrow x)\sigma$  gehören.

Die Bedingung, daß die Regel  $x \rightsquigarrow y$  (bzw. eine ihrer Instanzen) nur dann Gültigkeit haben soll, wenn  $\sigma(x) \succ \sigma(y)$ , läßt sich durch Substitutionen nicht ausdrücken — allenfalls dadurch, daß man alle Substitutionen  $\sigma$  aufzählte, die diese Bedingung einhalten, was aber völlig unpraktikabel ist.

Die — nicht nur in diesem Beispiel — naheliegende Lösung ist, einen Ausdruck der Form  $x \succ y$  als Bedingung zu verwenden.  $\square$

In den folgenden Abschnitten wird zunächst die allgemeine Form der verwendeten Constraints beschrieben, ihre Eigenschaften und ihre Handhabung wird diskutiert. In Abschnitt 4.4 werden die Transformationsregeln angegeben, die es erlauben, zu einem Gleichungssystem  $E$  ein System von Reduktionsregeln mit Constraint zu erzeugen, das für jede Substitution  $\sigma$  ein zu  $E\sigma$  äquivalentes Reduktionssystem „beinhaltet“; Sätze über die Eigenschaften des Verfahrens, insbesondere Korrektheit und Vollständigkeit, werden bewiesen. Es wird beschrieben, wie mit Hilfe eines solchen Reduktionssystems mit Constraints ein gemischtes  $E$ -Unifikationsproblem zu lösen ist, und wie eine bezüglich der Subsumtionsrelation  $\leq_E$  auf den Substitutionen vollständige Menge allgemeinsten Unifikatoren berechnet werden kann; und schließlich wird das Problem behandelt, wie eine bezüglich der Relation  $\leq$  vollständige Menge von MGUs erzeugt und mit ihrer Hilfe auch ein simultanes gemischtes  $E$ -Unifikationsproblem gelöst werden kann.

## 4.2 Ordnungsbedingungen und Constraints

Die an Regeln und Terme angehefteten Constraints bestehen aus zwei Teilen. Ein Teil ist — wie Beispiel 4.1.1 nahelegt — eine „Ordnungsbedingung“<sup>1</sup>, einer Aussage über die Anordnung von Termen bezüglich der zugrunde gelegten Reduktionsordnung.

**Definition 4.2.1** ([Atomare] Ordnungsbedingung, Grund-Ordnungsbedingung)  
Die atomaren Formeln

1. *true und false,*
2.  $s \succ t$  für beliebige Terme  $s, t \in \mathbf{Term}$

heißen *atomare Ordnungsbedingungen*.

Die mit Hilfe der Operatoren  $\neg, \wedge, \vee$  und  $\supset$  aus den atomaren Ordnungsbedingungen aufbaubaren logischen Formeln heißen *Ordnungsbedingungen*.

Eine Ordnungsbedingung, die keine Variablen enthält, heiße *Grund-Ordnungsbedingung*.

Den Grund-Ordnungsbedingungen kann in folgender Weise ein Wahrheitswert zugewiesen werden:

**Definition 4.2.2** (Wahrheitswert einer Grund-Ordnungsbedingung)

Der Wahrheitswert einer Grund-Ordnungsbedingung  $O$  sei dadurch gegeben, daß

- *zunächst den in  $O$  enthaltenen atomaren Ordnungsbedingungen der Wahrheitswert zugewiesen wird, den sie haben, wenn das (Prädikaten-)Symbol  $\succ$  durch eine totale Reduktionsordnung  $\succ_x$  auf den Termen interpretiert wird;*
- *die logischen Operatoren in der üblichen Weise interpretiert werden, um den Wahrheitswert von  $O$  zu bestimmen.*

Entsprechend dem ihr zugewiesenen Wahrheitswert heiße eine Grund-Ordnungsbedingung *wahr oder falsch*.

Die Ordnungsbedingungen bedürfen also der Interpretation durch eine auf den Termen definierte totale Reduktionsordnung  $\succ_x$  (Def. 3.6.6). Da diese eine Totalordnung auf den Grundtermen ist, ist der Wahrheitswert der atomaren Grund-Ordnungsbedingungen und damit auch derjenige allgemeiner Grund-Ordnungsbedingungen wohldefiniert.

### Beispiel 4.2.3

Die Grund-Ordnungsbedingungen  $f(a) \succ a$  und  $(a \succ b) \vee (b \succ a)$  sind wahr.  $\square$

<sup>1</sup> In [Peterson, 1990] werden Ordnungsbedingungen in ähnlicher Weise definiert. Dort werden sie allerdings als *constraints* bezeichnet (die wiederum nicht mit den Constraints aus Definition 4.2.6 zu verwechseln sind.)

Im folgenden wird sich zeigen, daß es nur sinnvoll ist, solche Reduktionsordnungen  $\succ_x$  zu verwenden, die gewisse (in Definition 4.2.13 präzisierte) Berechenbarkeitseigenschaften haben (insbesondere muß entscheidbar sein, ob die mit den Ordnungsbedingungen aufgebauten Constraints gemäß Definition 4.2.7 erfüllbar sind). Die lexikographischen Pfadordnungen  $\succ_{\text{LPO}}$  (Def. 3.6.7) beispielsweise haben diese Eigenschaften. Sie sind für die Implementierung des Verfahrens verwendet worden, und man kann sich im folgenden stets das Symbol  $\succ$  in den Ordnungsbedingungen als durch eine LPO  $\succ_{\text{LPO}} = \succ_x$  interpretiert denken.

Im Gegensatz zu Grund-Ordnungsbedingungen kann beliebigen Ordnungsbedingungen nicht immer ein Wahrheitswert zugewiesen werden. Sie können (ähnlich wie Formeln der Prädikatenlogik) in Abhängigkeit von der Belegung der Variablen, die sie enthalten, verschiedene Wahrheitswerte haben. Es liegt nahe, Ordnungsbedingungen mit wechselnden Wahrheitswerten als erfüllbar oder konsistent zu bezeichnen — diese Begriffe sind für Ordnungsbedingungen jedoch nicht modelltheoretisch definiert:

**Definition 4.2.4** (Wahre, falsche, konsistente Ordnungsbedingung, Äquivalenz von Ordnungsbedingungen)

*Eine Ordnungsbedingung  $O$  heie*

- *wahr, wenn für alle Grundsubstitutionen  $\sigma$  die Grund-Ordnungsbedingung  $O\sigma$  wahr ist;*
- *falsch (oder inkonsistent), wenn ihre Negation  $\neg O$  wahr ist;*
- *konsistent, wenn sie nicht falsch ist.*

*Zwei Ordnungsbedingungen  $O_1$  und  $O_2$  heißen äquivalent, wenn*

$$(O_1 \supset O_2) \wedge (O_2 \supset O_1)$$

*wahr ist.*

Entsprechend dieser Definition ist jede konsistente (und also nicht falsche) Grund-Ordnungsbedingung wahr.

Nun ist es also möglich, Aussagen über die Anordnung von Termen zu machen. Damit kann die Menge der zulässigen Substitutionen zunächst aber nur implizit eingegrenzt werden. Es gibt aber auch Fälle, in denen es notwendig ist, bestimmte (Teil-)Substitutionen vorzuschreiben:

#### **Beispiel 4.2.5**

Sei  $E = \{f(b) \approx a, f(x) \approx c\}$ . Die Regel  $c \rightsquigarrow a$  wird genau dann in der Vervollständigung von  $E\sigma$  liegen, wenn  $\sigma$  die Variable  $x$  mit  $b$  belegt.  $\square$

Eine Möglichkeiten, eine Bedingung wie „ $\sigma$  muß eine Spezialisierung der Substitution  $\{x/b\}$  sein.“ (Beispiel 4.2.5) auszudrücken, ist, neben  $\succ$  auch  $\approx$  als Prädikaten-symbol in Ordnungsbedingungen zuzulassen und als syntaktische Gleichheit auf den

Termen zu interpretieren.<sup>2</sup> Die Bedingung kann dann durch die Formel  $x \approx b$  dargestellt werden.<sup>3</sup> Von einem theoretischen Standpunkt ist das zwar elegant, weil damit die Constraints homogene Strukturen — logische Formeln — wären. Der Nachteil ist aber, daß solche Constraints schwer zu handhaben sind. In [Peterson, 1990, Comon, 1990] wird jede mit  $\succ$  und  $\approx$  aufgebaute Formel, bevor sie in irgendeiner Form weiterverarbeitet wird, durch Umformungen auf eine Normalform gebracht, die aus zwei konjunktiv verknüpften Teilformeln besteht — eine nur mit  $\approx$ -Atomen und eine nur mit  $\succ$ -Atomen.

Das legt es nahe, diese Aufteilung schon bei der Definition der Constraints festzuschreiben. Dann kann man auf die allgemeine Form  $s \approx t$  verzichten, und sich auf Gleichungen  $x \approx t$ , deren eine Seite eine Variable ist, beschränken. Von dieser Einschränkung ausgehend ist es dann nur noch ein kleiner Schritt, ganz darauf zu verzichten, eine Substitution — im Beispiel  $\{x/b\}$  — in eine Formel wie  $x \approx b$  umzuwandeln, und statt dessen die Substitution selbst zu einem von der Ordnungsbedingung abgesetzten Teil des Constraints machen. Im Beispiel hat der aus zwei Teilen bestehende Constraint die Form  $\langle \{x/b\}, \text{true} \rangle$ . Zusammenfassend kann man als Vorteile dieser Aufteilung nennen:

- Es entspricht der Intuition, eine Beschränkung auf solche Substitutionen, die Spezialisierungen einer gegebenen Substitution  $\sigma$  sind, durch  $\sigma$  selbst auszudrücken, was auch die nachfolgenden Definitionen einfacher und verständlicher macht.
- Es macht keinen Sinn, für Constraints eine kompliziertere Sprache und Logik (mit  $\approx$ ) zu verwenden, für deren Handhabung man auch Werkzeuge zur Verfügung stellen und implementieren müßte, obwohl die einfache Sprache (ohne  $\approx$ ) genauso ausdrucksstark ist.

**Definition 4.2.6** (Constraint, Erfüllung eines Constraints)

*Ein Constraint  $c = \langle \sigma, O \rangle$  ist ein Paar bestehend aus einer idempotenten Substitution  $\sigma$  mit endlichem Definitionsbereich  $\text{Dom}(\sigma)$  und einer Ordnungsbedingung  $O$  mit der Eigenschaft  $O = O\sigma$ .*

*Eine Substitution  $\tau$  erfülle einen Constraint  $c = \langle \sigma, O \rangle$ , wenn*

- $\tau$  eine Instanz von  $\sigma$  ist, und
- $O\tau$  wahr ist.

<sup>2</sup> Dieser Weg wird in [Peterson, 1990] und [Comon, 1990] beschritten. G. Peterson beschreibt eine (unter anderem) auf Constraints basierende Methode zum Testen, ob ein gegebenes rein universelles Reduktionssystem konfluent ist. Diese Methode ist mächtiger als der Knuth-Bendix-Algorithmus; sie bietet aber nur einen Test. Darauf, wie ein nicht kanonisches Reduktionssystem unter Verwendung von Constraints zu vervollständigen wäre, wird nicht eingegangen. H. Comon beweist, daß die Erfüllbarkeit von Constraints — bzw. die äquivalenter logischer Formeln (siehe Beweis zu Satz 4.2.14) — entscheidbar ist. Auf eine konkrete Anwendung dieser Entscheidbarkeit wird nicht eingegangen.

<sup>3</sup> Man muß dazu die Definition der Ordnungsbedingungen nicht wirklich ändern, weil  $s \approx t$  durch  $\neg(s \succ t) \wedge \neg(t \succ s)$  ausgedrückt werden kann.

Eine Substitution erfülle eine Menge  $C$  von Constraints, wenn sie ein  $c \in C$  erfüllt.

Die Menge aller einen Constraint  $c$  (bzw. eine Menge  $C$  von Constraints) erfüllenden idempotenten<sup>4</sup> Substitutionen sei mit  $\text{Sat}(c)$  (bzw.  $\text{Sat}(C)$ ) bezeichnet.

Der Constraint  $\epsilon = \langle id, \text{true} \rangle$  heie der leere Constraint.

Die Forderung, da  $\text{Dom}(\sigma)$  endlich sei, ist unabdingbar.<sup>5</sup> Dagegen knnte man die Einschrnkungen, da  $\langle \sigma, O \rangle$  nur dann ein Constraint ist, wenn  $\sigma$  idempotent ist und  $O = O\sigma$ , fallenlassen, denn sie spielt fr die hinter der Definition der Constraints stehende Idee keine Rolle. Allerdings sind diese zustzlichen Eigenschaften an einigen Stellen ntzlich und bei den im folgenden tatschlich auftretenden Constraints ohnehin erfllt. Die Bedingungen in die Definition aufzunehmen, erspart es, sie an den Stellen, an denen sie bentigt werden, explizit zu nennen.

Der aus der leeren Substitution  $id$  und der wahren Ordnungsbedingung  $\text{true}$  bestehende leere Constraint wird — was direkt aus Definition 4.2.6 folgt — von allen Substitutionen erfllt.

Ein Constraint hat fr sich genommen keinen Wahrheitswert, sondern nur in Verbindung mit einer Substitution. Darum kann ein Constraint zwar nicht „wahr“ oder „falsch“ wohl aber in folgendem Sinne erfllbar sein:

**Definition 4.2.7** (Erfllbarer Constraint)

Ein Constraint  $c$  heie erfllbar oder konsistent, wenn es eine Substitution gibt, die ihn erfllt. Andernfalls heie er unerfllbar oder inkonsistent.

Eine weitere interessante Fragestellung im Zusammenhang mit Constraints ist, ob gegebene Constraints quivalent sind, oder etwa der eine allgemeiner ist als der andere und ihn darum subsumiert:

**Definition 4.2.8** (Subsumtion von Constraints, quivalente Constraints)

Ein Constraint  $c_1$  subsumiert einen Constraint  $c_2$ , in Zeichen

$$c_1 \leq c_2 \quad ,$$

wenn  $\text{Sat}(c_1) \supset \text{Sat}(c_2)$ , d. h., wenn jede Substitution, die  $c_2$  erfllt, auch  $c_1$  erfllt.

Constraints, die sich gegenseitig subsumieren, fr die also  $\text{Sat}(c_1) = \text{Sat}(c_2)$  gilt, heien quivalent. In Zeichen:  $c_1 \sim c_2$ .

Es ist auch notwendig, einen Constraint negieren oder mehrere Constraints kombinieren zu knnen:

**Definition 4.2.9** (Negation, Kombination von Constraints)

Seien  $c = \langle \sigma, O \rangle$ ,  $c_1 = \langle \sigma_1, O_1 \rangle$  und  $c_2 = \langle \sigma_2, O_2 \rangle$  Constraints.

Ist  $\sigma = \{x_1/t_1, \dots, x_n/t_n\}$ , dann heie der Constraint

$$c^{-1} = \langle id, \neg O \vee x_1 \succ t_1 \vee \dots \vee x_n \succ t_n \vee t_1 \succ x_1 \vee \dots \vee t_n \succ x_n \rangle$$

<sup>4</sup> Die Beschrnkung auf idempotente Substitutionen ist nicht unbedingt notwendig; sie schadet aber auch nicht, und erleichtert vieles.

<sup>5</sup> Sonst wre nmlich die  $\sigma$  entsprechende, mit  $\approx$  aufgebaute Formel nicht endlich — und insbesondere auch nicht deren Negation, die im folgenden noch eine Rolle spielen wird (Def 4.2.9).

die Negation von  $c_1$ .

Der Constraint

$$c_1 \sqcup c_2 = \begin{cases} \langle id, false \rangle & \text{falls } \sigma_1 \sqcup \sigma_2 = \top \\ \langle \sigma_1 \sqcup \sigma_2, (O_1 \wedge O_2)(\sigma_1 \sqcup \sigma_2) \rangle & \text{sonst} \end{cases}$$

heie die Kombination der Constraints  $c_1$  und  $c_2$ .

Der folgende Satz zeigt, da diese Definitionen der Begriffe Negation und Kombination der Intuition entspricht:

#### Satz 4.2.10

Seien  $c$ ,  $c_1$  und  $c_2$  Constraints.

1.  $\text{Sat}(c^{-1}) = \mathbf{Subst}^* \setminus \text{Sat}(c)$ , d. h., genau die idempotenten Substitutionen erfllen  $c^{-1}$ , die  $c$  nicht erfllen.
2.  $\text{Sat}(c_1 \sqcup c_2) = \text{Sat}(c_1) \cap \text{Sat}(c_2)$ , d. h., genau die idempotenten Substitutionen erfllen den Constraint  $c_1 \sqcup c_2$ , die sowohl  $c_1$  als auch  $c_2$  erfllen.

#### Beweis:

Seien  $c = \langle \sigma, O \rangle$  und  $\sigma = \{x_1/t_1, \dots, x_n/t_n\}$ , und die idempotente Substitution  $\tau$  erflle den Constraint  $c$ . Es gilt  $x_i\tau = t_i\tau$  ( $1 \leq i \leq n$ ) und  $O\tau$  ist wahr. Dann ist aber

$$(\neg O \vee x_1 \succ t_1 \vee \dots \vee x_n \succ t_n \vee t_1 \succ x_1 \vee \dots \vee t_n \succ x_n)\tau$$

falsch, und  $\tau$  erfllt  $c^{-1}$  nicht.

Erfllt umgekehrt  $\tau$  den die Negation  $c^{-1}$ , dann ist  $(\neg O)\tau$  wahr also  $O\tau$  falsch, oder aber eine der atomaren Bedingungen  $(x_i \succ t_i)\tau$  oder  $(t_i \succ x_i)\tau$  ist wahr. Dann gilt aber  $x_i\tau \neq t_i\tau$  und  $\tau$  kann keine Spezialisierung von  $\sigma$  sein.

$\tau$  erfllt genau dann die Kombination  $c_1 \sqcup c_2$  von  $c_1 = \langle \sigma_1, O_1 \rangle$  und  $c_2 = \langle \sigma_2, O_2 \rangle$ , wenn  $\tau$  eine Spezialisierung von  $\sigma_1 \sqcup \sigma_2$  ist, also sowohl von  $\sigma_1$  als auch von  $\sigma_2$ , und wenn  $(O_1 \wedge O_2)\tau$  also sowohl  $O_1\tau$  als auch  $O_2\tau$  wahr ist. Insgesamt erfllt die Substitution  $\tau$  also genau dann  $c_1 \sqcup c_2$ , wenn sie  $c_1$  und  $c_2$  erfllt. ■

Aus Satz 4.2.10 folgt auch, da die an das Supremum in einem Verband erinnernde Schreibweise  $c_1 \sqcup c_2$  fr die Kombination von Constraints gerechtfertigt ist, da nmlich gilt:

#### Korollar 4.2.11

Die Menge der Constraints modulo der quivalenzrelation  $\sim$  bilden mit der Subsumtionsrelation  $\leq$  (Def. 4.2.8) modulo  $\sim$  einen Verband.

#### Beweis:

$[c]$  bezeichne die quivalenzklasse des Constraints  $c$  bezglich  $\sim$ . Die auf der Menge der Constraints definierte Funktion  $\text{Sat}$  werde durch  $\text{Sat}([c]) = \text{Sat}(c)$  auf die Menge der quivalenzklassen fortgesetzt, sie ist damit gem der Definition von  $\sim$  wohldefiniert.

Bekanntlich bildet jede Potenzmenge, insbesondere auch die von  $\mathbf{Subst}^*$ , einen Verband mit der Inklusion  $\supset$  als Ordnung und dem Schnitt  $\cap$  als Supremums-Operator. Dann ist aber auch  $\{[c] : c \text{ ein Constraint}\}$  ein Verband, denn  $\text{Sat}$  ist, gemäß der Definition der Subsumtion  $\leq$  ein Verbands-Isomorphismus.

Zudem folgt aus Satz 4.2.10, daß  $[c_1] \sqcup [c_2]$  das Supremum von  $[c_1]$  und  $[c_2]$  im Verband der Äquivalenzklassen ist. ■

Aus Satz 4.2.10, Punkt 2, kann man ableiten, daß der Operator  $\sqcup$  — wie der Mengenoperator  $\cap$  — kommutativ, assoziativ und idempotent ist.<sup>6</sup>

### Satz 4.2.12

*Ein Constraint  $c_1$  subsumiert einen Constraint  $c_2$  genau dann, wenn  $c_1^{-1} \sqcup c_2$  unerfüllbar ist.*

*$c_1$  und  $c_2$  sind genau dann äquivalent, wenn die beiden Kombinationen  $c_1^{-1} \sqcup c_2$  und  $c_1 \sqcup c_2^{-1}$  unerfüllbar sind.*

*Ein Constraint ist unerfüllbar genau dann, wenn er von  $\langle id, false \rangle$  subsumiert wird.*

### Beweis:

$c_1$  subsumiert  $c_2$  per Definition, wenn  $\text{Sat}(c_1) \supset \text{Sat}(c_2)$ , was genau dann der Fall ist, wenn  $(\mathbf{Subst}^* \setminus \text{Sat}(c_1)) \cap \text{Sat}(c_2) = \emptyset$ . Gemäß Satz 4.2.10 ist das gleichbedeutend damit, daß  $c_1^{-1} \sqcup c_2$  unerfüllbar ist.

$c_1$  und  $c_2$  sind per Definition dann äquivalent, wenn sie sich gegenseitig subsumieren, was nach dem gerade bewiesenen genau dann der Fall ist, wenn  $c_1^{-1} \sqcup c_2$  und  $c_1 \sqcup c_2^{-1}$  unerfüllbar sind.

Es gilt  $\text{Sat}(\langle id, false \rangle) \supset \text{Sat}(c)$  genau dann, wenn  $\text{Sat}(c) = \emptyset$ , also genau dann, wenn  $c$  unerfüllbar ist. ■

Da die Möglichkeit, erkennen zu können, ob ein Constraint einen anderen subsumiert, im folgenden absolut unabdingbar ist, ist es notwendig, daß die Erfüllbarkeit von Constraints entscheidbar ist. Daneben muß es aber auch möglich sein, zu jedem Constraint eine vollständige Menge von ihm erfüllenden Substitutionen anzugeben:

### Definition 4.2.13 (Geeignete Reduktionsordnung $\succ_x$ )

*Eine Reduktionsordnung  $\succ_x$  ist als Interpretation des Symbols  $\succ$  in Ordnungsbedingungen geeignet, wenn es möglich ist, zu jedem Constraint  $c$  eine Menge  $\text{Sat}^*(c)$  von (idempotenten) Substitutionen zu berechnen (zumindest aufzuzählen), so daß es für alle  $\sigma \in \text{Sat}(c)$  ein  $\sigma' \in \text{Sat}^*(c)$  gibt mit  $\sigma' \leq \sigma$ .*

Insbesondere ist jede lexikographische Pfadordnung geeignet:

### Satz 4.2.14

*Jede lexikographische Pfadordnung  $\succ_{\text{LPO}}$  ist im Sinne von Definition 4.2.13 geeignet für die Interpretation des Prädikatsymbols  $\succ$ .*

<sup>6</sup> Das rechtfertigt es, einen Ausdruck wie  $c_1 \sqcup c_2 \sqcup c_3$  ohne Klammern zu schreiben.

**Beweis:**

In [Comon, 1990] ist bewiesen, daß die Erfüllbarkeit von Ordnungsbedingungen mit Gleichheit<sup>7</sup> entscheidbar ist, d. h. von Ordnungsbedingungen, in denen neben dem Prädikensymbol  $\succ$ , das durch eine LPO interpretiert wird, auch das Prädikensymbol  $\approx$  verwendet werden kann;  $\approx$  wird durch die Gleichheit auf Termen interpretiert.<sup>8</sup>

Damit ist auch die Erfüllbarkeit von Constraints entscheidbar, denn eine Substitution  $\tau$  erfüllt genau dann den Constraint  $\langle \sigma, O \rangle$ , wobei  $\sigma = \{x_1/t_1, \dots, x_n/t_n\}$ , wenn sie die zu ihm äquivalente Ordnungsbedingung mit Gleichheit

$$O \wedge (x \approx t_1) \wedge \dots \wedge (x \approx t_n)$$

erfüllt.

Der Beweis in [Comon, 1990] ist konstruktiv. Das verwendete Verfahren kann verwendet werden, um zu einem Constraint  $c$  eine Menge  $\text{Sat}^*(c)$  zu berechnen.

■

Das in [Comon, 1990] angegebene Verfahren ist jedoch für eine Implementierung viel zu komplex. Tatsächlich ist es — H. Comon zufolge — ein NP-hartes Problem zu entscheiden, ob ein Constraint erfüllbar ist. Auf das Problem der effizienten Implementierung wird weiter in Abschnitt 6.2 eingegangen, und beschrieben, wie es zumindest teilweise gelöst werden kann.

### 4.3 Terme und Reduktionsregeln mit Constraint

Da nun die notwendigen Hilfsmittel zur Handhabung der Constraints zur Verfügung stehen, können Reduktionsregeln und Terme mit Constraints definiert werden.

**Definition 4.3.1** (Term mit Constraint, Reduktionsregel mit Constraint, Reduktionssystem)

*Ein Term mit Constraint*

$$\mathbf{t} = (\forall x_1) \dots (\forall x_n)(t \ll c)$$

*ist ein mit einem Constraint  $c = \langle \sigma, O \rangle$  versehener Term  $t$ ,<sup>9</sup> der*

- *all-quantifiziert sein kann bezüglich einiger oder aller Variablen, die er enthält,*
- *für den  $t\sigma = t$  gilt.*

*Eine Reduktionsregel mit Constraint*

$$\mathbf{r} = (\forall x_1) \dots (\forall x_n)(s \rightsquigarrow t \ll c)$$

*ist eine mit einem Constraint  $c$  versehene Reduktionsregel, die*

<sup>7</sup> Sie werden in [Comon, 1990] als *inequational problem* bezeichnet.

<sup>8</sup>  $\approx$  wird als syntaktische Gleichheit interpretiert, und *nicht* etwa als Gleichheit bezüglich der jeweils betrachteten Menge  $E$  von Gleichungen.

<sup>9</sup> Das Symbol  $\ll$  ist als „wenn“ zu lesen.



- *universell sein kann bezüglich einiger oder aller Variablen, die sie enthält,*
- *für die  $s\sigma = s$  und  $t\sigma = t$  gilt.*

Eine Menge  $\mathcal{R}$  von Reduktionsregeln mit Constraint heie Reduktionssystem mit Constraints.

Die Mglichkeit, Terme mit einer All-Quantifizierung zu versehen, mag auf den ersten Blick seltsam erscheinen. Man mu jedoch bercksichtigen, da spter auch durch einen Term  $\mathbf{t}$  eine Aussage gemacht werden kann; seine Zugehrigkeit zu einer Menge, die die quivalenzklassen eines Termes  $\mathbf{t}'$  darstellt, sagt nmlich aus, da  $\mathbf{t}$  — und seine Instanzen — aus  $\mathbf{t}'$  ableitbar sind. Hier spielt auch die fr gemischte  $E$ -Unifikationsprobleme charakteristische Unterscheidung in solche Variablen, die zum Beweis beliebig instantiiert werden knnen, und solche, die nur einmal instantiiert werden knnen, eine Rolle. Anschaulich kann man sich einen all-quantifizierten Term als endliche Darstellung der Menge aller seiner Instanzen denken, die man (beispielsweise in einem Beweis) alle zugleich verwenden kann. Zwar stellt auch ein nicht all-quantifizierter Term jede seiner Instanzen dar, man kann aber nur jeweils eine solche Instanz tatschlich benutzen.

Zudem kann wegen der Quantifizierung der Terme mit Constraint eine Reduktionsregel mit Constraint auch als Term aufgefat werden (allerdings ber einer anderen Signatur, die  $\rightsquigarrow$  als Funktionssymbol statt als Prdikatensymbol enthlt). Das erlaubt eine einheitliche Schreibweise und Begriffsbildung.

Die Quantifizierung einer Reduktionsregel mit Constraint erstreckt sich stets auch auf den Constraint, denn ob eine Regel wie  $(\forall x)(\forall y)(x \rightsquigarrow y \ll \langle id, x \succ y \rangle)$  anwendbar ist, soll von der Instantiiierung der gebundenen Variablen  $x$  und  $y$  abhngen. Die in dem Constraint auftretenden Variablen sind aber nur dann mit diesen identisch, wenn sich die Bindung auch auf den Constraint erstreckt.

Mit Hilfe von Constraints knnen zu jeder Gleichung quivalente Reduktionsregeln angegeben werden — und zwar auch zu solchen Gleichungen die ohne Constraints nicht orientierbar sind.

### Beispiel 4.3.2

Die Gleichung  $G = (f(x) \approx g(y))$  ist ohne Constraint nicht orientierbar, schon weil sie verschieden orientierte Instanzen hat:

- Mit  $\sigma = \{x/g(a), y/a\}$  ist  $G\sigma = (f(g(a)) \approx g(a))$ , wobei  $f(g(a)) \succ g(a)$ .
- Mit  $\sigma = \{x/a, y/f(a)\}$  ist  $G\sigma = (f(a) \approx g(f(a)))$ , wobei  $g(f(a)) \succ f(a)$ .

Das ist unabhngig davon, welche Reduktionsordnung  $\succ$  man verwendet. Mit Constraints jedoch kann man die Reduktionsregeln

$$f(x) \rightsquigarrow g(y) \ll \langle id, f(x) \succ g(y) \rangle \quad \text{und} \quad g(y) \rightsquigarrow f(x) \ll \langle id, g(y) \succ f(x) \rangle$$

bilden, die die gleiche Ableitbarkeitsrelation wie die Gleichung  $G$  definieren.

Auch zu anderen typischen Beispielen fr nicht orientierbare Gleichungen knnen durch die Verwendung von Constraints quivalente Reduktionsregeln angegeben werden:

- Der Gleichung  $x \approx y$  entsprechen die Regeln

$$x \rightsquigarrow y \ll \langle id, x \succ y \rangle \quad \text{und} \quad y \rightsquigarrow x \ll \langle id, y \succ x \rangle ;$$

- der Gleichung  $(\forall x)(\forall y)(f(x, y) \approx f(y, x))$  entspricht

$$(\forall x)(\forall y)(f(x, y) \rightsquigarrow f(y, x) \ll \langle id, f(x, y) \succ f(y, x) \rangle)$$

(in diesem Fall genügt wegen ihrer Symmetrie diese eine Reduktionsregel).<sup>10</sup>

□

Daß *jede* Gleichung orientiert werden kann, rechtfertigt es, in folgender Weise jeder Gleichungsmenge ein Reduktionssystem zuzuordnen. Mit diesem System beginnt der Vervollständigungsprozeß; es wird darum als initiales System bezeichnet:

**Definition 4.3.3** (Initiales Reduktionssystem)

Sei  $E$  eine Menge von Gleichungen. Dann ist

$$\{(\forall \bar{x})(s \rightsquigarrow t \ll \langle id, s \succ t \rangle) : (\forall \bar{x})(s \approx t) \in E \text{ oder } (\forall \bar{x})(t \approx s) \in E\}$$

das zu  $E$  gehörende initiale Reduktionssystem mit Constraints.

Da das Ziel ist, ein Reduktionssystem  $\mathcal{R}$  mit Constraints anzugeben, das eine einheitliche Darstellung ist von Reduktionssystemen  $\mathcal{R}[\sigma]$ , die äquivalent sind zu den verschiedenen Instanzen  $E\sigma$  einer Gleichungsmenge  $E$ , und weil — wie Beispiel 3.6.5 zeigt — im allgemeinen  $\mathcal{R}[\sigma] \neq \mathcal{R}\sigma$  gilt, muß zunächst festgelegt werden, wie das zu einer vorgegebenen Instanz  $E\sigma$  äquivalente System  $\mathcal{R}[\sigma]$  aus  $\mathcal{R}$  zu bestimmen ist. In der folgenden Definition werden Reduktionssysteme mit Constraints als Spezialfälle von Mengen von Termen mit Constraint aufgefaßt, und die Definition der Instanz zunächst für den allgemeinen Fall gegeben:

**Definition 4.3.4** (Instanz einer Menge von Termen mit Constraint, eines Reduktionssystems)

Sei  $\mathcal{T}$  eine Menge von Termen mit Constraint und  $\sigma$  eine Substitution. Dann heiße

$$\mathcal{T}[\sigma] = \{(\forall \bar{x})(t\sigma' \ll c \sqcup \langle \sigma, \text{true} \rangle) : (\forall \bar{x})(t \ll c) \in \mathcal{T}, \sigma \text{ erfüllt } c, \\ \sigma' = \sigma|_{\text{Free}((\forall \bar{x})(t \ll c))}\}$$

eine Instanz von  $\mathcal{T}$ .

Ist  $\sigma$  eine Grundsubstitution, dann heiße die Menge

$$\mathcal{T}(\sigma) = \{(\forall x)t : (\forall x)(t \ll c) \in \mathcal{T}[\sigma]\}$$

eine Grundinstanz von  $\mathcal{T}$ .

Die in obiger Weise definierten Instanzen eines als Menge von Termen mit Constraint aufgefaßten Reduktionssystems  $\mathcal{R}$  heißen Instanzen von  $\mathcal{R}$ .

<sup>10</sup> Verwendet man eine lexikographische Pfadordnung, kann die einfachere Form

$$(\forall x)(\forall y)(f(x, y) \rightsquigarrow f(y, x) \ll \langle id, x \succ y \rangle)$$

verwendet werden, denn dann gilt für jede Substitution  $\sigma$ :  $f(x, y)\sigma \succ f(y, x)\sigma$  genau dann, wenn  $x\sigma \succ y\sigma$ .

**Beispiel 4.3.5**

Sei  $\mathcal{R}_1 = \{f(x) \rightsquigarrow y \ll \langle id, f(x) \succ y \rangle\}$ . Dann ist

$$\begin{aligned}\mathcal{R}_1[\{x/y\}] &= \{f(y) \rightsquigarrow y \ll \langle id, f(y) \succ y \rangle\}^{11} \\ \mathcal{R}_1\langle\{x/a, y/b\}\rangle &= \{f(a) \rightsquigarrow b\}\end{aligned}$$

Sei  $\mathcal{R}_2 = \{(\forall x)(f(x) \approx x \ll \epsilon)\}$ . Entsprechend Definition 2.3.4 werden gebundene Variablen nicht belegt. Darum gilt für alle Substitutionen  $\sigma$

$$\mathcal{R}_2\langle\sigma\rangle = \{(\forall x)(f(x) \approx x)\}$$

und  $\mathcal{R}_2[\sigma] = \mathcal{R}_2$ . □

In Grundinstanzen die Constraints wegzulassen, ist dadurch gerechtfertigt, daß in ihnen alle Terme mit einem Constraint versehen sind, der zum leeren Constraint  $\epsilon = \langle id, true \rangle$  äquivalent ist. Die Grundinstanzen  $\mathcal{R}\langle\sigma\rangle$  eines Reduktionssystems  $\mathcal{R}$  sind herkömmliche Reduktionssysteme ohne Constraints, und definieren daher eine Ableitbarkeitsrelation  $\rightarrow_{\mathcal{R}\langle\sigma\rangle}$  auf den Termen ohne Constraint.

Im allgemeinen jedoch definiert ein Reduktionssystem  $\mathcal{R}$  mit Constraints Ableitbarkeitsrelationen  $\Rightarrow_{\mathcal{R}}$  und  $\ni_{\mathcal{R}}$  nur auf den Termen mit Constraint:

**Definition 4.3.6** (Die Ableitbarkeitsrelationen  $\Rightarrow_{\mathcal{R}}$  und  $\ni_{\mathcal{R}}$ )

Sei  $\mathcal{R}$  ein Reduktionssystem mit Constraints. Dann sind die Relationen  $\Rightarrow_{\mathcal{R}}$  und  $\ni_{\mathcal{R}}$  auf der Menge der Terme mit Constraint folgendermaßen definiert:

Sei

$$\mathbf{t} = (\forall \bar{x})(t \ll c_t)$$

ein Term mit Constraint. Genau dann, wenn es eine Reduktionsregel

$$\mathbf{r} = (\forall \bar{y})(l \rightsquigarrow r \ll c_r)$$

in  $\mathcal{R}$  gibt, so daß

1.  $\{x_1, \dots, x_n\} \cap \text{Var}(r) = \emptyset$  und  $\{y_1, \dots, y_m\} \cap \text{Var}(t) = \emptyset$ ,<sup>12</sup>
2. es eine Position  $p$  in  $t$  gibt, so daß

$$t|_p \notin \mathcal{V} \quad \text{oder} \quad t|_p = l ;^{13}$$

3.  $t|_p$  und  $l$  unifizierbar sind mit einem idempotenten<sup>14</sup> MGU  $\nu$ ,

<sup>11</sup> Da der Constraint der Regel wahr ist, kann sie zu  $f(y) \rightsquigarrow y \ll \epsilon$  vereinfacht werden.

<sup>12</sup> Diese Bedingung ist keine Beschränkung der Allgemeinheit; nötigenfalls läßt sie sich durch gebundene Variablenumbenennung einhalten.

<sup>13</sup> Eine Gleichung kann also nur dann auf eine Variable angewendet werden, wenn ihre linke Seite mit der Variablen identisch ist (was bedeutet, das es sich um eine freie Variable handelt).

<sup>14</sup> Auch das ist keine Beschränkung der Allgemeinheit, denn zwei unifizierbare Terme haben stets auch einen idempotenten allgemeinsten Unifikator.

## 4. die Kombination

$$c_{neu} = \langle \mu, O_{neu} \rangle = c_t \sqcup c_r \sqcup \langle \nu, \text{true} \rangle$$

konsistent ist,

gelte

$$\mathbf{t} \Rightarrow_{\mathcal{R}} \mathbf{t}' ,$$

wobei

$$\mathbf{t}' = (\forall \bar{x})(\forall \bar{y})((t[p/r])\mu \ll c_{neu}) .$$

In diesem Falle heie  $\langle \mathbf{r}, p, \mu \rangle$  eine Rechtfertigung fur  $\mathbf{t} \Rightarrow_{\mathcal{R}} \mathbf{t}'$ .

Seien  $\mathbf{t}$  und  $\mathbf{t}'$  Terme mit Constraint. Genau dann, wenn entsprechend obiger Definition  $\mathbf{t} \Rightarrow_{\mathcal{R}} \mathbf{t}'$  gilt mit einer Rechtfertigung  $\langle \mathbf{r}, p, \mu \rangle$ , so da

1.  $t|_p = l\mu$ ,
2.  $c_t$  von  $c_{neu}$  subsumiert wird,

gelte

$$\mathbf{t} \ni_{\mathcal{R}} \mathbf{t}' .$$

Wie man leicht nachvollzieht, sind die Ableitbarkeitsrelationen  $\Rightarrow_{\mathcal{R}}$  und  $\ni_{\mathcal{R}}$  wohldefiniert, d. h., ist  $\mathcal{R}$  ein Reduktionssystem mit Constraints und  $\mathbf{t}$  ein Term mit Constraint, und gilt

$$\mathbf{t} \Rightarrow_{\mathcal{R}} \mathbf{t}' \quad \text{oder} \quad \mathbf{t} \ni_{\mathcal{R}} \mathbf{t}' ,$$

dann ist auch  $\mathbf{t}'$  ein Term mit Constraint: Der Constraint in  $\mathbf{t}'$  ist als Kombination (Def. 4.2.9) von wohlgeformten Constraints wieder wohlgeformt, und die idempotente Substitution  $\mu$  wird auf den neuen Term  $t_{neu}$  angewendet, so da auch die Bedingung  $t_{neu}\mu = t_{neu}$  erfullt ist.

Aus Definition 4.3.6 folgt sofort, da  $\ni_{\mathcal{R}}$  eine Teilrelation von  $\Rightarrow_{\mathcal{R}}$  ist.

Die Idee bei der Definition der Ableitbarkeitsrelation  $\ni_{\mathcal{R}}$  ist, da sich alle durch die Grundinstanzen  $\mathcal{R}\langle \sigma \rangle$  von  $\mathcal{R}$  definierten Ableitbarkeitsrelationen  $\rightarrow_{\mathcal{R}}$  in ihr widerspiegeln. Wann immer aus einer Instanz  $\mathbf{s}\langle \sigma \rangle$  eines Terms mit Constraint  $\mathbf{s}$  durch Anwendung einer Regel im Reduktionssystem  $\mathcal{R}\langle \sigma \rangle$  ein Term  $t$  ableitbar ist, soll  $\mathbf{s} \ni_{\mathcal{R}} \mathbf{t}$  gelten, so da  $t = \mathbf{t}\langle \sigma \rangle$  eine Instanz von  $\mathbf{t}$  ist.

Der wesentliche Unterschied zwischen den beiden Ableitbarkeitsrelationen ist, da die in der Definition von  $\ni_{\mathcal{R}}$  zustzlich aufgestellten Bedingungen sicher stellen, da die Ableitung  $\mathbf{t} \ni_{\mathcal{R}} \mathbf{t}'$  — sieht man von der Ordnung auf den Termen ab — ein umkehrbarer Proze ist, der Term mit Constraint  $\mathbf{t}$  also durch den Term mit Constraint  $\mathbf{t}'$  ersetzt werden kann. Was genau „kann ersetzt werden“ bedeutet, wird erst im folgenden deutlich werden. Gemeint ist nicht „wird subsumiert“ — der Begriff Subsumtion ist fur Terme mit Constraint in etwas anderer Weise definiert (Def. 4.3.8) —, sondern da die jeweilige Menge, in der  $\mathbf{t}$  durch  $\mathbf{t}'$  ersetzt wird, keine ihrer fur Korrektheit und Vollstandigkeit notwendigen Eigenschaften verliert. Dabei

übernimmt der Term  $\mathbf{t}'$  die Funktionen von  $\mathbf{t}$  nur im Zusammenspiel mit den Regeln im Reduktionssystem  $\mathcal{R}$ ; während ein  $\mathbf{t}$  subsumierender Term diesen stets ersetzen kann — unabhängig vom jeweiligen System  $\mathcal{R}$ .

Eine Ableitung  $\mathbf{t} \Rightarrow_{\mathcal{R}} \mathbf{t}'$  stellt, falls  $\mathbf{t} \not\equiv_{\mathcal{R}} \mathbf{t}'$ , einen nicht umkehrbaren Prozeß dar. Der abgeleitete Term  $\mathbf{t}'$  kann nicht alle Funktionen von  $\mathbf{t}$  übernehmen, sondern nur die einiger seiner Instanzen.

### Beispiel 4.3.7

Hier einige Beispiele für Ableitungsschritte und ihre Rechtfertigungen:

$$\begin{aligned} (g(a, c) \ll \epsilon) &\Rightarrow (g(a, b) \ll \epsilon) && \text{mit } \langle (c \rightsquigarrow b \ll \epsilon), \langle 2 \rangle, id \rangle \\ (f(c) \ll \epsilon) &\Rightarrow (c \ll \epsilon) && \text{mit } \langle ((\forall x)(f(x) \rightsquigarrow x \ll \epsilon)), \langle \rangle, id \rangle \\ (a \ll \epsilon) &\Rightarrow (y \ll \langle \{x/a\}, a \succ y \rangle) && \text{mit } \langle (x \rightsquigarrow y \ll \langle id, x \succ y \rangle), \langle \rangle, \{x/a\} \rangle \\ (f(c) \ll \epsilon) &\Rightarrow (c \ll \langle \{x/c\}, \text{true} \rangle) && \text{mit } \langle (f(x) \rightsquigarrow x \ll \epsilon), \langle \rangle, \{x/c\} \rangle \end{aligned}$$

□

Gilt  $\mathbf{t} \Rightarrow_{\mathcal{R}} \mathbf{t}'$ , dann wird der Constraint  $c'$  des Terms  $\mathbf{t}'$  vom Constraint  $c$  des Terms  $\mathbf{t}$  subsumiert, d. h. die Constraints werden bei der Ableitung stets spezieller.

Es ist sinnvoll, eine Subsumtionsrelation auf den Termen mit Constraint zu definieren. Diese entspricht in etwa der Beziehung, die zwischen einem Term (ohne Constraint) und einer seiner Instanzen besteht. Im folgenden wird ein Term mit Constraint  $\mathbf{t}$ , der einen Term mit Constraint  $\mathbf{t}'$  subsumiert, sämtliche Funktionen von  $\mathbf{t}'$  übernehmen können.

### Definition 4.3.8 (Subsumtion von Termen mit Constraint)

Ein Term mit Constraint  $\mathbf{t}_1 = (\forall \bar{x})(t_1 \ll c_1)$  subsumiert einen Term mit Constraint  $\mathbf{t}_2 = (\forall \bar{y})(t_2 \ll c_2)$  genau dann, wenn — gegebenenfalls nach einer gebundenen Umbenennung der Variablen in  $\mathbf{t}_1$  —

1. der Term  $t_2$  eine Instanz ist von  $t_1$ , d. h. eine idempotente Substitution  $\mu$  existiert, so daß  $t_2 = t_1\mu$ ,
2. der Constraint  $c_1 \sqcup \langle \mu, \text{true} \rangle$  den Constraint  $c_2$  subsumiert.

Wie man leicht nachvollzieht, gilt, falls der Term mit Constraint  $\mathbf{t}_1$  den Term mit Constraint  $\mathbf{t}_2$  subsumiert, und  $\sigma$  eine Grundsubstitution ist, die den Constraint von  $\mathbf{t}_1$  erfüllt:

$$\mathbf{t}_1\langle \sigma \rangle = \mathbf{t}_2\langle \sigma \rangle .$$

### Beispiel 4.3.9

Beispielsweise subsumiert der Term  $a \ll \epsilon$  den Term  $a \ll \langle \{x/a\}, \text{true} \rangle$ .

Falls  $g(b) \succ f(a)$  wahr ist, subsumiert die Reduktionsregel

$$(\forall x)(\forall y)(g(x) \rightsquigarrow f(y) \ll \epsilon)$$

die Regel

$$g(b) \rightsquigarrow f(a) \ll \epsilon .$$

□

## 4.4 Transformationsregeln und Vervollständigung

### 4.4.1 Ziel der Vervollständigung

In diesem Abschnitt nun werden Transformationsregeln angegeben, die ein Verfahren für die Vervollständigung von Reduktionssystemen mit Constraints definieren. Durch wiederholte (faire) Anwendung der Transformationsregeln wird ausgehend von einem initialen System  $\mathcal{R} = \mathcal{R}^0$  ein System  $\mathcal{R}^\infty$  angenähert, daß eine einheitliche Darstellung der (klassischen) Vervollständigungen aller Grundinstanzen von  $\mathcal{R}$  ist.

Die Grundinstanzen  $\mathcal{R}^\infty\langle\sigma\rangle$  von  $\mathcal{R}^\infty$  sind jedoch im allgemeinen nicht irreduzibel und also nicht kanonisch. Das zu verlangen würde, wie das folgende Beispiel zeigt, auch keinen Sinn machen.

#### Beispiel 4.4.1

Es sei das Reduktionssystem

$$\mathcal{R}^\infty = \{f(x) \rightsquigarrow c \ll \epsilon, a \rightsquigarrow b \ll \epsilon\}$$

vorgelegt. Auf dieses System ist keine der im nächsten Abschnitt definierten Transformationsregeln anwendbar.

Dennoch ist mit  $\sigma = \{x/a\}$  die Grundinstanz

$$\mathcal{R}^\infty\langle\sigma\rangle = \{f(a) \rightsquigarrow c, a \rightsquigarrow b\}$$

keineswegs kanonisch. □

Die Eigenschaften der „Vervollständigung“  $\mathcal{R}^\infty$  selbst sind jedoch wichtiger als die ihrer Grundinstanzen. Die durch  $\mathcal{R}^\infty$  auf den Termen mit Constraint definierte Relation  $\Rightarrow_{\mathcal{R}^\infty}$  ist nämlich in gewissem (in Abschnitt 4.9 näher präzisierendem) Sinne konfluent. Das erlaubt es, deterministisch Lösungen  $\sigma$  eines gegebenen gemischten  $E$ -Unifikationsproblems  $\langle E, s, t \rangle$  zu berechnen (und nicht nur zu überprüfen, ob ein gegebenes  $\sigma$  Unifikator ist). Tatsächlich wird es möglich sein, eine Menge  $\mathcal{C}(\langle E, s, t \rangle)$  von Constraints zu bestimmen, so daß die Menge  $\text{Sat}(\mathcal{C}(\langle E, s, t \rangle))$  der  $\mathcal{C}$  erfüllenden Substitutionen eine bezüglich der Subsumtionsrelation  $\leq_E$  grund-vollständige Menge von Unifikatoren ist.

### 4.4.2 Löschung

Es ist sicherlich sinnvoll, solche Regeln zu löschen, deren Constraint unerfüllbar ist, und die ohnehin niemals angewendet werden können:

#### Definition 4.4.2 (Löschungsregel)

Sei  $\mathcal{R}$  ein Reduktionssystem mit Constraints. Die Löschungsregel ist definiert durch:

$$(L) \quad \frac{\mathcal{R} \cup \{(\forall \bar{x})(s \rightsquigarrow t \ll c)\}}{\mathcal{R}} \quad c \text{ inkonsistent}$$

#### Beispiel 4.4.3

Die Regel  $x \rightsquigarrow f(x) \ll \langle id, x \succ f(x) \rangle$  kann gelöscht werden, da  $\langle id, x \succ f(x) \rangle$  unerfüllbar ist. □

### 4.4.3 Äquivalenzumformung

Die Umformungsregel ist — wie die Lösungsregel — zwar sinnvoll, für die Vollständigkeit des Verfahrens jedoch keineswegs notwendig. Sie erlaubt es, den Constraint  $c$  einer Reduktionsregel durch eine Menge  $\{c_1, \dots, c_n\}$  von Constraints zu ersetzen, die — disjunktiv verknüpft — zu  $c$  äquivalent sind, d. h.

$$\text{Sat}(c) = \bigcup_{1 \leq i \leq n} \text{Sat}(c_i) .$$

Da jede Regel nur *einen* Constraint haben kann, werden  $n$  Kopien der ursprünglichen Reduktionsregel erzeugt und mit den Constraints  $c_1, \dots, c_n$  versehen.

Die Umformungsregel erlaubt es, beliebige Verfahren zu verwenden um Constraints auf eine einfache Normalform zu bringen. Das kann die Handhabung der Constraints entscheidend erleichtern (siehe Abschnitt 6.2). Bei der Umformung kann jedes Wissen verwendet werden, das man über die spezielle Reduktionsordnung  $\succ_x$  hat, durch die  $\succ$  interpretiert wird.

#### Definition 4.4.4 (Äquivalenzumformungsregel)

Sei  $\mathcal{R}$  ein Reduktionssystem mit Constraints. Die Regel zur Äquivalenzumformung ist definiert durch:

$$(\ddot{A}) \quad \frac{\mathcal{R} \cup \{(\forall \bar{x})(l \rightsquigarrow r \ll c)\}}{\mathcal{R} \cup \{(\forall \bar{x})(l\sigma \rightsquigarrow r\sigma \ll \langle \sigma, O \rangle) : \langle \sigma, O \rangle \in C\}} \quad \begin{array}{l} \text{Sat}(c) = \text{Sat}(C), \\ C \text{ endlich} \end{array}$$

#### Beispiel 4.4.5

Wird  $\succ$  durch eine LPO  $\succ_{\text{LPO}}$  (Def. 3.6.7) interpretiert, dann kann die Regel

$$f(x, y) \rightsquigarrow f(a, b) \ll \langle id, f(x, y) \succ f(a, b) \rangle$$

durch die beiden neuen Regeln

$$\begin{array}{l} f(x, y) \rightsquigarrow f(a, b) \ll \langle id, x \succ a \rangle \\ f(a, y) \rightsquigarrow f(a, b) \ll \langle \{x/a\}, y \succ b \rangle \end{array}$$

ersetzt werden. □

Für  $n = 0$  entspricht die Umformungsregel der Lösungsregel (Def. 4.4.2). Diese kann darum als ein Spezialfall der Äquivalenzumformung aufgefaßt werden.

### 4.4.4 Subsumtion

Wird eine in einem Reduktionssystem  $\mathcal{R}$  enthaltene Regel von einer anderen Regel in  $\mathcal{R}$  subsumiert (Def. 4.3.8), so kann sie gelöscht werden:

#### Definition 4.4.6 (Subsumtionsregel)

Sei  $\mathcal{R}$  ein Reduktionssystem mit Constraints. Die Subsumtionsregel ist definiert durch:

$$(S) \quad \frac{\mathcal{R} \cup \{\mathbf{r}, \mathbf{r}'\}}{\mathcal{R} \cup \{\mathbf{r}\}} \quad \mathbf{r} \text{ subsumiert } \mathbf{r}'$$

Man darf eine Regel nicht alleine schon deshalb löschen, weil sie sich selbst subsumiert; damit in Einklang steht, daß die Anwendung der Subsumptionsregel nichts bewirkt, wenn  $\mathbf{r} = \mathbf{r}'$  (ihre Anwendung ist in diesem Falle allerdings sinnlos).

#### 4.4.5 Kritisches Paar, Kombination, Vereinfachung

Durch die bisher beschriebenen Transformationen werden Reduktionsregeln gelöscht bzw. durch äquivalente Regeln ersetzt, ohne daß dabei die Ableitbarkeitsrelation  $\Rightarrow_{\mathcal{R}}$  eine Rolle spielte. Zur Vervollständigung eines Reduktionssystems  $\mathcal{R}$  muß aber  $\Rightarrow_{\mathcal{R}}$  berücksichtigt werden, indem eine Regel  $\mathbf{r}_2 \in \mathcal{R}$  auf eine Regel  $\mathbf{r}_1 \in \mathcal{R}$  angewendet werden kann.

Seien also

$$\mathbf{r}_1 = (\forall \bar{x})(s \rightsquigarrow t \ll c_1) \quad \text{und} \quad \mathbf{r}_2 = (\forall \bar{y})(l \rightsquigarrow r \ll c_2) ,$$

und die Regel  $\mathbf{r}_2$  sei auf  $\mathbf{r}_1$  anwendbar, d. h. es gelte  $\mathbf{r}_1 \Rightarrow \mathbf{r}'_1$  mit einer Rechtfertigung  $\langle \mathbf{r}_2, p, \mu \rangle$ . Die neue Regel

$$\mathbf{r}'_1 = (\forall \bar{x})(\forall \bar{y})(s_{neu} \rightsquigarrow t_{neu} \ll c_{neu})$$

kann man nun allerdings noch nicht ohne weiteres verwenden. Zum einen können ihre verschiedenen Instanzen unterschiedlich orientiert sein, weswegen die beiden symmetrischen Versionen

$$\begin{aligned} \mathbf{r}_{neu1} &= (\forall \bar{x})(\forall \bar{y})(s_{neu} \rightsquigarrow t_{neu} \ll c_{neu} \sqcup \langle id, s_{neu} \succ t_{neu} \rangle) \\ \mathbf{r}_{neu2} &= (\forall \bar{x})(\forall \bar{y})(t_{neu} \rightsquigarrow s_{neu} \ll c_{neu} \sqcup \langle id, t_{neu} \succ s_{neu} \rangle) , \end{aligned}$$

der Regel betrachtet werden müssen. Zum anderen hängt die Form der Transformationsregel davon ab,

- ob neben  $\mathbf{r}_1 \Rightarrow \mathbf{r}'_1$  auch  $\mathbf{r}_1 \ni \mathbf{r}'_1$  gilt (mit der gleichen Rechtfertigung).
- auf welche Seite von  $\mathbf{r}_1$  die Regel  $\mathbf{r}_2$  angewendet wurde, d. h., ob  $p$  eine Position in  $s$  oder in  $t$  ist.

Falls  $\mathbf{r}_1 \ni \mathbf{r}'_1$  gilt, können  $\mathbf{r}_{neu1}$  und  $\mathbf{r}_{neu2}$  die Funktion von  $\mathbf{r}_1$  im Zusammenspiel vollständig übernehmen, erlauben also die gleichen Ableitungen wie  $\mathbf{r}_1$ . Wurde dabei  $\mathbf{r}_2$  auf die rechte Seite von  $\mathbf{r}_1$  angewendet, kann man außerdem schließen, daß  $\mathbf{r}_{neu2}$  inkonsistent ist. In diesem Fall heißt die Transformation eine „Vereinfachung“, denn  $\mathbf{r}_1$  kann durch die einzelne neue Regel  $\mathbf{r}_{neu1}$  ersetzt werden. Wurde  $\mathbf{r}_2$  auf die rechte Seite von  $\mathbf{r}_1$  angewendet, kann  $\mathbf{r}_{neu2}$  konsistent sein und also nicht weggelassen werden; eine solche Transformation heißt „Komposition“.

Gilt  $\mathbf{r}_1 \ni \mathbf{r}'_1$  *nicht*,<sup>15</sup> ersetzen die neuen Regeln die alte Regel  $\mathbf{r}_1$  nicht. Eine solche Transformation führt daher zu zusätzlichen Regeln, und wird nur ausgeführt, wenn es zur Vervollständigung nötig ist — nämlich dann, wenn  $\mathbf{r}_2$  auf die linke Seite von  $\mathbf{r}_1$  angewendet wurde, und daher  $\mathbf{r}_1$  und  $\mathbf{r}_2$  ein kritisches Paar bilden. In diesem Fall werden  $\mathbf{r}_{neu1}$  und  $\mathbf{r}_{neu2}$  als zusätzliche Regeln in das Reduktionssystem aufgenommen.

<sup>15</sup> Das heißt: nicht mit der gleichen Rechtfertigung wie  $\mathbf{r}_1 \Rightarrow \mathbf{r}'_1$ ; ob  $\mathbf{r}_1 \ni \mathbf{r}'_1$  mit einer anderen Rechtfertigung gilt (insbesondere bei einer Anwendung an einer anderen Position  $p'$ ) ist unerheblich.



**Definition 4.4.7** (Vereinfachungs-, Kompositions-, Kritisches-Paar-Regel)  
 Sei  $\mathcal{R}$  ein Reduktionssystem mit Constraints, und für Regeln  $\mathbf{r}_1, \mathbf{r}_2 \in \mathcal{R}$  der Form

$$\mathbf{r}_1 = (\forall \bar{x})(s \rightsquigarrow t \ll c_1) \quad \text{und} \quad \mathbf{r}_2 = (\forall \bar{y})(l \rightsquigarrow r \ll c_2)$$

gelte  $\mathbf{r}_1 \Rightarrow \mathbf{r}'_1$  mit einer Rechtfertigung  $\langle \mathbf{r}_2, p, \mu \rangle$ . Hat  $\mathbf{r}'_1$  die Form

$$\mathbf{r}'_1 = (\forall \bar{x})(\forall \bar{y})(s_{neu} \rightsquigarrow t_{neu} \ll c_{neu}) ,$$

dann seien die neuen Regeln  $\mathbf{r}_{neu1}$  und  $\mathbf{r}_{neu2}$  durch

$$\begin{aligned} \mathbf{r}_{neu1} &= (\forall \bar{x})(\forall \bar{y})(s_{neu} \rightsquigarrow t_{neu} \ll c_{neu} \sqcup \langle id, s_{neu} \succ t_{neu} \rangle) \\ \mathbf{r}_{neu2} &= (\forall \bar{x})(\forall \bar{y})(t_{neu} \rightsquigarrow s_{neu} \ll c_{neu} \sqcup \langle id, t_{neu} \succ s_{neu} \rangle) \end{aligned}$$

definiert und die Vereinfachungsregel (V), die Kompositionsregel (Ko) und die Kritische-Paar-Regel (KP) durch:

$$\begin{aligned} \text{(V)} \quad & \frac{\mathcal{R}}{(\mathcal{R} \setminus \{\mathbf{r}_1\}) \cup \{\mathbf{r}_{neu1}\}} \quad p \text{ in } t \\ \text{(Ko)} \quad & \frac{\mathcal{R}}{(\mathcal{R} \setminus \{\mathbf{r}_1\}) \cup \{\mathbf{r}_{neu1}, \mathbf{r}_{neu2}\}} \quad p \text{ in } s \\ \text{(KP)} \quad & \frac{\mathcal{R}}{\mathcal{R} \cup \{\mathbf{r}_{neu1}, \mathbf{r}_{neu2}\}} \quad p \text{ in } s, \text{ nicht } \mathbf{r}_1 \Rightarrow_{\mathcal{R}} \mathbf{r}'_1 \end{aligned}$$

Der wesentliche Unterschied zu der Kritische-Paar-Regel bei dem Verfahren von Gallier et al. ist, daß der Unifikator  $\mu$  nur lokal auf die neue Regel angewendet wird, und nicht etwa auf ganz  $\mathcal{R}$ .

### Beispiel 4.4.8

Aus dem Reduktionssystem

$$\mathcal{R} = \{x \rightsquigarrow y \ll \langle id, x \succ y \rangle, \\ x \rightsquigarrow z \ll \langle id, x \succ z \rangle, \\ y \rightsquigarrow u \ll \langle id, y \succ u \rangle\}$$

sind in jeweils einem Schritt die folgenden Systeme  $\mathcal{R}'$ ,  $\mathcal{R}''$  und  $\mathcal{R}'''$  ableitbar, nämlich mit Hilfe der Vereinfachungsregel das System

$$\mathcal{R}' = \{x \rightsquigarrow u \ll \langle id, (x \succ y) \wedge (y \succ u) \wedge (x \succ u) \rangle, \\ x \rightsquigarrow z \ll \langle id, x \succ z \rangle, \\ y \rightsquigarrow u \ll \langle id, y \succ u \rangle\} ,$$

mit Hilfe der Kompositionsregel das System

$$\mathcal{R}'' = \{z \rightsquigarrow y \ll \langle id, (x \succ y) \wedge (x \succ z) \wedge (z \succ y) \rangle, \\ y \rightsquigarrow z \ll \langle id, (x \succ y) \wedge (x \succ z) \wedge (y \succ z) \rangle, \\ x \rightsquigarrow z \ll \langle id, x \succ z \rangle, \\ y \rightsquigarrow u \ll \langle id, y \succ u \rangle\}$$

und mit Hilfe der Kritischen-Paar-Regel das System

$$\begin{aligned} \mathcal{R}''' = \{ & x \rightsquigarrow y \ll \langle id, x \succ y \rangle, \\ & x \rightsquigarrow z \ll \langle id, x \succ z \rangle, \\ & y \rightsquigarrow u \ll \langle id, y \succ u \rangle, \\ & z \rightsquigarrow u \ll \langle \{x/y\}, (y \succ z) \wedge (y \succ u) \wedge (z \succ u) \rangle, \\ & u \rightsquigarrow z \ll \langle \{x/y\}, (y \succ z) \wedge (y \succ u) \wedge (u \succ z) \rangle \} \end{aligned}$$

□

Die Kompositionsregel (Ko) könnte weggelassen werden, wenn man die Kritische-Paar-Regel (KP) verallgemeinerte und nur mit der Bedingung „ $p$  eine Position in  $s$ “ versähe. Eine Anwendung der Regel (Ko) würde dann jeweils durch eine Anwendung dieser allgemeineren Regel (KP) und eine anschließende Anwendung der Subsumptionsregel (S) ersetzt. Die Zusammenfassung von (KP) und (Ko) würde jedoch die Effizienz des Verfahrens beeinträchtigen.

#### 4.4.6 Faire Vervollständigungsverfahren

Im allgemeinen kann die Vervollständigung eines Reduktionssystems unendlich viele Anwendungen der Transformationsregeln erfordern (ob es eine einfache Charakterisierung der Fälle gibt, in denen sie endlich ist, ist eine offene Frage). Auch wenn eine Folge von Transformationen nicht terminiert, wird aber eine Vervollständigung  $\mathcal{R}^\infty$  approximiert. Sie enthält die *persistenten* Reduktionsregeln, die in allen bis auf endlich viele der entstehenden Reduktionssysteme enthalten sind.

**Definition 4.4.9** (Relation  $\vdash$ , Transformationsprozedur, Vervollständigung)  
Sei  $\mathcal{R}$  ein Reduktionssystem.

$$\mathcal{R} \vdash \mathcal{R}'$$

bedeute, daß  $\mathcal{R}'$  durch Anwendung einer der Transformationsregeln (Def. 4.4.2, 4.4.4, 4.4.6 und 4.4.7) aus  $\mathcal{R}$  entsteht.

Eine Transformationsprozedur legt fest, in welcher Reihenfolge die Transformationsregeln ausgehend von einem Reduktionssystem  $\mathcal{R} = \mathcal{R}^0$  anzuwenden sind, so daß eine Vervollständigungsfolge

$$\mathcal{R}^0 \vdash \mathcal{R}^1 \vdash \mathcal{R}^2 \vdash \dots$$

entsteht.

Das dadurch eindeutig bestimmte Reduktionssystem

$$\mathcal{R}^\infty = \begin{cases} \mathcal{R}^m & \text{falls die Vervollständigungsfolge von der Länge } m \text{ ist} \\ \bigcup_{k \geq 0} \bigcap_{m \geq k} \mathcal{R}^m & \text{falls die Vervollständigungsfolge unendlich ist} \end{cases}$$

heiße die Vervollständigung von  $\mathcal{R}$ .

Ist  $\mathcal{R}$  das initiale System zu einer Menge  $E$  von Gleichungen, dann heiße  $\mathcal{R}^\infty$  auch eine Vervollständigung von  $E$ .

Die Vervollständigung eines Reduktionssystems wird die gewünschten Eigenschaften nur haben, wenn auch bei einer unendlichen Vervollständigungsfolge sichergestellt ist, daß jede mögliche Transformation schließlich auch ausgeführt wird — es sei denn die betroffenen Regeln gehören zwar zu einem Element  $\mathcal{R}^i$  der Vervollständigungsfolge, nicht aber zu  $\mathcal{R}^\infty$ , werden wieder gelöscht.

**Definition 4.4.10** (Fairneß einer Transformation)

Eine Transformationsprozedur heie *fair*, wenn

1. die Umformungsregel ( $\ddot{A}$ ) nicht auf Regeln angewendet wird, die selbst durch eine  $\ddot{A}$ quivalenzumformung entstanden sind;
2. fr jede Grundsubstitution  $\sigma$  die Menge der  $\sigma$ -persistenten Regeln

$$(\mathcal{R}\langle\sigma\rangle)^\infty = \begin{cases} \mathcal{R}^m\langle\sigma\rangle & \text{falls die Vervollstndigungsfolge endlich} \\ & \text{und von der Lnge } m \text{ ist} \\ \bigcup_{k \geq 0} \bigcap_{m \geq k} \mathcal{R}^m\langle\sigma\rangle & \text{falls sie unendlich ist} \end{cases}$$

Teilmenge von  $\mathcal{R}^\infty\langle\sigma\rangle$  ist.

3. es fr jedes kritische Paar  $\mathbf{r}_1, \mathbf{r}_2 \in \mathcal{R}^i$  ( $i \geq 0$ ), das fr eine Grundsubstitution  $\sigma$   $\sigma$ -persistent ist,<sup>16</sup> einen Index  $j$  gibt, so da das Reduktionssystem  $\mathcal{R}^{j+1}$  in der Vervollstndigungsfolge durch Anwendung der Kritisches-Paar-Regel auf die entsprechenden Regeln  $\mathbf{r}'_1, \mathbf{r}'_2 \in \mathcal{R}^j$  (deren Instanzen mit denen von  $\mathbf{r}_1$  und  $\mathbf{r}_2$  bereinstimmen) aus  $\mathcal{R}^j$  entstanden ist;

Vervollstndigungen und Vervollstndigungsfolgen heien *fair*, wenn sie durch die Anwendung einer fairen Transformationsprozedur entstanden sind.

Die erste Fairnebedingung stellt sicher, da nicht unendlich viele, mglicherweise zyklische  $\ddot{A}$ quivalenzumformungen ausgefhrt werden. Es ist auch unntig, eine Regel, die durch eine Umformung entstanden ist, noch einmal umzuformen; solche Transformationen knnen stets auch in einem Schritt zusammengefat werden.

Die zweite Bedingung ist bei endlichen Transformationsfolgen ohnehin eingehalten, und spielt in der Praxis keine Rolle. Sie stellt sicher, da eine Regel, die fr eine Substitution  $\sigma$  in dem Sinne persistent ist, da sie in allen bis auf endlich vielen der  $\mathcal{R}^n[\sigma]$  auftritt, auch in  $\mathcal{R}^\infty[\sigma]$  liegt. Diese Fairnebedingung knnte nur dann nicht eingehalten sein, wenn beispielsweise durch eine unendliche Folge von Substitutionen in  $\mathcal{R}^n$  immer allgemeinere Regeln auftreten, die gemeinsame „persistente“ Instanzen haben, von denen aber, da sie alle verschieden sind, keine persistent sind.

Die dritte Bedingung ist die wichtigste. Sie stellt sicher, da frher oder spter auf alle kritischen Paare die Kritisches-Paar-Regel angewendet wird, was fr die Konfluenzeigenschaften einer Vervollstndigung unerllich ist.

<sup>16</sup> Das heit  $\sigma$  erfllt die Constraints der beiden Regeln  $\mathbf{r}_1$  und  $\mathbf{r}_2$ , und es gilt

$$\{\mathbf{r}_1, \mathbf{r}_2\}\langle\sigma\rangle \subset (\mathcal{R}\langle\sigma\rangle)^\infty .$$

Vorausgesetzt, die Fairneßbedingungen werden eingehalten, können beliebige Heuristiken für die Auswahl der jeweils nächsten anzuwendenden Transformationsregel eingesetzt werden. Die für die Implementierung verwendete Transformationsprozedur ist in Abschnitt 6.4 beschrieben.

## 4.5 Berechnung von Normalformen

Hat man die Vervollständigung  $\mathcal{R}^\infty$  einer Menge  $E$  von Gleichungen berechnet, kann man ein  $E$ -Unifikationsproblem  $\langle E, s, t \rangle$  lösen, indem man Normalformen der Terme  $s$  und  $t$  bezüglich der Ableitbarkeitsrelation  $\Rightarrow_{\mathcal{R}^\infty}$  bestimmt. Meist hat ein Term nicht nur eine Normalform sondern mehrere — im allgemeinen sogar unendlich viele.

### Beispiel 4.5.1

Für  $\mathcal{R}^\infty = \{b \rightsquigarrow a \ll \epsilon, d \rightsquigarrow c \ll \epsilon\}$  hat  $x \ll \epsilon$  ( $\epsilon$  ist der leere Constraint) die Normalformen  $a \ll \langle \{x/b\}, \text{true} \rangle$ ,  $c \ll \langle \{x/d\}, \text{true} \rangle$  und  $x \ll \epsilon$  selbst.  $\square$

Wie das Beispiel zeigt, kann die Menge der Normalformen Redundanzen enthalten, denn der Gültigkeitsbereich der Normalform  $x \ll \epsilon$  ist nicht auf den Fall  $\sigma(x) \neq a$  und  $\sigma(x) \neq b$  beschränkt. Wollte man das vermeiden, müßte man im Beispiel statt  $x \ll \epsilon$  die Normalform  $x \ll \langle id, ((x \succ b) \vee (b \succ x)) \wedge ((x \succ d) \vee (d \succ x)) \rangle$  verwenden. Experimente haben aber gezeigt, daß der durch die Redundanzen entstehende Mehraufwand geringer ist, als derjenige, der durch die Handhabung der komplizierteren Ordnungsbedingungen in redundanzfreien Normalformen entsteht.

Wie die Transformationen zur Vervollständigung von Reduktionssystemen können auch die Regelanwendungen zur Berechnung von Normalformen als auf Mengen operierende Transformationen dargestellt werden. Die folgenden Normalisierungsregeln entsprechen im wesentlichen den auf Reduktionssystemen definierten Transformationsregeln (Abschnitt 4.4). Nur die Ableitungsregel hat dort keine direkte Entsprechung, ihre Eigenschaften, vor allem die Unterschiede zur Vereinfachungsregel, sind aber der der Kritisches-Paar-Regel ähnlich.

**Definition 4.5.2** (Normalisierungen, die Relation  $\vdash_{\mathcal{R}}$ )

In Abhängigkeit von einem Reduktionssystem  $\mathcal{R}$  sind auf Mengen  $\mathcal{T}$  von Termen mit Constraint die Normalisierungsregeln Löschung ( $L$ ), Äquivalenzumformung ( $\ddot{A}$ ), Subsumtion ( $S$ ), Vereinfachung ( $V$ ) und Ableitung ( $A$ ) definiert:

$$\begin{array}{ll}
 (L) & \frac{\mathcal{T} \cup \{(\forall \bar{x})(t \ll c)\}}{\mathcal{T}} \quad c \text{ inkonsistent} \\
 (\ddot{A}) & \frac{\mathcal{T} \cup \{(\forall \bar{x})(t \ll c)\}}{\mathcal{T} \cup \{(\forall \bar{x})(t\sigma \ll \langle \sigma, O \rangle) : \langle \sigma, O \rangle \in C\}} \quad \begin{array}{l} \text{Sat}(c) = \text{Sat}(C), \\ C \text{ endlich} \end{array} \\
 (S) & \frac{\mathcal{T} \cup \{t, t'\}}{\mathcal{T} \cup \{t\}} \quad t \text{ subsumiert } t' \\
 (V) & \frac{\mathcal{T} \cup \{t\}}{\mathcal{T} \cup \{t'\}} \quad t \Rightarrow_{\mathcal{R}} t' \\
 (A) & \frac{\mathcal{T} \cup \{t\}}{\mathcal{T} \cup \{t, t'\}} \quad t \Rightarrow_{\mathcal{R}} t'
 \end{array}$$

Seien  $\mathcal{T}$  und  $\mathcal{T}'$  Mengen von Termen mit Constraint.

$$\mathcal{T} \mid_{\mathcal{R}} \mathcal{T}'$$

bedeute, daß  $\mathcal{T}'$  durch Anwendung einer der Normalisierungen aus  $\mathcal{T}$  entsteht.

**Definition 4.5.3** (Normalisierungsprozedur, Menge von Normalformen)

Eine Normalisierungsprozedur legt fest, in welcher Reihenfolge die Normalisierungen (Def. 4.5.2) bei gegebenem Reduktionssystem  $\mathcal{R}$  und einer gegebenen Menge von Termen mit Constraints  $\mathcal{T} = \mathcal{T}^0$  anzuwenden sind, so daß eine Normalisierungsfolge

$$\mathcal{T}^0 \mid_{\mathcal{R}} \mathcal{T}^1 \mid_{\mathcal{R}} \mathcal{T}^2 \mid_{\mathcal{R}} \cdots$$

entsteht.

Die dadurch eindeutig bestimmte Menge

$$\mathcal{T}^\infty = \begin{cases} \mathcal{T}^m & \text{falls die Normalisierungsfolge von der Länge } m \text{ ist} \\ \bigcup_{k \geq 0} \bigcap_{m \geq k} \mathcal{T}^m & \text{falls die Normalisierungsfolge unendlich ist} \end{cases}$$

heiße Menge von Normalformen von  $\mathcal{T}$  (bezüglich  $\mathcal{R}$ ).

Wie bei der Vervollständigung von Reduktionssystemen unterliegt auch hier die Anwendung der Transformationsregeln gewissen Einschränkungen, muß nämlich in folgender Weise fair sein:

**Definition 4.5.4** (Fairneß einer Normalisierung)

Eine Normalisierungsprozedur heiße fair, wenn für jede Normalisierungsfolge

$$\mathcal{T}^0 \mid_{\mathcal{R}} \mathcal{T}^1 \mid_{\mathcal{R}} \mathcal{T}^2 \mid_{\mathcal{R}} \cdots$$

gilt:

1. Die Normalisierungsregel ( $\dot{A}$ ) wird nicht auf Terme angewendet, die selbst durch eine Äquivalenzumformung entstanden sind;
2. Für jede Grundsubstitution  $\sigma$  ist die Menge der  $\sigma$ -persistenten Terme

$$(\mathcal{T}\langle\sigma\rangle)^\infty = \begin{cases} \mathcal{T}^m\langle\sigma\rangle & \text{falls die Normalisierungsfolge endlich} \\ & \text{und von der Länge } m \text{ ist} \\ \bigcup_{k \geq 0} \bigcap_{m \geq k} \mathcal{T}^m\langle\sigma\rangle & \text{falls sie unendlich ist} \end{cases}$$

Teilmenge von  $\mathcal{T}^\infty\langle\sigma\rangle$ .

3. Für jeden Term  $\mathbf{t} \in \mathcal{T}^i$  ( $i \geq 0$ ), der für eine Grundsubstitution  $\sigma$   $\sigma$ -persistent ist,<sup>17</sup> und aus dem ein Term  $\mathbf{s}$  ableitbar ist, gibt es einen Index  $j$ , so daß die Menge von Normalformen  $\mathcal{T}^{j+1}$  in der Normalisierungsfolge durch Anwendung der Vereinfachung (V) oder der Ableitung (A) (Def. 4.5.3) auf den entsprechenden Term  $\mathbf{t}' \in \mathcal{T}^j$  entstanden ist, und  $\mathbf{s}' \in \mathcal{T}^{j+1}$  gilt ( $\mathbf{s}$  und  $\mathbf{s}'$  haben die gleichen Instanzen bezüglich  $\sigma$ ).

<sup>17</sup> Das heißt  $\sigma$  erfüllt den Constraint des Terms  $\mathbf{t}$ , und es gilt  $\{\mathbf{t}\}\langle\sigma\rangle \subset (\mathcal{T}\langle\sigma\rangle)^\infty$ .

*Mengen von Normalformen und Normalisierungsfolgen heißen fair, wenn sie durch die Anwendung einer fairen Normalisierungsprozedur entstanden sind.*

Die bei der Berechnung von Normalformen einzuhaltenen Fairneßbedingungen sind denen für eine faire Vervollständigung (Def. 4.4.10) sehr ähnlich. Das gilt insbesondere für die ersten beiden, eher technischen Bedingungen.

Die dritte Bedingung stellt hier sicher, daß jede mögliche und nicht durch Vereinfachungen obsolet werdende Regelanwendung schließlich ausgeführt wird.

### Beispiel 4.5.5

Sei  $\succ_x$  die alphabetische LPO,  $\mathcal{R}$  enthalte die beiden Regeln

$$\mathbf{r}_1 = (\forall x)(f(x) \rightsquigarrow x \ll \epsilon) \quad \text{und} \quad \mathbf{r}_2 = (y \rightsquigarrow b \ll \langle id, y \succ b \rangle) ,$$

und es sei

$$\mathcal{T}^0 = \{f(a) \ll \epsilon\} .$$

Dann ist

$$\mathcal{T}^\infty = \{a \ll \epsilon, \\ b \ll \langle \{y/f(a)\}, \text{true} \rangle\}$$

eine faire Menge von Normalformen von  $\mathcal{T}^0$ .

Die Regel  $\mathbf{r}_2$  ist auf  $a \ll \epsilon$  nicht anwendbar und  $b \ll \langle \{y/a\}, a \succ b \rangle$  keine Normalform, da der Constraint  $\langle \{y/a\}, a \succ b \rangle$  inkonsistent ist.  $\square$

## 4.6 Vermischen von Vervollständigung und Normalisierung

Nun stehen die notwendigen Mittel zur Verfügung, ein beliebiges gemischtes  $E$ -Unifikationsproblem  $\langle E, s, t \rangle$  zu lösen, indem man nämlich zunächst eine faire Vervollständigung  $\mathcal{R}^\infty$  des zu  $E$  gehörenden initialen Reduktionssystems  $\mathcal{R}^0$  bestimmt, und dieses dann benutzt, um die Mengen von Normalformen  $\mathcal{S}^\infty$  und  $\mathcal{T}^\infty$  der Terme mit Constraint

$$\{s \ll \epsilon\} \quad \text{und} \quad \{t \ll \epsilon\}$$

(bezüglich  $\mathcal{R}^\infty$ ) zu berechnen. Die Mengen  $\mathcal{S}^\infty$  und  $\mathcal{T}^\infty$  stellen in gewisser Weise alle Normalformen von  $s\sigma$  und  $t\sigma$  (bezüglich  $E\sigma$ ) für verschiedene Substitutionen  $\sigma$  dar. Indem man in ihnen nach zueinander passenden Normalformen von  $s$  und  $t$  sucht, d. h., nach Normalformen, die gemeinsame Instanzen haben, kann man eine Menge von Constraints bestimmen, die eine grund-vollständige Menge von Unifikatoren des  $E$ -Unifikationsproblems  $\langle E, s, t \rangle$  darstellt:

### Definition 4.6.1 (Lösungsmengen $\mathcal{C}_\infty^{j,k}$ , $\mathcal{C}_\infty$ )

Sei  $\langle E, s, t \rangle$  ein gemischtes  $E$ -Unifikationsproblem und  $\mathcal{R}^\infty$  eine faire Vervollständigung des zu  $E$  gehörenden initialen Reduktionssystems  $\mathcal{R}^0$ . Die Mengen von Termen mit Constraint  $\mathcal{S}_i^\infty$  und  $\mathcal{T}_i^\infty$  ( $i = 0, 1, 2, \dots, \infty$ ) seien die durch  $\mathcal{R}^i$  definierten fairen Menge von Normalformen von  $\mathcal{S}_i^0$  bzw.  $\mathcal{T}_i^0$ , wobei

$$\mathcal{S}_i^0 = \mathcal{S}^0 = \{s \ll \epsilon\} \quad \text{und} \quad \mathcal{T}_i^0 = \mathcal{T}^0 = \{t \ll \epsilon\} .$$

Dann sei  $\mathcal{C}_i^{j,k}(\langle E, s, t \rangle)$  ( $i, j, k = 0, 1, 2, \dots, \infty$ ) die Menge von Constraints

$$\{c_1 \sqcup c_2 \sqcup \langle \mu, \text{true} \rangle : (\forall \bar{x})(r_1 \ll c_1) \in \mathcal{S}_i^j, (\forall \bar{y})(r_2 \ll c_2) \in \mathcal{T}_i^k, \\ r_1 \text{ und } r_2 \text{ unifizierbar mit MGU } \mu \} .$$

Ferner sei

$$\mathcal{C}_i(\langle E, s, t \rangle) = \mathcal{C}_i^{\infty, \infty}(\langle E, s, t \rangle) .$$

### Beispiel 4.6.2

Als Beispiel soll noch einmal das schon in Beispiel 3.6.19 verwendete  $E$ -Unifikationsproblem

$$\langle E, s, t \rangle = \langle \{f(a) \approx a, g(g(x)) \approx f(a)\}, g(g(g(x))), x \rangle$$

dienen (siehe auch Abschnitt 7.1.2).

Verwendet man die alphabetische LPO, dann ist

$$\mathcal{R}^3 = \mathcal{R}^\infty = \{f(a) \rightsquigarrow a \ll \epsilon, \\ g(g(x)) \rightsquigarrow a \ll \epsilon\}$$

eine faire Vervollständigung von  $E$ . Damit ist

$$\mathcal{S}_3^1 = \mathcal{S}_3^\infty = \{g(a) \ll \epsilon\} \\ \mathcal{T}_3^0 = \mathcal{T}_3^\infty = \{x \ll \epsilon\} .$$

Also ist

$$\mathcal{C}_i^{j,k} = \{\{\{x/g(a)\}, \text{true}\}\} \quad (i \geq 3, j \geq 1, k \geq 0) .$$

Da  $\mathcal{S}_3^0$  und  $\mathcal{T}_3^0$  keine kombinierbaren Terme enthalten, ist  $\mathcal{C}_3^{0,0} = \emptyset$ .  $\square$

Die Mengen von Constraints  $\mathcal{C}_i$  approximieren die Menge  $\mathcal{C}_\infty$ . In Abschnitt 4.8 wird bewiesen, daß die Menge

$$\mathcal{U}_\infty(\langle E, s, t \rangle) = \text{Sat}^*(\mathcal{C}_\infty(\langle E, s, t \rangle))$$

für jedes  $E$ -Unifikationsproblem  $\langle E, s, t \rangle$  eine bezüglich der Subsumtionsrelation  $\leq_E$  grund-vollständige Menge von Unifikatoren ist. Das folgende Beispiel zeigt, daß  $\mathcal{U}_\infty$  im allgemeinen tatsächlich nur grund-vollständig, nicht aber vollständig ist:

### Beispiel 4.6.3

Sei  $\langle E, s, t \rangle = \langle \{x \approx y\}, x, y \rangle$ . Eine faire Vervollständigung ist beispielweise

$$\mathcal{R}^\infty = \mathcal{R}^0 = \{x \rightsquigarrow y \ll \langle id, x \succ y \rangle, y \rightsquigarrow x \ll \langle id, y \succ x \rangle\} .$$

Die damit konstruierte Lösungsmenge gemäß Definition 4.6.1 ist

$$\mathcal{C}_\infty(\langle E, s, t \rangle) = \{\{\{x/y\}, \text{true}\}, \langle id, x \succ y \rangle, \langle id, y \succ x \rangle\} .$$

Wie jede Substitution ist die allgemeinste Substitution  $id$  eine Lösung des Problems; sie erfüllt aber keinen der Constraints in  $\mathcal{C}_\infty$  und liegt nicht in  $\mathcal{U}_\infty(\langle E, s, t \rangle)$ ; darum ist  $\mathcal{U}_\infty$  nicht vollständig.  $\mathcal{U}_\infty$  enthält aber alle Grundsubstitutionen und ist also grund-vollständig.  $\square$

Das Problem ist nun aber, daß die Menge  $\mathcal{U}_\infty(\langle E, s, t \rangle)$  im allgemeinen nicht (ohne weiteres) aufzählbar ist.<sup>18</sup> Dazu bräuchte man nämlich die faire Vervollständigung  $\mathcal{R}^\infty$  als ganzes;  $\mathcal{R}^\infty$  ist aber selbst nur aufzählbar.

Dieses Problem könnte man lösen, indem man die Mengen  $\mathcal{C}_i^{j,k}(\langle E, s, t \rangle)$  für alle  $i, j, k \geq 0$  berechnet und nicht nur  $\mathcal{C}_\infty = \mathcal{C}_{\infty, \infty}$ .<sup>19</sup> Diese Mengen sind, da man zu ihrer Konstruktion die endlichen Reduktionssysteme  $\mathcal{R}^i$  verwendet, berechenbar. Daraus und aus Definition 4.2.13 folgt, daß die Menge von Substitutionen

$$\mathcal{U}'_\infty(\langle E, s, t \rangle) = \text{Sat}^*(\bigcup_{i,j,k \geq 0} \mathcal{C}_i^{j,k}(\langle E, s, t \rangle))$$

aufzählbar ist.

Mit der Menge  $\mathcal{U}'_\infty$  statt  $\mathcal{U}_\infty$  bleibt das Verfahren vollständig, denn jeder Constraint  $c \in \mathcal{C}_\infty(\langle E, s, t \rangle)$  muß sich in *endlich* vielen Schritten konstruieren lassen,<sup>20</sup> weswegen

$$\mathcal{C}_\infty(\langle E, s, t \rangle) \subset \bigcup_{i,j,k \geq 0} \mathcal{C}_i^{j,k}(\langle E, s, t \rangle)$$

gilt.

Die Umkehrung der obigen Inklusion gilt nicht (sonst wäre  $\mathcal{C}_\infty(\langle E, s, t \rangle)$  aufzählbar). Es kann nämlich sein, daß eine Reduktionsregel in  $\mathcal{R}^i$ , die bei der Konstruktion eines Elementes von  $\mathcal{C}_i^{j,k}(\langle E, s, t \rangle)$  verwendet wird, nicht persistent ist, in  $\mathcal{R}^\infty$  also nicht mehr auftritt. Darum muß für den Beweis, daß das Verfahren *korrekt* bleibt, jede einzelne der Mengen  $\mathcal{C}_i^{j,k}$  — und nicht etwa nur  $\mathcal{C}_\infty$  — als korrekt nachgewiesen werden (was im nächsten Abschnitt geschieht).

Mit der Berechnung der Menge  $\mathcal{U}'_\infty$  statt  $\mathcal{U}_\infty$  handelt man sich aber nun ein neues Problem ein: sie ist ineffizient. Man muß nun nämlich die Mengen  $\mathcal{S}_i^j$  und  $\mathcal{T}_i^k$  für alle<sup>21</sup>  $i \geq 0$  berechnen. Bei praktisch allen Unifikationsproblemen bedeutet das die Ableitung einer großen Menge redundanter Information. In der Regel werden sich  $\mathcal{S}_i^j$  und  $\mathcal{S}_{i+1}^j$  kaum unterscheiden, weil sich auch  $\mathcal{R}^i$  und  $\mathcal{R}^{i+1}$  nur wenig, nämlich nur in den durch die Transformation betroffenen Regeln unterscheiden.

Es ist unerlässlich diese Redundanzen zu vermeiden; sonst würde eine Implementierung des Verfahrens schon bei einfachen Unifikationsproblemen scheitern.  $\mathcal{S}_\infty^j$  und  $\mathcal{T}_\infty^k$  zu berechnen ist, wie schon gesagt, nicht möglich, weil  $\mathcal{R}^\infty$  nur aufzählbar ist. Es würde auch genügen, statt dessen  $\mathcal{S}_i^j$  und  $\mathcal{T}_i^k$  für ein *hinreichend* großes  $i$  zu berechnen. Leider gibt es aber keine Methode, im voraus zu bestimmen, was „hinreichend groß“ bedeutet.

Es gibt aber eine andere Möglichkeit, nämlich die Vervollständigung mit der Normalisierung der Terme in folgender Weise zu vermischen:

<sup>18</sup> Jedenfalls nicht mit Hilfe des beschriebenen Verfahrens, daß  $\mathcal{U}_\infty(\langle E, s, t \rangle)$  tatsächlich nicht aufzählbar ist, ist damit nicht bewiesen.

<sup>19</sup> Es genügte auch, hinreichend viele dieser Mengen zu bestimmen, so daß nämlich für beliebige  $i, j, k \geq 0$  eine Menge  $\mathcal{C}_{i',j',k'}^{j,k}$  berechnet wird mit  $i' \geq i$ ,  $j' \geq j$ ,  $k' \geq k$ .

<sup>20</sup> Alle in den unendlichen Mengen  $\mathcal{R}^\infty$  und  $\mathcal{C}_\infty(\langle E, s, t \rangle)$  enthaltenen Elemente entstehen durch endliche Transformations- und Normalisierungsfolgen mit endlichen Beweisen — was der Tatsache, daß  $\mathcal{C}_\infty(\langle E, s, t \rangle)$  nicht aufzählbar ist, nicht widerspricht.

<sup>21</sup> Die Berechnung für hinreichend viele statt alle  $i$  genügt (siehe Fußnote 19).



**Definition 4.6.4** (Vervollständigungs- und Normalisierungsfolge, -prozedur)

Eine Folge  $(\langle \mathcal{R}^i, \mathcal{T}^i \rangle)_{(i \geq 0)}$  von Paaren, bestehend aus Reduktionssystemen mit Constraints  $\mathcal{R}^i$  und Mengen von Termen mit Constraint  $\mathcal{T}^i$ , heie eine Vervollstndigungs- und Normalisierungsfolge, wenn fr alle  $i \geq 0$  genau eine der folgenden Bedingungen erfllt ist:

- $\mathcal{R}^i \mid\text{-} \mathcal{R}^{i+1}$  (Def. 4.4.9) und  $\mathcal{T}^i = \mathcal{T}^{i+1}$ ;
- $\mathcal{R}^i = \mathcal{R}^{i+1}$  und  $\mathcal{T}^i \mid\text{-}_{\mathcal{R}^i} \mathcal{T}^{i+1}$  (Def. 4.5.3);
- $\mathcal{R}^i = \mathcal{R}^{i+1}$  und  $\mathcal{T}^i = \mathcal{T}^{i+1}$ .

Eine Vervollstndigungs- und Normalisierungsprozedur legt fest, in welcher Reihenfolge Transformationen und Normalisierungen bei einem gegebenen Paar  $\langle \mathcal{R}^0, \mathcal{T}^0 \rangle$  anzuwenden sind, so da eine Vervollstndigungs- und Normalisierungsfolge entsteht.

Dadurch sind Mengen  $\mathcal{R}^\infty$  persistenter Regeln und  $\mathcal{T}^\infty$  persistenter Normalformen wie in den Definitionen 4.4.9 und 4.5.3 eindeutig bestimmt,

Die dritte Mglichkeit bei der Bildung einer Vervollstndigungs- und Normalisierungsfolge, sowohl  $\mathcal{R}^i = \mathcal{R}^{i+1}$  als auch  $\mathcal{T}^i = \mathcal{T}^{i+1}$  unverndert zu lassen, ist fr den Fall zugelassen, da mehrere Unifikationsprobleme zugleich betrachtet werden, und beim bergang von  $i$  auf  $i+1$  eine andere Menge von Normalformen verndert wird.

Auch wenn die Vervollstndigung und die Normalisierung vermischt werden, mssen die Fairnebedingungen aus den Definitionen 4.4.10 und 4.5.4 eingehalten werden. Diese nehmen nun die folgende Form an:

**Definition 4.6.5** (Faire Vervollstndigungs- und Normalisierungsprozedur)

Eine Vervollstndigungs- und Normalisierungsprozedur heie fair, wenn fr jede Normalisierungsfolge  $(\langle \mathcal{R}^i, \mathcal{T}^i \rangle)_{(i \geq 0)}$  die Folge der Reduktionssysteme  $\mathcal{R}^i$  ( $i \geq 0$ ) eine faire Vervollstndigungsfolge ist (Def. 4.4.10), und fr die Folge der Mengen  $\mathcal{T}^i$  von Normalformen ( $i \geq 0$ ) gilt:

1. Die Normalisierungsregel ( $\ddot{A}$ ) wird nicht auf Terme angewendet, die selbst durch eine quivalenzumformung entstanden sind;
2. Fr jede Grundsubstitution  $\sigma$  ist die Menge der  $\sigma$ -persistenten Terme

$$(\mathcal{T}(\sigma))^\infty = \begin{cases} \mathcal{T}^m(\sigma) & \text{falls die Normalisierungsfolge endlich} \\ & \text{und von der Lnge } m \text{ ist} \\ \bigcup_{k \geq 0} \bigcap_{m \geq k} \mathcal{R}^m(\sigma) & \text{falls sie unendlich ist} \end{cases}$$

Teilmenge von  $\mathcal{T}^\infty(\sigma)$ .

3. Fr jeden Term  $\mathbf{t} \in \mathcal{T}^i$ , ( $i \geq 0$ ), der fr eine Grundsubstitution  $\sigma$   $\sigma$ -persistent ist, und aus dem mit Hilfe einer  $\sigma$ -persistenten Regel  $\mathbf{r} \in \mathcal{R}^i$  ein Term  $\mathbf{s}$  ableitbar ist, gibt es einen Index  $j$ , so da die Menge von Normalformen  $\mathcal{T}^{j+1}$  durch Anwendung der Vereinfachung (V) oder der Ableitung (A) (Def. 4.5.3) auf  $\mathbf{t}' \in \mathcal{T}^j$  entstanden ist, und  $\mathbf{s}' \in \mathcal{T}^{j+1}$  gilt ( $\mathbf{s}$  und  $\mathbf{s}'$  haben bezglich  $\sigma$  die gleichen Instanzen).

Auch mit Hilfe einer vermischten Vervollständigung und Normalisierung kann man zu einem gegebenen  $E$ -Unifikationsproblem  $\langle E, s, t \rangle$  Mengen  $\mathcal{C}^i$  von Constraints berechnen (ähnlich wie die Mengen  $\mathcal{C}_i^{j,k}$  aus Definition 4.6.1), die nur von Unifikatoren des Problems erfüllt werden, und die — vorausgesetzt die Prozedur ist fair — eine Menge  $\mathcal{C}$  approximieren, für die  $\text{Sat}(\mathcal{C})$  eine grund-vollständige Menge von Unifikatoren ist:

**Definition 4.6.6** (Lösungsmengen  $\mathcal{C}^i, \mathcal{C}$ )

Sei  $\langle E, s, t \rangle$  ein gemischtes  $E$ -Unifikationsproblem,  $\mathcal{R}^0$  das zu  $E$  gehörenden initiale Reduktionssystem,

$$\mathcal{S}^0 = \{s \ll \epsilon\} \quad \text{und} \quad \mathcal{T}^0 = \{t \ll \epsilon\} .$$

Ferner seien  $(\langle \mathcal{R}^i, \mathcal{S}^i \rangle)_{(i \geq 0)}$  und  $(\langle \mathcal{R}^i, \mathcal{T}^i \rangle)_{(i \geq 0)}$  faire Vervollständigungs und Normalisierungsfolgen.

Dann sei  $\mathcal{C}^i(\langle E, s, t \rangle)$  ( $i = 0, 1, 2, \dots, \infty$ ) die Menge von Constraints

$$\{c_1 \sqcup c_2 \sqcup \langle \mu, \text{true} \rangle : (\forall \bar{x})(r_1 \ll c_1) \in \mathcal{S}^i, (\forall \bar{y})(r_2 \ll c_2) \in \mathcal{T}^i, \\ r_1 \text{ und } r_2 \text{ unifizierbar mit MGU } \mu \} ,$$

und

$$\mathcal{C}(\langle E, s, t \rangle) = \bigcup_{i \geq 0} \mathcal{C}^i(\langle E, s, t \rangle) .$$

Die Menge  $\mathcal{C}(\langle E, s, t \rangle)$  ist also die Vereinigung der  $\mathcal{C}^i(\langle E, s, t \rangle)$  und nicht etwa gleich  $\mathcal{C}^\infty$  (diese Definition wäre auch möglich, würde aber wieder zu dem Problem führen, daß  $\mathcal{C} = \mathcal{C}^\infty$  im allgemeinen nicht aufzählbar wäre).

Im Gegensatz zu  $\bigcup_{i,j,k \geq 0} \mathcal{C}_i^{j,k}$  enthält  $\mathcal{C} = \bigcup_{i \geq 0} \mathcal{C}^i$  in der Regel nur wenige Redundanzen und ist darum für die Implementierung geeignet.

## 4.7 Korrektheit

In diesem Abschnitt wird bewiesen, daß die Mengen  $\mathcal{C}_i^{j,k}(\langle E, s, t \rangle)$  und  $\mathcal{C}(\langle E, s, t \rangle)$  aus den Definition 4.6.1 und 4.6.6 für  $i, j, k \in \{0, 1, 2, \dots, \infty\}$  korrekt sind, d. h., daß jede Substitution in

$$\text{Sat}(\mathcal{C}_i^{j,k}(\langle E, s, t \rangle)) \quad \text{und} \quad \text{Sat}(\mathcal{C}^i(\langle E, s, t \rangle))$$

eine Lösung von  $\langle E, s, t \rangle$  ist.

Dazu muß man zeigen, daß die Ableitbarkeitsrelation  $\xrightarrow{*}_{\mathcal{R}^i}$  für jedes Reduktionssystem einer Vervollständigungsfolge korrekt ist, daß also jede Ableitung der durch die Gleichungsmenge  $E$  definierten Gleichungstheorie entspricht (in dem in Lemma 4.7.4 präzisierten Sinne). Man muß sogar etwas mehr beweisen, nämlich daß der „symmetrische Abschluß“ von  $\xrightarrow{*}_{\mathcal{R}^i}$  korrekt ist. Mit „symmetrischer Abschluß“ ist hier allerdings nicht der mengentheoretische Abschluß  $\overset{*}{\Leftrightarrow}_{\mathcal{R}^i}$  gemeint, sondern die Relation, die man erhält, wenn man bei der Ableitung die Orientierung der Regeln nicht berücksichtigt:

**Definition 4.7.1** (Inverse einer Reduktionsregel und eines Reduktionssystems, das System  $\mathcal{R}^\pm$ )

*Die Inverse einer Reduktionsregel mit Constraint*

$$\mathbf{r} = (\forall \bar{x})(s \rightsquigarrow t \ll c)$$

*ist die Regel*

$$\mathbf{r}^- = (\forall \bar{x})(t \rightsquigarrow s \ll c) .$$

*Das inverse System zu einem Reduktionssystem  $\mathcal{R}$  ist das System*

$$\mathcal{R}^- = \{\mathbf{r}^- : \mathbf{r} \in \mathcal{R}\} .$$

*Das System  $\mathcal{R}^\pm$  ist die Vereinigung von  $\mathcal{R}$  mit seiner Inversen:*

$$\mathcal{R}^\pm = \mathcal{R} \cup \mathcal{R}^- .$$

Bei der Bildung der Inversen einer Reduktionsregel wird also nur ihre Orientierung verändert; ihr Constraint wird beibehalten (also nicht etwa negiert), denn der Gültigkeitsbereich muß der gleiche bleiben. Wie das folgende Beispiel zeigt, gilt im allgemeinen  $\overset{*}{\Rightarrow}_{\mathcal{R}} \neq \overset{*}{\Rightarrow}_{\mathcal{R}^\pm}$ .

**Beispiel 4.7.2**

Sei  $\mathcal{R} = \{x \rightsquigarrow a \ll \epsilon\}$ . Dann gilt

$$(b \ll \epsilon) \Rightarrow_{\mathcal{R}} (a \ll \langle \{x/b\}, \text{true} \rangle) ,$$

und es gilt

$$(a \ll \langle \{x/b\}, \text{true} \rangle) \Rightarrow_{\mathcal{R}^-} (b \ll \langle \{x/b\}, \text{true} \rangle) .$$

Betrachtet man dagegen den symmetrischen Abschluß von  $\Rightarrow_{\mathcal{R}}$ , dann gilt

$$(a \ll \langle \{x/b\}, \text{true} \rangle) \Leftrightarrow_{\mathcal{R}} (b \ll \epsilon) ,$$

was ein unerwünschter Effekt ist, da bei einer Ableitung der an den Term angeheftete Constraint niemals allgemeiner werden darf.  $\square$

Zunächst wird ein relativ einfach zu überprüfendes Kriterium für die Korrektheit eines Reduktionssystems bewiesen:

**Definition 4.7.3** (Korrektes Reduktionssystem)

*Ein Reduktionssystem  $\mathcal{R}$  heie korrekt bezüglich einer Menge  $E$  von Gleichungen, falls für jede Reduktionsregel*

$$(\forall x)(l \rightsquigarrow r \ll c) \in \mathcal{R}^\pm$$

*gilt:*

*Wenn eine Substitution  $\sigma$  den Constraint  $c$  erfüllt, dann ist sie eine Lösung des gemischten  $E$ -Unifikationsproblems  $\langle E, l, r \rangle$ .*

**Lemma 4.7.4**

Das Reduktionssystem  $\mathcal{R}$  sei bezüglich einer Menge  $E$  von Gleichungen korrekt (Def. 4.7.3).

Gilt für Terme mit Constraint  $\mathbf{s} = (\forall \bar{x})(s \ll c_s)$  und  $\mathbf{t} = (\forall \bar{y})(t \ll c_t)$

$$\mathbf{s} \xrightarrow{\mathcal{R}^\pm}^* \mathbf{t} \quad \text{oder} \quad \mathbf{s} \xrightarrow{\mathcal{R}}^* \mathbf{t} ,$$

dann ist jede Substitution  $\sigma$ , die den Constraint  $c_t$  erfüllt, eine Lösung des gemischten  $E$ -Unifikationsproblems  $\langle E, s, t \rangle$ .

**Beweis:**

Den Fall, daß  $\mathbf{s} \xrightarrow{\mathcal{R}}^* \mathbf{t}$  gilt, braucht man nicht gesondert zu betrachten, denn wegen  $\mathcal{R} \subset \mathcal{R}^\pm$  gilt, falls  $\mathbf{s} \xrightarrow{\mathcal{R}}^* \mathbf{t}$ , auch  $\mathbf{s} \xrightarrow{\mathcal{R}^\pm}^* \mathbf{t}$ .

Gelte zunächst  $\mathbf{s} \Rightarrow_{\mathcal{R}^\pm} \mathbf{t}$  mit  $\langle \mathbf{r}, p, \mu \rangle$  als Rechtfertigung, wobei  $\mathbf{r} = (\forall \bar{z})(l \rightsquigarrow r \ll c_r)$ .

1. Da  $\sigma$  den Constraint  $c_t$  erfüllt, und also eine Spezialisierung von  $\mu$  sein muß, gilt  $s\sigma = s\sigma[p/l\sigma]$  und  $t\sigma = t\sigma[p/r\sigma]$ ;
2. Da  $\mathcal{R}$  korrekt ist bezüglich  $E$ , ist  $\sigma$  Lösung von  $\langle E, l, r \rangle$  (Def. 4.7.3), und also gilt  $l\sigma \xrightarrow{E\sigma}^* r\sigma$ .

Aus der Monotonie von  $\xrightarrow{E\sigma}^*$  bezüglich der Termstruktur (Satz 3.2.7) folgt

$$s\sigma = s\sigma[p/l\sigma] \xrightarrow{E\sigma}^* t\sigma[p/r\sigma] = t\sigma .$$

Sei nun

$$\mathbf{s} = \mathbf{t}_0 \Rightarrow_{\mathcal{R}^\pm} \mathbf{t}_1 \Rightarrow_{\mathcal{R}^\pm} \cdots \Rightarrow_{\mathcal{R}^\pm} \mathbf{t}_n = \mathbf{t}$$

ein Beweis für  $\mathbf{s} \xrightarrow{\mathcal{R}^\pm}^* \mathbf{t}$ . Da die Constraints bei der Ableitung spezieller werden, erfüllt  $\sigma$  nicht nur  $c_t$ , sondern jeden Constraint  $c_i$  eines Terms  $\mathbf{t}_i$  ( $1 \leq i \leq n$ ). Daher folgt aus dem bisher bewiesenen

$$s\sigma = t_0\sigma \xrightarrow{E\sigma}^* t_1\sigma \xrightarrow{E\sigma}^* \cdots \xrightarrow{E\sigma}^* t_n\sigma = t\sigma ,$$

also  $s\sigma \xrightarrow{E\sigma}^* t\sigma$ , und daraus, daß  $\sigma$  eine Lösung des  $E$ -Unifikationsproblems  $\langle E, s, t \rangle$  ist (Korollar 3.2.3). ■

Man kann zeigen, daß das initiale System zu einer Menge von Gleichungen korrekt ist, und daß die Korrektheit bei Transformationen erhalten bleibt, also auch alle Reduktionssysteme in einer Vervollständigungsfolge korrekt sind:

**Lemma 4.7.5**

Sei  $E$  eine Menge von Gleichungen und  $\mathcal{R}^0$  das initiale Reduktionssystem zu  $E$ . Dann ist jedes Reduktionssystem  $\mathcal{R}^n$  ( $n \geq 0$ ) in einer von  $\mathcal{R}^0$  ausgehenden Vervollständigungsfolge

$$\mathcal{R}^0 \mid\!-\! \mathcal{R}^1 \mid\!-\! \mathcal{R}^2 \mid\!-\! \cdots$$

und auch die daraus resultierende Vervollständigung  $\mathcal{R}^\infty$  (Def. 4.4.9) korrekt bezüglich  $E$ .

**Beweis:**

Da die Korrektheit eines Reduktionssystems aus der Korrektheit jeder in ihm (und seiner Inversen) enthaltenen Regel folgt (Def. 4.7.3), und da gemäß Definition 4.4.9 jede Regel in  $\mathcal{R}^\infty$  schon in einem (sogar unendlich vielen) der  $\mathcal{R}^n$  ( $n \geq 0$ ) enthalten sein muß, genügt es, die Korrektheit der Systeme in der Vervollständigungsfolge zu zeigen. Dies geschieht durch vollständige Induktion über  $n$ :

„ $n = 0$ “:

Ist  $(\forall x)(l \rightsquigarrow r \ll c) \in \mathcal{R}^{0^\pm}$ , dann ist, da  $\mathcal{R}^0$  das initiale System zu  $E$  ist (Def. 4.3.3),

$$(\forall x)(l \approx r) \in E \quad \text{oder} \quad (\forall x)(r \approx l) \in E .$$

Darum ist die leere Substitution  $id$  ein Unifikator von  $\langle E, l, r \rangle$ , und wegen Satz 3.2.6 auch jede den Constraint  $c$  erfüllende Substitution.

„ $n \rightarrow n + 1$ “:

*Fall 1:* Sei  $\mathcal{R}^{n+1}$  aus  $\mathcal{R}^n$  durch eine Transformation entstanden, die keine neue Reduktionsregel hinzufügt (Löschung, Subsumtion), dann folgt die Korrektheit des Systems  $\mathcal{R}^{n+1}$  trivialerweise aus der von  $\mathcal{R}^n$ .

*Fall 2:* Sei  $\mathcal{R}^{n+1}$  aus  $\mathcal{R}^n$  durch eine Äquivalenzumformung (Def. 4.4.4) entstanden, und sei

$$(\forall \bar{x})(l\sigma \rightsquigarrow r\sigma \ll \langle \sigma, O \rangle) \in \mathcal{R}^{n+1}$$

eine beliebige der neu hinzugekommenen, aus  $(\forall \bar{x})(l \rightsquigarrow r \ll c) \in \mathcal{R}^n$  entstandenen Reduktionsregeln. Erfüllt eine Substitution  $\tau$  den Constraint  $\langle \sigma, O \rangle$ , dann erfüllt sie wegen  $\text{Sat}(\langle \sigma, O \rangle) \subset \text{Sat}(c)$  auch den Constraint  $c$ . Aus der Korrektheit von  $\mathcal{R}^n$  folgt

$$l\tau \xleftrightarrow{E\tau} r\tau ,$$

und da  $\sigma$  idempotent und  $\tau$  eine Spezialisierung von  $\sigma$  ist,

$$l\sigma\tau \xleftrightarrow{E\tau} r\sigma\tau .$$

Also ist  $\tau$  Unifikator von  $\langle E, l\sigma, r\sigma \rangle$ . Die Korrektheit der Inversen der neuen Regeln folgt analog.

*Fall 3:* Sei  $\mathcal{R}^{n+1}$  aus  $\mathcal{R}^n$  durch Anwendung der Kritsches-Paar-, der Vereinfachungs- oder der Kombinationsregel entstanden (Def. 4.4.7).

$$\mathbf{r}_{neu} = (\forall \bar{x})(\forall \bar{y})(s_{neu} \rightsquigarrow t_{neu} \ll c) \in \mathcal{R}^{n+1}$$

sei eine neue Regel oder die Inverse einer neuen Regel. Dann gibt es gemäß Definition 4.4.7 Regeln  $\mathbf{r}_1 = (\forall \bar{x})(s \rightsquigarrow t \ll c_1)$  und  $\mathbf{r}_2 = (\forall \bar{y})(l \rightsquigarrow r \ll c_2)$  in  $(\mathcal{R}^n)^\pm$ , eine Position  $p$  in  $s \rightsquigarrow t$  und einen idempotenten Unifikator  $\mu$ , so daß entweder

- $s\mu = (s\mu)[p/l\mu]$ ,  $s_{neu}\mu = (s\mu)[p/r\mu]$ ,  $t\mu = t_{neu}\mu$  (wenn  $p$  eine Position in  $s$  ist),  
oder
- $s\mu = s_{neu}\mu$ ,  $t\mu = (t\mu)[p/l\mu]$ ,  $t_{neu}\mu = (t\mu)[p/r\mu]$ , (wenn  $p$  eine Position in  $t$  ist).

Erfüllt eine Substitution  $\tau$  den Constraint  $c$ , dann erfüllt sie auch die Constraints  $c_1$  und  $c_2$ , die in  $c$  eingehen, und es gilt  $\mu \leq \tau$ . Aus der Korrektheit von  $\mathcal{R}^n$  folgt

$$s\tau \xleftrightarrow{E\tau} t\tau \quad \text{und} \quad l\tau \xleftrightarrow{E\tau} r\tau ,$$

und daraus mit Satz 3.2.7 entweder

- $s_{neu}\tau = (s\tau)[p/r\tau] \xleftrightarrow{*}_{E\tau} (s\tau)[p/l\tau] = s\tau \xleftrightarrow{*}_{E\tau} t\tau = t_{neu}\tau$ , oder
- $t_{neu}\tau = (t\tau)[p/r\tau] \xleftrightarrow{*}_{E\tau} (t\tau)[p/l\tau] = t\tau \xleftrightarrow{*}_{E\tau} s\tau = s_{neu}\tau$ .

In jedem Fall gilt also  $s_{neu}\tau \xleftrightarrow{*}_{E\tau} t_{neu}\tau$ , was zu zeigen war.  $\blacksquare$

Mit Hilfe des gerade bewiesenen Lemmas kann man nun zeigen, daß die Berechnung von Normalformen eines Termes mit Constraint gemäß Definition 4.5.3 korrekt ist:

#### Satz 4.7.6

Sei  $E$  eine Menge von Gleichungen,  $\mathcal{R}$  und  $\mathcal{R}^0, \mathcal{R}^1, \mathcal{R}^2, \dots$  bezüglich  $E$  korrekte Reduktionssysteme, und

$$\mathcal{T}^0 = \{s \ll \epsilon\} .$$

Ferner sei

$$\mathcal{T}^0 \vdash_{\mathcal{R}} \mathcal{T}^1 \vdash_{\mathcal{R}} \mathcal{T}^2 \vdash_{\mathcal{R}} \dots$$

eine von  $\mathcal{T}^0$  ausgehende Normalisierungsfolge (Def. 4.5.3) oder aber

$$((\mathcal{R}^i, \mathcal{T}^i))_{i \geq 0}$$

eine Vervollständigungs- und Normalisierungsfolge (Def. 4.6.5).

Dann gilt für jeden Term mit Constraint  $(\forall \bar{x})(t \ll c)$  in einer der Mengen  $\mathcal{T}^n$  ( $n \geq 0$ ) und in der Normalisierung  $\mathcal{T}^\infty$  (Def. 4.5.3): Erfüllt die Substitution  $\sigma$  den Constraint  $c$ , dann ist  $\sigma$  Lösung des gemischten  $E$ -Unifikationsproblems  $\langle E, s, t \rangle$ .

#### Beweis:

Die Struktur des Beweises ähnelt der des Beweises von Lemma 4.7.5. Wie dort muß der Fall  $n = \infty$  nicht weiter betrachtet werden, da jeder in  $\mathcal{T}^\infty$  enthaltene Term mit Constraint gemäß Definition schon in einer der Mengen  $\mathcal{T}^n$  ( $n \geq 0$ ) enthalten sein muß. Für diese wird die Behauptung durch Induktion über  $n$  bewiesen.

„ $n = 0$ “:

$\mathcal{T}^0$  enthält nur den Term mit Constraint  $s \ll \epsilon$ , und jede Substitution ist Lösung von  $\langle E, s, s \rangle$ .

„ $n \rightarrow n + 1$ “:

*Fall 1:* Sei  $\mathcal{T}^{n+1}$  aus  $\mathcal{T}^n$  durch eine Normalisierung entstanden, die keinen neuen Term hinzufügt (Löschung, Subsumtion), dann folgt die Korrektheit von  $\mathcal{T}^{n+1}$  trivialerweise aus der von  $\mathcal{T}^n$ .

*Fall 2:* Sei  $\mathcal{T}^{n+1}$  aus  $\mathcal{T}^n$  durch eine Äquivalenzumformung entstanden, und sei

$$(\forall \bar{x})(t\sigma \ll \langle \sigma, O \rangle) \in \mathcal{T}^{n+1}$$

ein beliebiger der neu hinzugekommenen, aus  $(\forall \bar{x})(t \ll c) \in \mathcal{T}^n$  entstandenen Terme. Erfüllt eine Substitution  $\tau$  den Constraint  $\langle \sigma, O \rangle$ , dann erfüllt sie auch den Constraint  $c$ , da  $\text{Sat}(\langle \sigma, O \rangle) \subset \text{Sat}(c)$ . Aus der Korrektheit von  $\mathcal{T}^n$  folgt

$$s\tau \xleftrightarrow{*}_{E\tau} t\tau ,$$

und da  $\sigma$  idempotent und  $\tau$  eine Spezialisierung von  $\sigma$  ist,

$$s\tau \xleftrightarrow{*}_{E\tau} t\sigma\tau .$$

Also ist  $\tau$  Unifikator von  $\langle E, s, t\sigma \rangle$ .

*Fall 3:* Sei  $\mathcal{T}^{n+1}$  aus  $\mathcal{T}^n$  durch Anwendung der Vereinfachungs- oder der Ableitungsregel entstanden. Sei  $\mathbf{t}' = (\forall \bar{y})(t' \ll c') \in \mathcal{T}^{n+1}$  der neue, aus  $\mathbf{t} = (\forall \bar{y})(t \ll c) \in \mathcal{T}^n$  entstandene Term (es gilt also  $\mathbf{t} \Rightarrow_{\mathcal{R}} \mathbf{t}'$  bzw.  $\mathbf{t} \Rightarrow_{\mathcal{R}^n} \mathbf{t}'$ ); schließlich sei  $\tau$  eine Substitution, die  $c'$  erfüllt. Da  $c$  in den Constraint  $c'$  eingeht, erfüllt  $\tau$  auch  $c$ , und da die Behauptung für  $\mathcal{T}^n$  schon bewiesen ist, folgt

$$s\tau \xleftrightarrow{*}_{E\tau} t\tau .$$

Weil die Systeme  $\mathcal{R}$  und  $\mathcal{R}^n$  korrekt sind bezüglich  $E$  folgt aus  $\mathbf{t} \Rightarrow_{\mathcal{R}} \mathbf{t}'$

$$t\tau \xleftrightarrow{*}_{E\tau} t'\tau .$$

Schließlich gilt also auch

$$s\tau \xleftrightarrow{*}_{E\tau} t'\tau .$$

*Fall 4:* Sei  $\mathcal{T}^{n+1} = \mathcal{T}^n$ . Dann überträgt sich die Behauptung trivialerweise von  $n$  auf  $n+1$ . ■

**Satz 4.7.7** (Korrektheit des Verfahrens)

Sei  $\langle E, s, t \rangle$  ein gemischtes  $E$ -Unifikationsproblem. Jede Substitution  $\sigma$ , die einen der Constraints in einer der gemäß Definition 4.6.1 gebildeten Mengen

$$\mathcal{C}_i^{j,k}(\langle E, s, t \rangle) \quad i, j, k \in \{0, 1, 2, \dots, \infty\} ,$$

eine der gemäß Definition 4.6.6 gebildeten Mengen

$$\mathcal{C}^i(\langle E, s, t \rangle) \quad i \in \{0, 1, 2, \dots, \infty\}$$

oder

$$\mathcal{C}(\langle E, s, t \rangle) = \bigcup_{i \geq 0} \mathcal{C}^i(\langle E, s, t \rangle)$$

erfüllt, ist ein Unifikator des Problems  $\langle E, s, t \rangle$ .

**Beweis:**

Zunächst ist festzuhalten, daß die Reduktionssysteme  $\mathcal{R}^i$  ( $i = 0, 1, 2, \dots, \infty$ ) gemäß Lemma 4.7.5 sämtlich korrekt sind, so daß Satz 4.7.6 anwendbar ist.

Seien  $i, j, k \in \{0, 1, 2, \dots, \infty\}$  beliebig und  $\tau$  eine Substitution, die einen Constraint  $c \in (\mathcal{C}_i^{j,k} \cup \mathcal{C}^i)$  erfüllt. Dann gibt es gemäß Definitionen 4.6.1 und 4.6.6 (die Bezeichnungen sind die gleichen wie dort) Terme mit Constraint

$$(\forall \bar{x})(r_1 \ll c_1) \in \mathcal{S}^{i,j} \quad \text{und} \quad (\forall \bar{y})(r_2 \ll c_2) \in \mathcal{T}^{i,k}$$

bzw.

$$(\forall \bar{x})(r_1 \ll c_1) \in \mathcal{S}^i \quad \text{und} \quad (\forall \bar{y})(r_2 \ll c_2) \in \mathcal{T}^i ,$$

so daß  $c = c_1 \sqcup c_2 \sqcup \langle \mu, \text{true} \rangle$ , wobei  $\mu$  ein idempotenter Unifikator von  $r_1$  und  $r_2$  ist. Es gilt  $s\tau \xleftrightarrow{*}_E r_1\tau$  (mit Satz 4.7.6),  $t\tau \xleftrightarrow{*}_E r_2\tau$  (ebenfalls mit Satz 4.7.6) und  $r_1\tau = r_2\tau$ . Daraus folgt

$$s\tau \xleftrightarrow{*}_E t\tau ,$$

und also mit Korollar 3.2.3 die Behauptung.

Die Korrektheit der Menge  $\mathcal{C}(\langle E, s, t \rangle)$  folgt aus der der Mengen  $\mathcal{C}^i(\langle E, s, t \rangle)$ , deren Vereinigung sie ist. ■

## 4.8 Vollständigkeit

Das Ziel in diesem Abschnitt ist, zu zeigen, daß die Menge  $\mathcal{C}_\infty(\langle E, s, t \rangle)$  von Constraints, die durch getrennte Vervollständigung und Normalisierung berechnet wird (Def. 4.6.1), und die Menge  $\mathcal{C}(\langle E, s, t \rangle)$ , die durch vermischte Vervollständigung und Normalisierung entsteht (Def. 4.6.1), beide (grund-)vollständig sind. Dabei ist die Menge  $\mathcal{C}$  (und in analoger Weise die Menge  $\mathcal{C}_\infty$ ) grund-vollständig, wenn es für jeden Grund-Unifikator  $\sigma$  von  $\langle E, s, t \rangle$  eine Grundsubstitution  $\sigma'$  gibt, die einen der Constraints in  $\mathcal{C}$  erfüllt, und für die  $\sigma' \leq_E \sigma$  gilt (aus der Korrektheit des Verfahrens (Satz 4.7.7) folgt, daß auch  $\sigma'$  ein Unifikator ist).

Es ist allerdings sehr schwierig, direkt und ohne Umwege zu beweisen, daß ein bestimmter Unifikator  $\sigma'$  durch eine  $\overset{*}{\Rightarrow}_{\mathcal{R}^\infty}$ -Ableitungsfolge gefunden werden kann (was eigentlich notwendig wäre), zumal zunächst auch unklar ist, wie sich derjenige *ableitbare* Unifikator  $\sigma'$  aus einem gegebenen Grund-Unifikator  $\sigma$  ergibt, der  $\sigma$  subsumiert.

Darum wird hilfsweise eine Ableitbarkeitsrelation  $\rightarrow_{\mathcal{R}}$  eingeführt (die  $\Rightarrow_{\mathcal{R}}$  als Teilrelation enthält) und zunächst bewiesen, daß mit ihr zu jedem Grund-Unifikator  $\sigma$  ein Grund-Unifikator  $\sigma'$  aufgefunden werden kann, der  $\sigma$  subsumiert. Es wird dann gezeigt, daß für diese  $\sigma' \leq_E \sigma$  aus einer  $\overset{*}{\Rightarrow}_{\mathcal{R}}$ -Ableitung eine  $\overset{*}{\Rightarrow}_{\mathcal{R}}$ -Ableitung konstruieren kann. Der Beweis besteht aus den folgenden (hier nur grob umrissenen) Schritten:

1. Es wird gezeigt, daß die Ableitbarkeitsrelation  $\overset{*}{\Rightarrow}_{(\mathcal{R}^0)^\pm}$  vollständig ist, d. h. in gewissem (in Lemma 4.8.5 präzisierten) Sinne für jede Grundsubstitution  $\sigma$  die Relation  $\overset{*}{\leftrightarrow}_{E\sigma}$  mit einschließt.
2. Dann wird eine wohlfundierte Ordnung auf  $\overset{*}{\Rightarrow}_{\mathcal{R}}$ -Beweisen definiert (Def. 4.8.7) und gezeigt, daß sich jede Transformation des Reduktionssystems  $\mathcal{R}$  in eine Transformation der  $\overset{*}{\Rightarrow}_{\mathcal{R}}$ -Beweise übertragen läßt, die diese bezüglich der wohlfundierten Ordnung verkleinert oder gleich läßt (Lemma 4.8.9).
3. Auf den Grund-Unifikatoren wird ebenfalls eine wohlfundierte Ordnung definiert (Def. 4.8.10) und gezeigt, daß  $\mathcal{R}^{\infty\pm}$ -Beweise für minimale Unifikatoren minimal sind, und darum zumindest für diese keinen Peak enthalten können, es also analoge  $\mathcal{R}^\infty$ -Beweise geben muß (Lemma 4.8.13).
4. Schließlich wird gezeigt, daß man für minimale  $\sigma'$  aus jedem  $\overset{*}{\Rightarrow}_{\mathcal{R}^\infty}$ -Beweis einen  $\overset{*}{\Rightarrow}_{\mathcal{R}^\infty}$ -Beweis konstruieren kann (Satz 4.8.15). Das genügt, denn die Ordnung auf den Substitutionen wird so definiert, daß es zu jedem Unifikator  $\sigma$  ein minimales  $\sigma'$  gibt mit  $\sigma' \leq_E \sigma$  (Lemma 4.8.11).
5. Es folgt nun schon, daß das Verfahren vollständig ist, falls man bei der Normalisierung von Termen darauf verzichtet, die Vereinfachungsregel anzuwenden. In einem letzten Schritt wird bewiesen, daß das Verfahren vollständig bleibt, wenn man die Vereinfachungsregel hinzunimmt (Satz 4.8.17).

Die erste zentrale Beweisidee ist der des Vollständigkeitsbeweises für den UKBA ähnlich (beispielsweise in [Dershowitz, 1989]), nämlich eine wohlfundierte Ordnung auf Beweisen zu definieren, und mit ihrer Hilfe zu zeigen, daß es eine terminierende



Folge von Transformationen gibt, die stets zu einem Beweis führt, der keine Peaks<sup>22</sup> enthält. Die zweite wesentliche Idee, auch auf den Substitutionen eine wohlfundierte Ordnung zu definieren, wird in ähnlicher Weise für den Vollständigkeitsbeweis in [Gallier *et al.*, 1992] verwendet.

**Definition 4.8.1** (Die Ableitbarkeitsrelation  $\rightarrow_{\mathcal{R}}$ ,  $\sigma$ -Beweis)

Sei  $\mathcal{R}$  ein Reduktionssystem mit Constraints. Dann ist die Relation  $\rightarrow_{\mathcal{R}}$  auf der Menge der Terme mit Constraint folgendermaßen definiert:

Sei

$$\mathbf{t} = (\forall \bar{x})(t \ll c_t)$$

ein Term mit Constraint. Genau dann, wenn es eine Reduktionsregel

$$\mathbf{r} = (\forall \bar{y})(l \rightsquigarrow r \ll c_r)$$

gibt, die aus einer Reduktionsregel  $\mathbf{r}' \in \mathcal{R}$  durch gebundene Variablenumbenennung hervorgeht, so daß

1.  $\{x_1, \dots, x_n\} \cap \text{Var}(r) = \emptyset$  und  $\{y_1, \dots, y_m\} \cap \text{Var}(t) = \emptyset$ ,
2. es eine Substitution  $\mu$  und eine Position  $p$  in  $t\mu$  gibt, so daß  $(t\mu)|_p = l\mu$ ,
3. die Kombination

$$c_{neu} = c_t \sqcup c_r \sqcup \langle \mu, \text{true} \rangle$$

konsistent ist,

gelte

$$\mathbf{t} \rightarrow_{\mathcal{R}} \mathbf{t}' ,$$

wobei

$$\mathbf{t}' = (\forall \bar{x})(\forall \bar{y})((t\mu)[p/(r\mu)] \ll c_{neu}) .$$

In diesem Falle heie  $\langle \mathbf{r}, p, \mu \rangle$  eine Rechtfertigung fur  $\mathbf{t} \rightarrow_{\mathcal{R}} \mathbf{t}'$ .

Eine endliche Folge  $\langle \mathbf{t}_0, \dots, \mathbf{t}_n \rangle$  ( $n \geq 0$ ) von Termen mit Constraint, so da

$$\mathbf{t}_0 \rightarrow_{\mathcal{R}} \mathbf{t}_1 \rightarrow_{\mathcal{R}} \mathbf{t}_2 \rightarrow_{\mathcal{R}} \dots \rightarrow_{\mathcal{R}} \mathbf{t}_n ,$$

heie ein Beweis fur

$$\mathbf{t}_0 \overset{*}{\rightarrow}_{\mathcal{R}} \mathbf{t}_n .$$

Enthalten alle Rechtfertigungen  $\langle \mathbf{r}_i, p_i, \mu \rangle$  ( $i = 1, \dots, n$ ) dieses Beweises dieselbe Substitution  $\mu$ , und gilt zudem  $t_0 = t_0\mu$  (wobei  $t_0$  der Term in  $\mathbf{t}_0$  ist), so heie der Beweis ein  $\mu$ -Beweis (bzw. ein  $\mathcal{R}$ - $\mu$ -Beweis).

Die Definition der Relation  $\rightarrow_{\mathcal{R}}$  ist der von  $\Rightarrow_{\mathcal{R}}$  sehr hnlich. Die Unterschiede sind:

1. Die zu einer  $\rightarrow$ -Regelanwendung ausgefuhrte Substitution  $\mu$  mu nicht ein allgemeinsten Unifikator sein.

<sup>22</sup> Ein Peak ist ein Teilbeweis der Form  $\mathbf{t}_i \rightarrow_{\mathcal{R}} \mathbf{t}_{i+1} \rightarrow_{\mathcal{R}} \mathbf{t}_{i+2}$ , bzw. bei Termen und Reduktionssystemen ohne Constraints der Form  $t_i \leftarrow t_{i+1} \rightarrow t_{i+2}$ .

2. Die Position  $p$ , an der die Regel angewendet wird, muß nicht eine Position in  $t$  sein, sondern es genügt, wenn  $p$  eine Position in  $t\mu$  ist.

Die Relation  $\overset{*}{\rightarrow}_{\mathcal{R}}$  statt  $\overset{*}{\Rightarrow}_{\mathcal{R}}$  für die Implementierung oder überhaupt für etwas anderes als Beweiszwecke zu verwenden ist nicht sinnvoll:

### Beispiel 4.8.2

Mit  $\mathcal{R} = \{a \rightsquigarrow b \ll \epsilon\}$  gilt

$$\begin{aligned} (f(x) \ll \epsilon) &\rightarrow_{\mathcal{R}} (f(b) \ll \langle \{x/a\}, \text{true} \rangle, \\ &(f(f(b)) \ll \langle \{x/f(a)\}, \text{true} \rangle, \\ &(f(f(f(b))) \ll \langle \{x/f(f(a))\}, \text{true} \rangle, \\ &\dots \end{aligned}$$

Wie dieses Beispiel zeigt, ist jede Regel auf jeden Term, der eine freie Variable enthält auf unendlich viele verschiedene Arten anwendbar.  $\square$

Da für die Frage, ob eine Substitution Unifikator eines gegebenen Problems  $\langle E, s, t \rangle$  ist, nur die Belegung der in  $\langle E, s, t \rangle$  frei auftretenden Variablen eine Rolle spielt, liegt folgende Definition nahe:

### Definition 4.8.3 (Wesentlich gleiche Substitutionen)

Sei  $\langle E, s, t \rangle$  ein gemischtes  $E$ -Unifikationsproblem. Substitutionen  $\sigma$  und  $\tau$  heißen wesentlich gleich (bezüglich  $\langle E, s, t \rangle$ ), wenn sie auf den in  $\langle E, s, t \rangle$  frei auftretenden Variablen übereinstimmen, d. h.

$$\sigma|_{\text{Free}(\langle E, s, t \rangle)} = \tau|_{\text{Free}(\langle E, s, t \rangle)} .$$

### Lemma 4.8.4

Ist  $\sigma$  ein Unifikator des gemischten  $E$ -Unifikationsproblems  $\langle E, s, t \rangle$ , und ist die Substitution  $\tau$  wesentlich gleich  $\sigma$  (bezüglich  $\langle E, s, t \rangle$ ), dann ist auch  $\tau$  ein Unifikator.

### Beweis:

Da  $\langle E, s, t \rangle \sigma = \langle E, s, t \rangle \tau$ , folgt  $s\tau \overset{*}{\leftrightarrow}_{E\tau} t\tau$  aus  $s\sigma \overset{*}{\leftrightarrow}_{E\sigma} t\sigma$ .  $\blacksquare$

Zunächst wird nun die Vollständigkeit der Ableitbarkeitsrelation  $\overset{*}{\rightarrow}_{(\mathcal{R}^0)\pm}$  bewiesen:

### Lemma 4.8.5

Sei  $\sigma$  ein Grund-Unifikator des gemischten  $E$ -Unifikationsproblems  $\langle E, s, t \rangle$ , und  $\mathcal{R}^0$  das initiale Reduktionssystem zu  $E$ .

Dann gibt es einen Grund-Unifikator  $\sigma'$  des Problems, der wesentlich gleich  $\sigma$  ist, so daß

$$(s\sigma' \ll \epsilon) \overset{*}{\rightarrow}_{(\mathcal{R}^0)\pm} (t\sigma' \ll \epsilon) ,$$

und es hierfür einen  $\sigma'$ -Beweis gibt.

### Beweis:

Da  $\sigma$  ein Unifikator ist, gilt  $s\sigma \overset{*}{\leftrightarrow}_{E\sigma} t\sigma$ , und dafür muß es einen Beweis

$$s\sigma = t_0 \leftrightarrow_{E\sigma} \dots \leftrightarrow_{E\sigma} t_n = t\sigma$$

geben mit Rechtfertigungen  $\langle (\forall \bar{x}_i)(l_i \approx r_i), p_i, \sigma_i \rangle$  ( $1 \leq i \leq n$ ). Dieser Beweis werde nun durch — gemäß der Definition von  $\rightarrow$  (Def. 4.8.1) zulässige — Umbenennung der Variablen in  $\bar{x}_i$  ( $1 \leq i \leq n$ ) so umgeformt, daß die neuen Variablen-Tupel  $\bar{x}'_i$  und die Menge  $\text{Free}\langle E, s, t \rangle$  paarweise disjunkt sind (die Tupel auch untereinander). Dann gibt es eine Grundsubstitution  $\sigma'$ , die wesentlich gleich  $\sigma$  ist (bezüglich  $\langle E, s, t \rangle$ ) und mit den durch die Umbenennung entstandenen Substitutionen  $\sigma'_i$  auf den Variablen in  $\bar{x}_i$  übereinstimmt. Eine solche Substitution  $\sigma'$  sei beliebig gewählt.

Dann ist

$$(s\sigma' \ll \epsilon) = (t_0 \ll c_0) \rightarrow_{(\mathcal{R}^0)^\pm} \cdots \rightarrow_{(\mathcal{R}^0)^\pm} (t_n \ll c_n) = (t\sigma' \ll c)$$

mit Rechtfertigungen

$$\begin{aligned} & \langle (\forall \bar{x}'_i)(l'_i \rightsquigarrow r'_i \ll \langle id, l'_i \succ r'_i \rangle), p_i, \sigma' \rangle \quad \text{oder} \\ & \langle (\forall \bar{x}'_i)(l'_i \rightsquigarrow r'_i \ll \langle id, r'_i \succ l'_i \rangle), p_i, \sigma' \rangle \quad (1 \leq i \leq n) \end{aligned}$$

ein Beweis für  $(s\sigma' \ll \epsilon) \xrightarrow{*(\mathcal{R}^0)^\pm} (t\sigma' \ll c)$ . Dies sind korrekte Rechtfertigungen, denn:

1. Da  $\mathcal{R}^0$  das initiale System zu  $E$  ist, entstehen die Reduktionsregeln in den Rechtfertigungen durch gebundene Umbenennung aus Regeln in  $(\mathcal{R}^0)^\pm$ .
2. Es gilt  $l'_i\sigma' = l_i\sigma$  und  $r'_i\sigma' = r_i\sigma$ .

■

### Beispiel 4.8.6

Sei

$$\langle E, s, t \rangle = \langle \{(\forall x)(f(x) \approx x)\}, g(f(a), f(b), c), g(a, b, y) \rangle .$$

Beispielsweise ist  $\sigma = \{y/c\}$  ein Unifikator des Problems. Ein Beweis für  $s\sigma \xrightarrow{*(E\sigma)} t\sigma$  ist

$$g(f(a), f(b), c) \leftrightarrow_{E\sigma} g(a, f(b), c) \leftrightarrow_{E\sigma} g(a, b, c)$$

mit Rechtfertigungen  $\langle (\forall x)(f(x) \approx x), \langle 1 \rangle, \{x/a\} \rangle$  und  $\langle (\forall x)(f(x) \approx x), \langle 2 \rangle, \{x/b\}, \rangle$ .

Eine gemäß Lemma 4.8.5 existierende, Substitution  $\sigma'$ , die wesentlich gleich  $\sigma$  ist, ist beispielsweise  $\sigma' = \{x/a, x'/b, y/c\}$ , und

$$\begin{aligned} (g(f(a), f(b), c) \ll \epsilon) & \rightarrow_{(\mathcal{R}^0)^\pm} (g(a, f(b), c) \ll \langle \{x/a\}, \text{true} \rangle) \\ & \rightarrow_{(\mathcal{R}^0)^\pm} (g(a, b, c) \ll \langle \{x/a, x'/b\}, \text{true} \rangle) \end{aligned}$$

ist ein Beweis für

$$(s\sigma' \ll \epsilon) \xrightarrow{*(\mathcal{R}^0)^\pm} (t\sigma' \ll c)$$

mit Rechtfertigungen

$$\langle (\forall x)(f(x) \approx x) \ll \epsilon, \langle 1 \rangle, \sigma' \rangle \quad \text{und} \quad \langle (\forall x')(f(x') \approx x') \ll \epsilon, \langle 2 \rangle, \sigma' \rangle .$$

□

Nun kann die wohlfundierte Ordnung  $\ggg_x$  auf  $\sigma$ -Beweisen definiert werden. Anschließend wird gezeigt, daß sich jeder Transformation eines Reduktionssystems  $\mathcal{R}$  in einer Transformation der  $\mathcal{R}$ - $\sigma$ -Beweise widerspiegelt, die zu bezüglich  $\ggg_x$  kleiner werdenden oder gleich bleibenden Beweisen führt.

**Definition 4.8.7** (Die Ordnung  $\ggg_x$  auf  $\sigma$ -Beweisen, wesentlich gleiche  $\sigma$ -Beweise)

Sei  $\ggg_x$  die von  $\succ_x$  induzierte<sup>23</sup> Multiset-Ordnung auf aus Tupeln von Termen bestehenden Multisets.<sup>24</sup> Das heißt, es gilt

$$\mathcal{X} \ggg_x \mathcal{Y}$$

genau dann, wenn

1.  $\mathcal{X} \neq \mathcal{Y}$ ,
2. es zu jedem Tupel  $t_y \in \mathcal{Y} \setminus \mathcal{X}$  ein Tupel  $t_x \in \mathcal{X} \setminus \mathcal{Y}$  gibt mit

$$t_x \succ_x t_y .$$

Die Ordnung  $\ggg_x$  werde dadurch auf die Menge aller  $\mathcal{R}$ - $\sigma$ -Beweise

$$(t_0 \ll c_0) \rightarrow_{\mathcal{R}} \cdots \rightarrow_{\mathcal{R}} (t_k \ll c_k) \quad (k \geq 0) ,$$

so daß  $\sigma$  eine Grundsubstitution ist,<sup>25</sup> übertragen, daß einem Beweis mit Rechtfer-tigungen

$$\langle (\forall \bar{x}_i)(l_i \rightsquigarrow r_i \ll c_i), p_i, \sigma \rangle \quad (1 \leq i \leq k)$$

der aus Tripeln von Termen bestehende Multiset

$$\{\langle t_{i-1}, l_i \sigma, t_i \rangle : 1 \leq i \leq k\}$$

zugeordnet wird.

Wird  $\mathcal{R}$ - $\sigma$ -Beweisen der gleiche Multiset zugeordnet, so heißen sie wesentlich gleich.

### Lemma 4.8.8

Die Ordnung  $\ggg_x$  (Def. 4.8.7) ist wohlfundiert.

#### Beweis:

Eine wie die Ordnung  $\ggg_x$  definierte Multiset-Ordnung auf Tupeln von Termen ist immer dann wohlfundiert, wenn die zugrunde gelegte Ordnung auf Termen wohlfundiert ist [Bachmair *et al.*, 1986]. Die Ordnung  $\succ_x$  aber ist per Definition (Def. 4.2.2) wohlfundiert. ■

<sup>23</sup>  $\ggg_x$  bezeichnet die durch die totale Reduktionsordnung  $\succ_x$ , die für die Interpretation des Symbols  $\succ$  verwendet wird, lexikographisch induzierte Ordnung auf Tupeln von Termen (Def. 3.6.7).

<sup>24</sup> Ein Multiset (Vielfachmenge) ist eine Menge, in der Elemente mehrfach auftreten können.

<sup>25</sup> Da es sich um einen  $\sigma$ -Beweis handelt, und  $\sigma$  eine Grundsubstitution ist, enthalten die  $t_i$  ( $1 \leq i \leq k$ ) keine Variablen (insbesondere auch keine gebundenen).

**Lemma 4.8.9**

Seien  $\mathcal{R}$  und  $\mathcal{R}'$  Reduktionssysteme mit  $\mathcal{R} \mid\text{-} \mathcal{R}'$ . Dann gibt es zu jedem  $\mathcal{R}$ - $\sigma$ -Beweis

$$\mathbf{t}_0 \rightarrow_{\mathcal{R}^\pm} \cdots \rightarrow_{\mathcal{R}^\pm} \mathbf{t}_k \quad (k \geq 0) ,$$

so daß  $\sigma$  eine Grundsubstitution ist, einen bezüglich der Ordnung  $\ggg_x$  kleineren oder wesentlich gleichen  $\mathcal{R}'$ - $\sigma$ -Beweis für

$$\mathbf{t}_0 \xrightarrow{*(\mathcal{R}')^\pm} \mathbf{t}_k .$$

**Beweis:**

Sei  $\langle (\forall \bar{x}_i)(l_i \rightsquigarrow r_i \ll c_i), p_i, \sigma \rangle$  die Rechtfertigung für  $\mathbf{t}_{i-1} \rightarrow_{\mathcal{R}^\pm} \mathbf{t}_i$  ( $1 \leq i \leq k$ ). In Abhängigkeit davon, durch welche Transformation  $\mathcal{R}'$  aus  $\mathcal{R}$  hervorgegangen ist, sind die folgenden Fälle zu unterscheiden:

*Fall 1:* Ist  $\mathcal{R}'$  aus  $\mathcal{R}$  durch Anwendung der Lösungsregel hervorgegangen, dann kann die gelöschte Reduktionsregel, da ihr Constraint unerfüllbar ist, nicht an dem  $\mathcal{R}$ - $\sigma$ -Beweis beteiligt sein, und dieser ist also auch ein  $\mathcal{R}'$ - $\sigma$ -Beweis.

*Fall 2:* Ist  $\mathcal{R}'$  aus  $\mathcal{R}$  durch Anwendung der Subsumptionsregel entstanden, wobei die Regel  $\mathbf{r} = (\forall \bar{x})(l \rightsquigarrow r \ll c) \in \mathcal{R} \cap \mathcal{R}'$  die Regel  $\mathbf{r}_i = (\forall \bar{x})(l_i \rightsquigarrow r_i \ll c_i) \in \mathcal{R}$  subsumiert, dann erhält man den gesuchten  $\mathcal{R}'$ - $\sigma$ -Beweis dadurch, daß man die Rechtfertigung  $\langle (\forall \bar{x})(l \rightsquigarrow r \ll c), p_i, \sigma \rangle$  statt  $\langle \mathbf{r}_i, p_i, \sigma_i \rangle$  verwendet, und den  $\mathcal{R}$ - $\sigma$ -Beweis ansonsten beibehält. Dies ist zulässig, da  $l_i \sigma = l \sigma$ ,  $r_i \sigma = r \sigma$  und  $\sigma \in \text{Sat}(c_i) \subset \text{Sat}(c)$ .

*Fall 3:* Ist  $\mathcal{R}'$  aus  $\mathcal{R}$  durch Anwendung der Äquivalenzumformungsregel entstanden, die auf eine der Reduktionsregeln  $\mathbf{r}_i = (\forall \bar{x})(l_i \rightsquigarrow r_i \ll c_i) \in \mathcal{R}^n$  angewendet wurde, dann gibt es eine Regel  $\mathbf{r} = (\forall \bar{x})(l \rightsquigarrow r \ll c) \in \mathcal{R}$ , so daß  $l_i \sigma = l \sigma$ ,  $r_i \sigma = r \sigma$  und  $\sigma \in \text{Sat}(c_i) \subset \text{Sat}(c)$ , und man kann wie in Fall 2 vorgehen.

*Fall 4:* Ist  $\mathcal{R}$  aus  $\mathcal{R}'$  durch Anwendung der Kritisches-Paar-Regel entstanden, gilt  $\mathcal{R} \subset \mathcal{R}'$ , und also ist der  $\mathcal{R}$ - $\sigma$ -Beweis auch ein  $\mathcal{R}'$ - $\sigma$ -Beweis.

*Fall 5:* Ist  $\mathcal{R}'$  aus  $\mathcal{R}$  durch Anwendung der Vereinfachungsregel entstanden, wobei auf eine Regel  $\mathbf{r}_1 = (\forall \bar{x})(l \rightsquigarrow r \ll c)$  eine Regel  $\mathbf{r}_2 = (\forall \bar{y})(l' \rightsquigarrow r' \ll c')$  angewendet wurde ( $\mathbf{r}_1, \mathbf{r}_2 \in \mathcal{R}$ ), gilt also  $\mathbf{r}_1 \rightleftharpoons_{\mathcal{R}} \mathbf{r}_{neu}$  mit der Rechtfertigung  $\langle \mathbf{r}_2, p, \mu \rangle$ , und ist  $\mathbf{r}_{neu} = (\forall \bar{z})(l_{neu} \rightsquigarrow r_{neu} \ll c_{neu})$ , dann kann man zunächst folgern — allerdings nur unter der zusätzlichen Annahme, daß die (in  $\mathcal{R}'$  nicht mehr auftretende) Regel  $\mathbf{r}_1$  überhaupt an dem  $\mathcal{R}$ - $\sigma$ -Beweis beteiligt ist:

- $r \sigma = (r \sigma)[l' \sigma]$ ,<sup>26</sup>
- $l_{neu} \sigma = l \sigma$ ,
- $r_{neu} \sigma = (r \sigma)[r' \sigma]$ .

Es folgt, daß man den gesuchten  $\mathcal{R}'$ - $\sigma$ -Beweis aus dem  $\mathcal{R}$ - $\sigma$ -Beweis dadurch erzeugen kann, daß man

<sup>26</sup> Die Schreibweise  $r \sigma[l' \sigma]$  ist eine Abkürzung für  $r \sigma[p'/l' \sigma]$ . Die Position  $p'$  ist, da die Vereinfachungsregel angewendet wurde, bestimmt durch  $p = \langle 2 \rangle \circ p'$ .

1. jeden Teilbeweis

$$\mathbf{t}_i = (t[l\sigma] \ll c_i) \Rightarrow_{\mathcal{R}} (t[(r\sigma)[l'\sigma]] \ll c_{i+1}) = \mathbf{t}_{i+1}$$

bei dem  $\mathbf{r}_1 \in \mathcal{R}^n$  die angewendete Regel ist, durch

$$\mathbf{t}_i = (t[l\sigma] \ll c_i) \Rightarrow_{\mathcal{R}} (t[(r\sigma)[r'\sigma]] \ll c_{i+1}) \Rightarrow_{\mathcal{R}^-} (t[(r\sigma)[l'\sigma]] \ll c_{i+1}) = \mathbf{t}_{i+1}$$

ersetzt;

2. jeden Teilbeweis

$$\mathbf{t}_i = (t[(r\sigma)[l'\sigma]] \ll c_i) \Rightarrow_{\mathcal{R}} (t[l\sigma] \ll c_{i+1}) = \mathbf{t}_{i+1}$$

bei dem  $(\mathbf{r}_1)^- \in (\mathcal{R}^n)^-$  die angewendete Regel ist, durch

$$\mathbf{t}_i = (t[(r\sigma)[l'\sigma]] \ll c_i) \Rightarrow_{\mathcal{R}^-} (t[(r\sigma)[r'\sigma]] \ll c_{i+1}) \Rightarrow_{\mathcal{R}} (t[l\sigma] \ll c_{i+1}) = \mathbf{t}_{i+1}$$

ersetzt;

3. alle übrigen Teilbeweise beibehält.

*Fall 6:* Der Fall, daß  $\mathcal{R}'$  aus  $\mathcal{R}$  durch Anwendung der Kombinationsregel entstanden ist, kann völlig analog zu Fall 5 behandelt werden, denn die Anwendung der Kombinationsregel auf eine Reduktionsregel  $\mathbf{r} \in \mathcal{R}$  hat genau die gleichen Auswirkungen auf  $\mathcal{R}^\pm$  wie die Anwendung der Vereinfachungsregel auf ihre Inverse  $\mathbf{r}^- \in (\mathcal{R}^n)^-$  — und umgekehrt.

Damit ist die Fallunterscheidung abgeschlossen. Wie man leicht nachvollzieht, ist in jedem Fall der konstruierte  $\mathcal{R}'$ - $\sigma$ -Beweis bezüglich der Ordnung  $\ggg_x$  kleiner als oder wesentlich gleich wie der  $\mathcal{R}$ - $\sigma$ -Beweis. In den Fällen 1–4 bleibt der dem Beweis zugeordnete Multiset unverändert. In den Fällen 5 und 6 werden für einen Teilbeweis zwei neue Teilbeweise eingesetzt. Die den neuen Teilbeweisen zugeordneten Termtripel sind beide bezüglich  $\ggg_x$  kleiner als das dem ersetzten Teilbeweis zugeordnete Tripel. ■

Wie schon in der Einführung angedeutet, ist die Relation  $\overset{*}{\Rightarrow}_{\mathcal{R}^\infty}$  nur für bestimmte Grund-Unifikatoren  $\sigma$  vollständig — nämlich für solche Unifikatoren, die bezüglich einer auf den Grundsubstitutionen definierten Ordnungsrelation minimal sind:

**Definition 4.8.10** (Minimaler Grund-Unifikator)

Sei  $\langle E, s, t \rangle$  ein gemischtes  $E$ -Unifikationsproblem. Die in  $\langle E, s, t \rangle$  frei auftretenden Variablen seien in einer beliebigen aber festen Folge  $\langle x_1, \dots, x_n \rangle$  angeordnet.

Dann sei die Ordnungsrelation  $\ggg_{\langle E, s, t \rangle}$  auf der Menge der Grundsubstitutionen definiert durch:  $\tau \ggg_{\langle E, s, t \rangle} \sigma$  genau dann, wenn

$$\langle \tau(x_1), \dots, \tau(x_n) \rangle \ggg_x \langle \sigma(x_1), \dots, \sigma(x_n) \rangle .$$

$\sigma$  sei ein Grund-Unifikator von  $\langle E, s, t \rangle$ . Dann enthalte die Menge  $\mathcal{G}(\langle E, s, t \rangle, \sigma)$  genau die Substitutionen  $\tau$ , für die gilt:

1.  $\tau$  ist Grund-Unifikator von  $\langle E, s, t \rangle$ .
2.  $\tau \leq_E \sigma$ .

Der Grund-Unifikator  $\sigma$  heie *minimal*, wenn er ein bezglich  $\succ_{\langle E, s, t \rangle}$  minimales Element von  $\mathcal{G}(\langle E, s, t \rangle, \sigma)$  ist.

### Lemma 4.8.11

Sei  $\sigma$  Grund-Unifikator eines gemischten  $E$ -Unifikationsproblems  $\langle E, s, t \rangle$ . Dann gibt es einen minimalen Grund-Unifikator  $\sigma'$  von  $\langle E, s, t \rangle$  (Def. 4.8.10) mit  $\sigma' \leq_E \sigma$ .

### Beweis:

Der Grund-Unifikator  $\sigma$  selbst liegt in der Menge  $\mathcal{G}(\langle E, s, t \rangle, \sigma)$ , die also nicht leer ist. Da auerdem die Relation  $\succ_{\langle E, s, t \rangle}$  wohlfundiert ist (weil sie lexikographisch durch die wohlfundierte Ordnung  $\succ_x$  induziert wird), besitzt  $\mathcal{G}(\langle E, s, t \rangle, \sigma)$  (mindestens) ein minimales Element. Dieses sei mit  $\sigma'$  bezeichnet.  $\sigma'$  ist per Definition wieder ein Grund-Unifikator von  $\langle E, s, t \rangle$ , und es gilt  $\sigma' \leq_E \sigma$ .

Wre  $\sigma'$  zwar in  $\mathcal{G}(\langle E, s, t \rangle, \sigma)$  nicht aber in  $\mathcal{G}(\langle E, s, t \rangle, \sigma')$  minimal, gbe es also ein  $\sigma'' \in \mathcal{G}(\langle E, s, t \rangle, \sigma')$  mit  $\sigma'' \leq_E \sigma'$ , dann wre  $\sigma''$  wegen der Transitivitt von  $\leq_E$  (Lemma 3.3.5) auch Element von  $\mathcal{G}(\langle E, s, t \rangle, \sigma)$ , was der Forderung widersprche, da  $\sigma'$  minimal sei in  $\mathcal{G}(\langle E, s, t \rangle, \sigma)$ . ■

### Beispiel 4.8.12

Sei  $\langle E, s, t \rangle = \langle \{c \approx b\}, f(x), f(c) \rangle$ , und es gelte  $c \succ_x b$ . Dann sind  $\{x/c\}$  und  $\{x/b\}$  Grund-Unifikatoren. Aber nur  $\{x/b\}$  ist minimal. □

Da fr die Minimalitt eines Grund-Unifikators nur die Belegung der in  $\langle E, s, t \rangle$  frei auftretenden Variablen eine Rolle spielt, ist ein Grund-Unifikator  $\tau$ , der wesentlich gleich einem minimalen Grund-Unifikator  $\sigma$  ist, ebenfalls minimal.

Nun kann gezeigt werden, da sich die Vollstndigkeit von  $\rightarrow_{(\mathcal{R}^0)^\pm}$  auf die von  $\rightarrow_{\mathcal{R}^\infty}$  bertrgt, falls  $\mathcal{R}^\infty$  eine faire Vervollstndigung von  $\mathcal{R}^0$  ist:

### Lemma 4.8.13

Sei  $\sigma$  ein minimaler Grund-Unifikator eines gemischten  $E$ -Unifikationsproblems  $\langle E, s, t \rangle$ . Das Reduktionssystem  $\mathcal{R}^\infty$  sei eine faire Vervollstndigung des initialen Systems  $\mathcal{R}^0$  zu  $E$ .

Dann gibt es einen Grund-Unifikator  $\sigma'$  von  $\langle E, s, t \rangle$ , der wesentlich gleich  $\sigma$  ist, einen Term  $u$  und Constraints  $c_{u1}$  und  $c_{u2}$ , so da

$$(s\sigma' \ll \epsilon) \xrightarrow{\mathcal{R}^\infty} (u \ll c_{u1}) \quad \text{und} \quad (t\sigma' \ll \epsilon) \xrightarrow{\mathcal{R}^\infty} (u \ll c_{u2})$$

gilt, und es fr beides  $\mathcal{R}^\infty$ - $\sigma'$ -Beweise gibt.<sup>27</sup>

### Beweis:

Der Beweis beruht darauf, mit wachsendem  $n$  bezglich  $\succ_x$  kleiner werdende oder wesentlich gleich bleibende  $(\mathcal{R}^n)^\pm$ - $\sigma$ -Beweise fr

$$(s\sigma \ll \epsilon) \xrightarrow{(\mathcal{R}^n)^\pm} (t \ll c)$$

<sup>27</sup> Woraus folgt, da  $\sigma'$  die Constraints  $c_{u1}$  und  $c_{u2}$  erfllt.

zu konstruieren, und zu zeigen, daß ein Beweis zumindest immer dann echt verkleinert werden kann, wenn er einen Peak enthält. Da die Ordnung  $\ggg_x$  wohlfundiert ist, muß sich auf diese Weise ein Beweis konstruieren lassen, der keinen Peak enthält. Aus diesem  $\sigma'-(\mathcal{R}^n)^\pm$ -Beweis ohne Peak kann man den gesuchten Term  $\mathbf{u}$  und die entsprechenden  $\mathcal{R}^\infty$ -Beweise konstruieren.

Gemäß Lemma 4.8.5 existiert eine Grundsubstitution  $\sigma_0$ , die wesentlich gleich  $\sigma$  ist (und darum wieder ein minimaler Grund-Unifikator ist), und ein  $(\mathcal{R}^0)^\pm$ - $\sigma_0$ -Beweis  $B_0$  für

$$(s\sigma_0 \ll \epsilon) \xrightarrow{*(\mathcal{R}^0)^\pm} (t\sigma_0 \ll c_0) .$$

Von diesem ausgehend seien für jedes  $n \geq 0$  eine Substitution  $\sigma_n$  und ein Beweis für

$$(s\sigma_n \ll \epsilon) \xrightarrow{*(\mathcal{R}^n)^\pm} (t\sigma_n \ll c_0)$$

in der folgenden Weise für  $n \geq 1$  konstruiert:

1. Zunächst sei  $B'_n$  ein  $\sigma_{n-1}-(\mathcal{R}^n)^\pm$ -Beweis für

$$(s\sigma_{n-1} \ll \epsilon) \xrightarrow{*(\mathcal{R}^n)^\pm} (t\sigma_{n-1} \ll c_{n-1}) .$$

Seine Existenz folgt mit Lemma 4.8.9 aus der des Beweises  $B_{n-1}$ , und  $B'_n$  ist kleiner oder wesentlich gleich  $B_{n-1}$ .

2. Der Beweis  $B_n$  gehe dadurch aus  $B'_n$  hervor, daß Teilbeweise, die ein Peak sind, d. h. von der Form

$$\mathbf{t}_i \rightarrow_{(\mathcal{R}^n)-} \mathbf{t}_{i+1} \rightarrow_{\mathcal{R}^n} \mathbf{t}_{i+2} ,$$

sukzessive durch kleinere (nicht gleich große) Teilbeweise ersetzt werden, falls und solange das möglich ist. Dabei ist es zulässig, zu einer anderen Grundsubstitution  $\sigma'$  überzugehen, die wesentlich gleich  $\sigma_{n-1}$  und damit auch wesentlich gleich  $\sigma$  ist ( $\sigma'$  ist also ein minimaler Grund-Unifikator). Der neue Teilbeweis muß also ein  $\sigma'-(\mathcal{R}^n)^\pm$ -Beweis für  $\mathbf{t}_i \xrightarrow{*\mathcal{R}^n} \mathbf{t}_{i+2}$  sein. Auch Peaks in den neuen Teilbeweisen werden, soweit das möglich ist, durch kleinere Teilbeweise ersetzt.<sup>28</sup> Sind verschiedene Ersetzungen möglich, die sich gegenseitig ausschließen, werde jeweils eine beliebige ausgewählt. Diese Beweistransformation muß terminieren, da die Ordnung  $\ggg_x$  wohlfundiert ist (Lemma 4.8.8). Die für den auf diese Weise entstandenen Beweis  $B_n$  verwendete Substitution werde mit  $\sigma_n$  bezeichnet. Der Beweis  $B_n$  ist kleiner oder gleich  $B'_n$  und damit kleiner oder gleich  $B_{n-1}$ , und  $\sigma_n$  ist wie  $\sigma_{n-1}$  minimal.

Da  $\ggg_x$  wohlfundiert ist, muß es ein  $n \geq 1$  geben, so daß alle Beweise  $B_m$  ( $m \geq n$ ) *nicht* kleiner als und also wesentlich gleich  $B_n$  sind. Alle zum Beweis  $B_n$  verwendeten Regeln bzw. ihre Inversen aus  $\mathcal{R}^n$  sind also  $\sigma_n$ -persistent (Def. 4.4.10), da die den Beweisen  $B_m$  ( $m \geq n$ ) zugeordneten Multisets von Termtupeln (Def. 4.8.7) gleich bleiben, und die Instanzen der verwendeten Regeln in diese Multisets mit eingehen.

<sup>28</sup> Die neuen Teilbeweise dürfen Peaks enthalten. Sie müssen nur bezüglich der Ordnung  $\ggg_x$  kleiner sein als der alte Teilbeweis (der ersetzte Peak).



Angenommen, ein solcher Beweis  $B_m$  enthält als Teilbeweis einen Peak

$$(t_0 \ll c_0) \rightarrow_{(\mathcal{R}^m)-} (t_1 \ll c_1) \rightarrow_{\mathcal{R}^m} (t_2 \ll c_2) ,^{29}$$

mit Rechtfertigungen  $\langle \mathbf{r}_1, p_1, \sigma_m \rangle$  und  $\langle \mathbf{r}_2, p_2, \sigma_m \rangle$ , wobei

$$\mathbf{r}_1 = (\forall \bar{x})(l_1 \rightsquigarrow r_1 \ll c_{r_1}) \quad \text{und} \quad \mathbf{r}_2 = (\forall \bar{y})(l_2 \rightsquigarrow r_2 \ll c_{r_2}) .$$

Es folgt (da es sich um einen  $\sigma_m$ -Beweis handelt, gilt  $t_j = t_j \sigma_m$  für  $j = 0, 1, 2$ )

$$\begin{aligned} t_{0|p_1} &= l_1 \sigma_m \\ t_1 &= t_0[p_1/r_1 \sigma_m] \\ t_{1|p_2} &= l_2 \sigma_m \\ t_2 &= t_1[p_2/r_2 \sigma_m] = (t_0[p_1/r_1 \sigma_m])[p_2/r_2 \sigma_m] . \end{aligned}$$

Dann sind die folgenden Fälle zu unterscheiden, die sämtlich zu einem Widerspruch führen (siehe auch Beispiel 4.8.14).

*Fall 1:* Sei weder  $p_1$  eine Position in  $t_{0|p_2}$  noch  $p_2$  eine Position in  $t_{0|p_1}$ . Dann gilt

$$t_2 = (t_0[p_2/r_2 \sigma_m])[p_1/r_1 \sigma_m] ,$$

und der Peak kann durch den kleineren Teilbeweis

$$(t_0 \ll c_0) \rightarrow_{\mathcal{R}^m} (t_0[p_2/r_2 \sigma_m] \ll c_0 \sqcup c_{r_2}) \rightarrow_{(\mathcal{R}^m)-} (t_2 \ll c_2)$$

(mit den gleichen Rechtfertigungen wie die des Peaks) ersetzt werden, was der Forderung, daß  $B_m$  nicht weiter in dieser Weise zu reduzieren sei, widerspricht.

*Fall 2:* Ohne Beschränkung der Allgemeinheit sei  $p_1$  eine Position in  $t_{1|p_2}$ .<sup>30</sup> In diesem Fall gibt es eine Position  $p'_2$  in  $l_2 \sigma_m$ , so daß  $p_1 = p_2 \circ p'_2$  und  $t_{0|p_1} = (l_2 \sigma_m)_{|p'_2}$ .

*Fall 2.1:*  $p'_2$  sei eine Position in  $l_2$  (nicht nur in  $l_2 \sigma_m$ ), und  $l_2|_{p'_2}$  sei keine Variable. Dann bilden  $\mathbf{r}_2$  und die Inverse von  $\mathbf{r}_1$  (die beide in  $\mathcal{R}^m$  liegen) ein kritisches Paar, und dieses kritische Paar muß, wie oben ausgeführt,  $\sigma_m$ -persistent sein. Aus der Fairneß der Vervollständigung (Def. 4.4.10) folgt, daß es ein  $m' \geq 0$  geben muß, so daß in  $\mathcal{R}^{m'}$  die entsprechende Regel enthalten ist, die das kritische Paar ersetzen kann. Es gibt also einen  $\mathcal{R}^{m'}\text{-}\sigma_m$ - (Teil-)Beweis, der kleiner ist als der Peak. Ist  $m' < m$ , muß es wegen Lemma 4.8.9 auch einen solchen  $\mathcal{R}^m\text{-}\sigma_m$ -Beweis geben. Das widerspricht wiederum der Forderung, daß  $B_m$  nicht weiter zu reduzieren sei. Ist  $m' \geq m$ , kann der Peak bei der Konstruktion von  $B_{m'}$  entfernt werden, und  $B'_{m'}$  wird — im Widerspruch zur Annahme — kleiner sein als  $B_m$ . Daß der gleiche Peak in  $B_{m'}$  noch auftritt, folgt wie die Persistenz des kritischen Paares daraus, daß die den Beweisen  $B_{m'}$  ( $m' \geq m$ ) zugeordneten Termtupel gleich bleiben.

*Fall 2.2:*  $p'_2$  sei eine Position innerhalb einer Variablen  $x$  in  $l_2$ . Es gebe also eine Position  $p''$ , so daß  $\sigma_m(x)_{|p''} = t_{1|p_1} = r_1 \sigma_m$ . Dann sei

$$\sigma'_m(x) = \sigma_m(x)[p''/l_1 \sigma_m] ,$$

<sup>29</sup> Die Indizes 0, 1, 2 sollen nicht andeuten, daß der Peak den Anfang des Beweises bildet; sie werden der Einfachheit halber statt  $i$ ,  $i+1$  und  $i+2$  verwendet.

<sup>30</sup> Der andere, spiegelbildliche Fall, daß  $p_2$  eine Position in  $t_{1|p_1}$  ist, kann völlig analog behandelt werden.

und sonst stimme  $\sigma'_m$  mit  $\sigma_m$  überein.

*Fall 2.2.1:* Sei  $x$  eine in  $\mathbf{r}_2$  freie Variable und  $p_2'' = \langle \rangle$ . Dann ist  $r_1 = x$  (wegen der Orientierung der Regel kann  $x$  nicht nur in  $l_1$  auftreten). Auch in diesem Fall bilden  $\mathbf{r}_2$  und die Inverse von  $\mathbf{r}_1$  ein kritisches Paar, was wie in Fall 2.1 zum Widerspruch führt.

*Fall 2.2.2:* Sei  $x$  eine in  $\mathbf{r}_2$  freie Variable und  $p_2'' \neq \langle \rangle$ . Dann kann  $x$  nicht in  $r_1$  auftreten, denn sonst wäre  $\sigma_m(x)$  ein Unterterm von  $r_1\sigma$  und damit ein echter Unterterm von sich selbst.  $x$  kann auch nicht in  $l_1$  auftreten, denn dann wäre der Term  $r_1\sigma$ , der größer ist als  $l\sigma$  (bezüglich  $\succ_x$ ), wegen der Monotonie von  $\succ_x$  größer als er selbst. Dann folgt aber aus

$$l_1\sigma_m \xleftrightarrow{*}_{E\sigma_m} r_1\sigma_m$$

(was wegen der Korrektheit des Reduktionssystems  $\mathcal{R}^m$  gelten muß) und der Konstruktion von  $\sigma'_m$

$$\sigma_m(x) \xleftrightarrow{*}_{E\sigma'_m} \sigma'_m(x) .$$

Es gilt also  $\sigma'_m \leq_E \sigma_m$ , was wegen  $l_1\sigma_m \succ_x r_1\sigma_m$  und also  $\sigma'_m \not\prec_{(E,s,t)} \sigma_m$  der Minimalität von  $\sigma_m$  widerspricht.

*Fall 2.2.3:* Sei  $x$  eine in  $\mathbf{r}_2$  gebunden auftretende Variable. Wegen der gebundenen Umbenennung der Variablen vor der Anwendung von Regeln (Def. 4.8.1) tritt  $x$  dann im Rest des Beweises  $B_m$  nicht auf. Dann ist auch  $\sigma'_m$  minimal und wesentlich gleich  $\sigma_m$ . Außerdem gilt  $(t_0\sigma'_m)_{|p_2} = l_2\sigma'_m$  und also

$$(t_0 \ll c_0) \rightarrow_{\mathcal{R}^m} (t_0[p_2/r_2\sigma'_m] \ll c'_1) .$$

$t'_1 = t_0[p_2/r_2\sigma'_m]$  unterscheidet sich nur dadurch von  $t_1$ , daß an Positionen, an denen in  $t_1$  der Term  $r_1\sigma_m = r_1\sigma'_m$  auftritt, in  $t'_1$  der Term  $l_1\sigma_m = l_1\sigma'_m$  auftritt. Es gilt also

$$(t_0 \ll c_0) \rightarrow_{\mathcal{R}^m} (t'_1 \ll c'_1) \xrightarrow{*}_{(\mathcal{R}^m)^-} (t_2 \ll c_2) ,$$

und dieser neue Beweis ist, wie man leicht nachvollzieht kleiner als der Peak. Dies widerspricht nun wieder der Forderung, daß der Beweis  $B_m$  nicht weiter reduzierbar sein soll.

In jedem Fall also führt die Annahme,  $B_m$  enthalte einen Peak, zu einem Widerspruch, und also enthält keiner der Beweise  $B_m$  ( $m \geq n$ ) einen Peak, insbesondere auch der Beweis  $B_n$  nicht.

Sei  $\sigma' = \sigma_n$ . Da alle zum Beweis  $B_n$  verwendeten Regeln, wie schon gesagt,  $\sigma'$ -persistent sind, und die Vervollständigung fair ist, liegen entsprechende Instanzen der Regeln auch in  $\mathcal{R}^\infty\langle\sigma'\rangle$ , und es gibt also einen  $(\mathcal{R}^\infty)^\pm$ - $\sigma'$ -Beweis ohne Peak, d. h. von der Form

$$(s\sigma' \ll \epsilon) \xrightarrow{*}_{\mathcal{R}^\infty} (u \ll c_{u1}) \xrightarrow{*}_{(\mathcal{R}^\infty)^-} (t \ll c_t) .$$

Zu zeigen bleibt, daß aus dem  $(\mathcal{R}^\infty)^-$ - $\sigma'$ -Beweis

$$(u \ll c_{u1}) = (t_0 \ll c_0) \rightarrow_{(\mathcal{R}^\infty)^-} \cdots \rightarrow_{(\mathcal{R}^\infty)^-} (t_k \ll c_k) = (t \ll c)$$

mit Rechtfertigungen  $R_i$  ( $1 \leq i \leq k$ ) ein  $(\mathcal{R}^\infty)$ - $\sigma'$ -Beweis für

$$(t \ll \epsilon) \xrightarrow{*}_{\mathcal{R}^\infty} (u \ll c_{u2})$$

konstruiert werden kann.

Es gibt Constraints  $c'_i$  ( $1 \leq i \leq k$ ), so daß  $c_i = c_0 \sqcup c'_1 \sqcup \dots \sqcup c'_i$ , und da  $\sigma'$  eine Grundsubstitution ist, ist

$$(t \ll \epsilon) = (t_k \ll c''_k) \rightarrow_{\mathcal{R}^\infty} \dots \rightarrow_{\mathcal{R}^\infty} (t_0 \ll c''_0) = (u \ll c_{u2})$$

der gesuchte  $(\mathcal{R}^\infty)$ - $\sigma'$ -Beweis — und zwar mit den gleich bleibenden Rechtfertigungen  $R_i$  ( $1 \leq i \leq k$ ) und den Constraints

$$c''_i = c'_{k-1} \sqcup c'_{k-2} \sqcup \dots \sqcup c'_i .$$

■

#### Beispiel 4.8.14

Das folgende sind Beispiele für die im Beweis von Lemma 4.8.13 unterschiedenen Formen, die Peaks haben können. ( $\succ_x$  sei die alphabetische LPO.)

*Zu Fall 1:* Der Peak

$$(g(a, d) \ll \epsilon) \rightarrow_{\mathcal{R}-} (g(c, d) \ll \epsilon) \rightarrow_{\mathcal{R}} (g(c, b) \ll \epsilon)$$

mit Rechtfertigungen

$$\langle (a \rightsquigarrow c \ll \epsilon), \langle 1 \rangle, id \rangle , \\ \langle (d \rightsquigarrow b \ll \epsilon), \langle 2 \rangle, id \rangle$$

kann durch

$$(g(a, d) \ll \epsilon) \rightarrow_{\mathcal{R}} (g(a, b) \ll \epsilon) \rightarrow_{\mathcal{R}-} (g(c, b) \ll \epsilon)$$

ersetzt werden.

*Zu Fall 2.1:* Der Peak

$$(a \ll \langle \{x/c\}, \text{true} \rangle) \rightarrow_{\mathcal{R}-} (f(c) \ll \langle \{x/c\}, \text{true} \rangle) \rightarrow_{\mathcal{R}} (c \ll \langle \{x/c\}, \text{true} \rangle)$$

mit Rechtfertigungen

$$\langle (a \rightsquigarrow f(c) \ll \epsilon), \langle \rangle, \{x/c\} \rangle , \langle (f(x) \rightsquigarrow x \ll \epsilon), \langle \rangle, \{x/c\} \rangle$$

kann durch

$$(a \ll \langle \{x/c\}, \text{true} \rangle) \rightarrow_{\mathcal{R}-} (c \ll \langle \{x/c\}, \text{true} \rangle)$$

ersetzt werden, nachdem die Kritische-Paar-Regel auf das kritische Paar

$$(f(x) \rightsquigarrow x \ll \epsilon) \quad \text{und} \quad (f(c) \rightsquigarrow a \ll \epsilon)$$

angewendet wurde.

*Zu Fall 2.2.1:* Der Peak

$$(c \ll \epsilon) \rightarrow_{\mathcal{R}-} (f(c) \ll \epsilon) \rightarrow_{\mathcal{R}} (f(a) \ll \epsilon)$$

mit Rechtfertigungen

$$\langle (x \rightsquigarrow f(x) \ll \epsilon), \langle \rangle, \{x/c\} \rangle , \\ \langle (c \rightsquigarrow a \ll \epsilon), \langle 1 \rangle, \{x/c\} \rangle$$

kann durch

$$(c \ll \epsilon) \rightarrow_{\mathcal{R}-} (f(a) \ll \epsilon)$$

ersetzt werden, nachdem die Kritisches-Paar-Regel auf

$$(f(x) \rightsquigarrow x \ll \epsilon) \quad \text{und} \quad (x \rightsquigarrow a \ll \epsilon)$$

angewendet wurde.

*Zu Fall 2.2.2: Der Peak*

$$(g(a) \ll \epsilon) \rightarrow_{\mathcal{R}-} (g(b) \ll \epsilon) \rightarrow_{\mathcal{R}} (f(b, b) \ll \epsilon)$$

mit Rechtfertigungen

$$\begin{aligned} &\langle (a \rightsquigarrow b \ll \epsilon), \langle 1 \rangle, \{x/b\} \rangle, \\ &\langle (g(x) \rightsquigarrow f(x, x) \ll \epsilon), \langle \rangle, \{x/b\} \rangle \end{aligned}$$

kann durch

$$(g(a) \ll \epsilon) \rightarrow_{\mathcal{R}} (f(a, a) \ll \epsilon) \xrightarrow{*}_{\mathcal{R}-} (f(b, b) \ll \epsilon)$$

mit Rechtfertigungen

$$\begin{aligned} &\langle (g(x) \rightsquigarrow f(x, x) \ll \epsilon), \langle \rangle, \{x/a\} \rangle, \\ &\langle (a \rightsquigarrow b \ll \epsilon), \langle 1 \rangle, \{x/a\} \rangle, \\ &\langle (a \rightsquigarrow b \ll \epsilon), \langle 2 \rangle, \{x/a\} \rangle \end{aligned}$$

ersetzt werden. Dabei ist  $\sigma = \{x/b\}$  nicht minimal, wohl aber  $\sigma' = \{x/a\}$ .

*Zu Fall 2.2.3: Das Beispiel für diesen Fall unterscheidet sich von vorhergehenden nur dadurch, daß die Variable  $x$  gebunden ist. Der gleiche Peak wie oben, jedoch mit Rechtfertigungen*

$$\begin{aligned} &\langle (a \rightsquigarrow b \ll \epsilon), \langle 1 \rangle, \{x/b\} \rangle, \\ &\langle (\forall x)(g(x) \rightsquigarrow f(x, x) \ll \epsilon), \langle \rangle, \{x/b\} \rangle \end{aligned}$$

kann wie oben ersetzt werden. □

Als nächstes wird die Vollständigkeit der Ableitbarkeitsrelation  $\xrightarrow{*}_{\mathcal{R}^\infty}$  auf die der Relation  $\xrightarrow{*}_{\mathcal{R}^\infty}$  übertragen.

#### Satz 4.8.15

*Sei  $\langle E, s, t \rangle$  ein gemischtes  $E$ -Unifikationsproblem und  $\sigma$  ein minimaler Grund-Unifikator von  $\langle E, s, t \rangle$ . Ferner sei  $\mathcal{R}^\infty$  eine faire Vervollständigung des zu  $E$  gehörenden initialen Systems. Dann gibt es Terme mit Constraint*

$$\mathbf{u}_1 = (\forall \bar{x})(r_1 \ll c_1) \quad \text{und} \quad \mathbf{u}_2 = (\forall \bar{y})(r_2 \ll c_2) ,$$

so daß:

1.  $(s \ll \epsilon) \xrightarrow{*}_{\mathcal{R}^\infty} \mathbf{u}_1$  und  $(t \ll \epsilon) \xrightarrow{*}_{\mathcal{R}^\infty} \mathbf{u}_2$ ;
2.  $r_1$  und  $r_2$  sind unifizierbar mit einem MGU  $\mu$ ;

3.  $\sigma$  erfüllt den Constraint  $c_1 \sqcup c_2 \sqcup \langle \mu, \text{true} \rangle$ .

**Beweis:**

Durch Induktion über  $n$  kann man folgendes technische Lemma beweisen: Unter den Voraussetzungen

1. Es gibt einen  $\mathcal{R}^\infty$ - $\sigma$ -Beweis

$$(t_0 \ll \epsilon) \rightarrow_{\mathcal{R}^\infty} \dots \rightarrow_{\mathcal{R}^\infty} (t_n \ll c_n)$$

mit Rechtfertigungen  $\langle \mathbf{r}_i, p_i, \sigma \rangle$  ( $1 \leq i \leq n$ ).

2.  $\mathbf{t}'_0 = (\forall \bar{x})(t'_0 \ll c'_0)$  ist ein Term mit Constraint, so daß  $\sigma$  den Constraint  $c_0$  erfüllt, und

3.  $t'_0 \sigma = t_0$  gilt.

4. Falls sich eine Substitution  $\tau$  von der Substitution  $\sigma$  nur in der Belegung einer in  $\mathbf{t}'_0$  gebunden auftretenden Variablen  $x$  unterscheidet, und  $\sigma(x) \succ_x \tau(x)$  gilt, dann erfüllt  $\tau$  den Constraint  $c'_0$ .

gibt es einen Term mit Constraint  $\mathbf{t} = (\forall y)(t \ll c)$  und eine Grundsubstitution  $\sigma'$ , so daß:

1.  $\mathbf{t}'_0 \xrightarrow{\ast}_{\mathcal{R}^\infty} \mathbf{t}$ ,

2.  $\sigma'$  stimmt mit  $\sigma$  wesentlich überein ( $\sigma'$  ist also ein minimaler Grund-Unifikator),

3.  $\sigma'$  erfüllt den Constraint  $c$ ,

4.  $t \sigma' = t_n$ .

„ $n = 0$ “:

Für  $n = 0$  folgt aus den Voraussetzungen, daß  $\mathbf{t} = \mathbf{t}'_0$  und  $\sigma' = \sigma$  die gewünschten Eigenschaften haben.

„ $n - 1 \rightarrow n$ “:

Die für den Beweisschritt  $(t_0 \ll \epsilon) \rightarrow_{\mathcal{R}^\infty} (t_1 \ll c_1)$  verwendete Reduktionsregel sei  $\mathbf{r}_1 = (\forall \bar{z})(l \rightsquigarrow r \ll c_r)$ . Folgende Fälle (siehe auch Beispiel 4.8.16) sind zu unterscheiden:

*Fall 1:* Sei  $p_1$  eine Position in  $t'_0$ , und  $t'_{0|p_1}$  keine Variable, oder aber, falls  $t'_{0|p_1} = x$  eine Variable ist, sei  $l = x$ . Es folgt

$$(t'_{0|p_1})\sigma = t_{0|p_1} = l\sigma .$$

$t'_{0|p_1}$  und  $l$  sind also unifizierbar mit einem idempotenten MGU  $\mu \leq \sigma$ . Unter diesen Voraussetzungen gilt

$$(\forall \bar{x})(t'_0 \ll c'_0) \Rightarrow_{\mathcal{R}^\infty} (\forall \bar{y})(t'_1 \ll c'_1)$$

mit der Rechtfertigung  $\langle \mathbf{r}_1, p_1, \mu \rangle$ , wobei

$$t'_1 \sigma = (t'_0[p_1/r])\mu\sigma = t_0[p_1/r\sigma] = t_1 \quad ,$$

und  $\sigma$  erfüllt als Spezialisierung von  $\mu$  den Constraint  $c'_1$ .

Sei  $x$  eine in  $\mathbf{t}'_1$  gebunden auftretende Variable, und die Substitution  $\tau$  unterscheide sich nur in der Belegung der Variablen  $x$  von  $\sigma$ .

Es ist nur dann möglich, daß  $x$  in  $\mathbf{t}'_1$  aber nicht schon in  $\mathbf{t}'_0$  vorkommt, wenn  $x$  in  $r$  aber nicht in  $l$  enthalten ist. Dann erfüllt  $\tau$  den Constraint  $c'_1$ , der die Vereinigung von  $c'_0$ ,  $c_r$  und  $\langle \mu, \text{true} \rangle$  ist, immer dann, wenn  $\tau$  den Constraint  $c_r$  der Reduktionsregel erfüllt (in  $\mu$  kommt  $x$  nicht vor). Wie man aber anhand der Transformationsregeln (Abschnitt 4.4) nachvollzieht, haben die Constraints  $c_r$  aller Reduktionsregeln in einer Vervollständigung die Eigenschaft: Erfüllt die Substitution  $\sigma$  den Constraint  $c_r$ , unterscheidet sich die Substitution  $\tau$  nur in der Belegung einer Variablen  $x$  von  $\sigma$ , die in der rechten Seite, nicht aber in der linken Seite der Reduktionsregel auftritt, und gilt  $\sigma(x) \succ_x \tau(x)$ , dann erfüllt auch  $\tau$  den Constraint  $c_r$ .

Ist dagegen  $x$  eine in  $\mathbf{t}'_1$  gebundene Variable, die auch schon in  $\mathbf{t}'_0$  auftritt, dann kommt sie in  $c_r$  nicht vor,  $\tau$  erfüllt also  $c_r$ . Außerdem folgt aus den vorausgesetzten Eigenschaften von  $\mathbf{t}'_0$ , daß  $\tau$  den Constraint  $c'_0$  und also auch  $c'_1$  erfüllt.

Damit sind alle Voraussetzungen für die Anwendung der Induktionsannahme erfüllt; es gibt also einen Term mit Constraint  $\mathbf{t}$  und eine Grundsubstitution  $\sigma'$  mit den gewünschten Eigenschaften und

$$(\forall \bar{x})(t'_0 \ll c'_0) \Rightarrow_{\mathcal{R}^\infty} (\forall \bar{y})(t'_1 \ll c'_1) \stackrel{*}{\Rightarrow}_{\mathcal{R}^\infty} \mathbf{t} \quad .$$

*Fall 2:*  $p_1$  sei in  $t'_0$  eine Position innerhalb einer Variablen  $x$ . Es gebe also Positionen  $p'$  und  $p''$ , so daß  $p_1 = p' \circ p''$  und  $t'_{0|p'} = x$ . Dann gilt  $\sigma(x)_{|p''} = l\sigma$ . Sei  $\sigma'$  diejenige Grundsubstitution, die mit  $\sigma$  auf allen Variablen außer  $x$  übereinstimmt, und für die

$$\sigma'(x) = \sigma(x)[p''/r\sigma]$$

gilt.

*Fall 2.1:* Sei  $x$  eine in  $t'_0$  freie Variable.  $x$  kann nur in  $l$  oder  $r$  enthalten sein, wenn  $l = x$ , also Fall 1 eintritt. Die Begründung dafür ist die gleiche wie in Fall 2.2.2 des Beweises von Lemma 4.8.13; und mit der gleichen Argumentation wie dort kann man folgern, daß  $\sigma'$  ein Grund-Unifikator ist, und  $\sigma \succ_{(E,s,t)} \sigma'$  und  $\sigma' \leq_E \sigma$  gilt, was der Minimalität von  $\sigma$  widerspricht. Dieser Fall ( $x$  eine freie Variable,  $l \neq x$ ) muß also nicht weiter betrachtet werden.

*Fall 2.2:* Sei  $x$  eine in  $t'_0$  gebundene Variable. In diesem Fall erfüllt die Substitution  $\sigma'$  die Bedingung, daß sie sich nur in der Belegung einer gebundenen Variable von  $\sigma$  unterscheidet, und daß

$$\sigma(x) = \sigma(x)[p''/l\sigma] \succ_x \sigma(x)[p''/r\sigma] = \sigma'(x) \quad .$$

Also erfüllt  $\sigma'$  den Constraint  $c_0$ . Außerdem gilt  $t'_0\sigma' = t_1$ , und der  $\mathcal{R}^\infty$ - $\sigma$ -Beweis

$$(t_1 \ll c_1) \rightarrow_{\mathcal{R}^\infty} \dots \rightarrow_{\mathcal{R}^\infty} (t_n \ll c_n)$$

wird, da die Variable  $x$  (sie ist in  $t_0$  gebunden und also nicht frei) in ihm nicht auftritt durch Ersetzung von  $\sigma$  durch  $\sigma'$  zu einem  $\mathcal{R}^\infty$ - $\sigma'$ -Beweis. Dieser hat nur die Länge  $n - 1$ , und die Induktionsannahme ist auf ihn anwendbar. Es existiert also der gesuchte Term mit Constraint  $\mathbf{t}$  mit

$$(\forall \bar{x})(t'_0 \ll c'_0) = (\forall \bar{x})(t'_1 \ll c'_1) \xrightarrow{*}_{\mathcal{R}^\infty} (\forall \bar{y})(t \ll c)$$

und eine Substitution  $\sigma''$ , die die gewünschten Eigenschaften hat (da  $\sigma''$  mit  $\sigma'$  wesentlich übereinstimmt, stimmt  $\sigma''$  auch mit  $\sigma$  wesentlich überein).

Damit ist der Induktionsbeweis abgeschlossen.

Gemäß Lemma 4.8.13 existieren ein Term  $r$ , Constraints  $c_1$  und  $c_2$  und ein mit  $\sigma$  wesentlich übereinstimmender (und also minimaler) Grund-Unifikator  $\sigma'$  von  $\langle E, s, t \rangle$ , so daß

$$(s\sigma' \ll \epsilon) \xrightarrow{*}_{\mathcal{R}^\infty} (r \ll c_1) \quad \text{und} \quad (t\sigma' \ll \epsilon) \xrightarrow{*}_{\mathcal{R}^\infty} (r \ll c_2)$$

gilt, und es für beides  $\mathcal{R}^\infty$ - $\sigma'$ -Beweise gibt. Auf diese Beweise und auf die Terme mit Constraint  $(s \ll \epsilon)$  und  $(t \ll \epsilon)$  ist das durch Induktion bewiesene Lemma anwendbar. Es gibt also Substitutionen  $\sigma_1$  und  $\sigma_2$  und Terme mit Constraint<sup>31</sup>

$$\mathbf{u}_1 = (\forall \bar{x})(r'_1 \ll c'_1) \quad \text{und} \quad \mathbf{u}_2 = (\forall \bar{y})(r'_2 \ll c'_2) ,$$

so daß

$$(s \ll \epsilon) \xrightarrow{*}_{\mathcal{R}^\infty} \mathbf{u}_1 \quad \text{und} \quad (t \ll \epsilon) \xrightarrow{*}_{\mathcal{R}^\infty} \mathbf{u}_2 , \\ r'_1 \sigma_1 = r = r'_2 \sigma_2 .$$

Gemäß Konstruktion unterscheiden sich  $\sigma_1$  und  $\sigma_2$  nur in jeweils unterschiedlichen Variablen von  $\sigma'$ . Daher erfüllt die Substitution

$$\sigma''(x) = \begin{cases} \sigma_1(x) & \text{falls } \sigma_1(x) \neq \sigma'(x) \\ \sigma_2(x) & \text{falls } \sigma_2(x) \neq \sigma'(x) \\ \sigma'(x) & \text{sonst} \end{cases}$$

sowohl  $c_1$  als auch  $c_2$  ( $\sigma'$  erfüllt  $c_1$  und  $c_2$ ), und es gilt

$$r'_1 \sigma'' = r = r'_2 \sigma'' .$$

Daher sind  $r'_1$  und  $r'_2$  unifizierbar mit einem (idempotenten) MGU  $\mu$ , und  $\sigma''$  erfüllt den Constraint  $c'_1 \sqcup c'_2 \sqcup \langle \mu, \text{true} \rangle$ . ■

### Beispiel 4.8.16

Das folgende sind Beispiele für die im Beweis von Satz 4.8.15 unterschiedenen Transformationen von  $\rightarrow_{\mathcal{R}^\infty}$ -Beweisschritten in  $\Rightarrow_{\mathcal{R}^\infty}$ -Teilbeweise.

*Zu Fall 1:* Die Ersetzung des  $\rightarrow$ -Beweisschrittes

$$(f(g(a)) \ll \epsilon) \rightarrow (a \ll \langle \{x/g(a), y/a\}, \text{true} \rangle)$$

mit der Rechtfertigung  $\langle \mathbf{r}_1, \langle \rangle, \{x/g(a), y/a\} \rangle$  sei gesucht, wobei

$$\mathbf{r}_1 = (f(g(y)) \rightsquigarrow y \ll \epsilon) .$$

<sup>31</sup> Zur Verdeutlichung: der im Lemma mit  $\sigma$  bezeichneten Substitution entspricht  $\sigma'$ ;  $\sigma'$  entsprechen  $\sigma_1$  und  $\sigma_2$ ;  $t'_0$  entsprechen  $(s \ll \epsilon)$  und  $(t \ll \epsilon)$ ; und  $\mathbf{t}$  entsprechen  $\mathbf{u}_1$  und  $\mathbf{u}_2$ .

Ist  $(f(x) \ll \epsilon)$  der im Beweis von Satz 4.8.15 mit  $\mathbf{t}'_0$  bezeichnete,  $\mathbf{t}_0 = (f(g(a)) \ll \epsilon)$  subsumierende Term, dann liegt die Position  $\langle \rangle$  in  $f(x)$  nicht innerhalb einer Variablen, und Fall 1 tritt ein. Der gesuchte  $\Rightarrow$ -Teilbeweis ist nun

$$(f(x) \ll \epsilon) \Rightarrow (y \ll \langle \{x/g(y)\}, \text{true} \rangle)$$

mit der Rechtfertigung  $\langle \mathbf{r}_1, \langle \rangle, \{x/g(y)\} \rangle$ .

Zu Fall 2.1: Ist  $\langle \mathbf{r}_2, \langle 1 \rangle, \{x/g(a), y/a\} \rangle$  mit

$$\mathbf{r}_2 = (g(y) \rightsquigarrow y \ll \epsilon)$$

die Rechtfertigung für den  $\rightarrow$ -Beweisschritt

$$\mathbf{t}_0 = (f(g(a)) \ll \epsilon) \rightarrow (f(a) \ll \langle \{x/g(a), y/a\}, \text{true} \rangle) = \mathbf{t}_1 \quad ,$$

und wie oben  $\mathbf{t}'_0 = (f(x) \ll \epsilon)$ , dann liegt die Position  $\langle 1 \rangle$  in  $f(x)$  innerhalb der Variablen  $x$ . Nun ist tatsächlich die Substitution  $\{x/g(a), y/a\}$  nicht minimal, denn wann immer die Vervollständigung der Menge  $E$  von Gleichungen die Regel  $\mathbf{r}_2$  enthält, ist die Substitution  $\{x/a, y/a\}$  kleiner.

Zu Fall 2.2: Sei derselbe  $\rightarrow$ -Beweisschritt mit derselben Rechtfertigung wie im Beispiel zu Fall 2.1 gegeben, nun sei aber  $\mathbf{t}'_0 = (\forall z)(f(z) \ll \epsilon)$ . Dann liegt die Position  $\langle 1 \rangle$  in  $f(x)$  innerhalb der *gebundenen* Variablen  $z$ .  $(\forall z)(f(z) \ll \epsilon)$  subsumiert — anders als der nicht quantifizierte Term im vorangegangenen Beispiel — nicht nur  $\mathbf{t}_0$ , sondern auch  $\mathbf{t}_1$ , und der Beweisschritt kann also ausgelassen werden. □

Mit Satz 4.8.15 ist die Vollständigkeit des Verfahrens „fast“ bewiesen. Es bleibt zu zeigen, daß die nach Satz 4.8.15 existierenden und ableitbaren Terme mit Constraint  $\mathbf{u}_1$  und  $\mathbf{u}_2$  (oder Terme mit den gleichen Eigenschaften) in *allen* fairen Mengen  $\mathcal{S}^\infty$  und  $\mathcal{T}^\infty$  von Normalformen von

$$\mathcal{S}^0 = \{(s \ll \epsilon)\} \quad \text{und} \quad \mathcal{T}^0 = \{(t \ll \epsilon)\}$$

zu finden sind bzw. für ein hinreichend großes  $n$  in Mengen  $\mathcal{S}^n$  und  $\mathcal{T}^n$  jeder fairen Vervollständigungs- und Normalisierungsfolge. Es ist also zu zeigen, daß das Verfahren beweiskonfluent ist.

Das würde ohne weiteres aus Satz 4.8.15 folgen, könnten nicht durch die Anwendung der Vereinfachungsregel schon abgeleitete Terme wieder gelöscht werden. Mit der Vereinfachungsregel aber ist zunächst unklar, ob nicht auch für die Vollständigkeit unverzichtbare Terme wieder entfernt werden könnten. Dieser Fall kann aber tatsächlich, wie der folgende Satz zeigt, nicht eintreten:

**Satz 4.8.17** (Vollständigkeit des Verfahrens)

Für jedes gemischte  $E$ -Unifikationsproblem  $\langle E, s, t \rangle$  sind die Mengen von Unifikatoren

$$\text{Sat}(\mathcal{C}(\langle E, s, t \rangle)) \quad \text{und} \quad \text{Sat}(\mathcal{C}_\infty(\langle E, s, t \rangle))$$

(Def. 4.6.1 und 4.6.6) bezüglich der Subsumtionsrelation  $\leq_E$  grund-vollständig.



**Beweis:**

Sei  $\sigma$  ein beliebiger minimaler Grund-Unifikator von  $\langle E, s, t \rangle$ . Dann sei die Ordnung  $\succ_{\sigma}$  auf Termen mit Constraint definiert durch:

$$(\forall \bar{x})(t \ll c) \succ_{\sigma} (\forall \bar{y})(t' \ll c')$$

genau dann, wenn

- $\sigma$  die Constraints  $c$  und  $c'$  erfüllt, und
- $t\sigma \succ_x t'\sigma$ .

Da  $\succ_x$  als Reduktionsordnung wohlfundiert ist, ist auch  $\succ_{\sigma}$  wohlfundiert.

Sei  $\mathcal{T}^0 = \{(t \ll \epsilon)\}$ , und  $\mathcal{T}^0, \mathcal{T}^1, \mathcal{T}^2, \dots$  sei entweder eine faire Normalisierungsfolge (Def. 4.5.4) oder Teil einer fairen Vervollständigungs- und Normalisierungsfolge (Def. 4.6.5). Dann kann man durch Induktion über  $n$  zeigen, daß es für alle  $n \geq 0$  einen Term mit Constraint  $\mathbf{t}_n = (\forall \bar{x}_n)(t_n \ll c_n)$  in  $\mathcal{T}^n$  gibt mit

1.  $\sigma$  erfüllt  $c_n$ ,
2.  $t_0\sigma \xrightarrow{*}_{E\sigma} t_n\sigma$ ,
3.  $\mathbf{t}_{n-1} \succ_{\sigma} \mathbf{t}_n$  oder  $t_{n-1}\sigma = t_n\sigma$ .

„ $n = 0$ “:

$\mathbf{t}_0 = (t \ll \epsilon) \in \mathcal{T}^0$  erfüllt die Bedingungen.

„ $n \rightarrow n + 1$ “:

Ist der Term  $\mathbf{t}_n \in \mathcal{T}^n$  bei der Konstruktion von  $\mathcal{T}^{n+1}$  aus  $\mathcal{T}^n$  erhalten geblieben, kann man  $\mathbf{t}_{n+1} = \mathbf{t}_n \in \mathcal{T}^{n+1}$  verwenden. Das ist insbesondere der Fall, wenn  $\mathcal{T}^{n+1} = \mathcal{T}^n$ .

Sei also der Term  $\mathbf{t}_n$  nicht erhalten geblieben, es gelte also  $\mathbf{t}_n \notin \mathcal{T}^{n+1}$ . Dann sind in Abhängigkeit davon, durch Anwendung welcher Normalisierungsregel (Def. 4.5.2)  $\mathcal{T}^{n+1}$  aus  $\mathcal{T}^n$  hervorgegangen ist, folgende Fälle zu unterscheiden:

*Fall 1:* Ist  $\mathcal{T}^{n+1}$  durch Anwendung der Löschregel aus  $\mathcal{T}^n$  entstanden, dann kann  $\mathbf{t}_n \in \mathcal{T}^n$  nicht der gelöschte Term sein, weil  $\sigma$  den Constraint  $c_n$  erfüllt. Das steht im Widerspruch zu  $\mathbf{t}_n \notin \mathcal{T}^{n+1}$ .

*Fall 2:* Ist  $\mathcal{T}^{n+1}$  durch Anwendung der Äquivalenzumformungsregel auf  $\mathbf{t}_n$  aus  $\mathcal{T}^n$  entstanden, dann gibt es einen Term

$$\mathbf{t}_{n+1} = (\forall \bar{x}'_n)(t'_n \ll c'_n) \in \mathcal{T}^{n+1}, \quad \mathbf{t}_{n+1} \notin \mathcal{T}^n,$$

so daß  $\sigma$  den Constraint  $c'_n$  erfüllt, und  $t_n\sigma = t'_n\sigma$  gilt.

*Fall 3:* Ist  $\mathcal{T}^{n+1}$  durch Anwendung der Subsumtionregel auf  $\mathbf{t}_n$  aus  $\mathcal{T}^n$  entstanden, dann gibt es einen Term

$$\mathbf{t}_{n+1} = (\forall \bar{x}'_n)(t'_n \ll c'_n) \in \mathcal{T}^{n+1}, \quad \mathbf{t}_{n+1} \in \mathcal{T}^n,$$

der  $\mathbf{t}_n$  subsumiert. Also erfüllt  $\sigma$  den Constraint  $c'_n$ , und es gilt  $t_n\sigma = t'_n\sigma$ .

*Fall 4:* Ist  $\mathcal{T}^{n+1}$  durch Anwendung der Vereinfachungsregel auf  $\mathbf{t}_n$  aus  $\mathcal{T}^n$  entstanden, dann gibt es einen Term

$$\mathbf{t}_{n+1} = (\forall \bar{x}_{n+1})(t_{n+1} \ll c_{n+1}) \in \mathcal{T}^{n+1}$$

mit  $\mathbf{t}_n \Rightarrow_{\mathcal{R}^\infty} \mathbf{t}_{n+1}$  bzw.  $\mathbf{t}_n \Rightarrow_{\mathcal{R}^n} \mathbf{t}_{n+1}$ . Hierfür sei  $\langle (\forall \bar{z})(l \rightsquigarrow r \ll c_r), p, \mu \rangle$  eine Rechtfertigung. Es gilt  $t_{n+1} = (t_n[p/r])\mu$ ; außerdem subsumiert  $c_{n+1}$  den Constraint  $c_n$  (das folgt aus der Definition der Ableitbarkeitsrelation  $\Rightarrow$ , Def. 4.3.6). Also erfüllt  $\sigma$  den Constraint  $c_n$ , und es gilt  $\mu \leq \sigma$  und  $l\sigma \succ_x r\sigma$ . Da zudem  $\mu$  idempotent ist, folgt

$$t_n\sigma = (t_n[p/l])\sigma = (t_n\sigma)[p/l\sigma] \succ_x (t_n\sigma)[p/r\sigma] = (t_n[p/r])\sigma = t_{n+1}\sigma$$

und also  $\mathbf{t}_n \succ_\sigma \mathbf{t}_{n+1}$ .

*Fall 5:* Die Ableitungsregel kann nicht zur Konstruktion von  $\mathcal{T}^{n+1}$  aus  $\mathcal{T}^n$  verwendet worden sein, da bei ihrer Anwendung niemals ein Term entfernt wird, was im Widerspruch zu  $\mathbf{t}_n \notin \mathcal{T}^{n+1}$  steht.

Damit ist der Induktionsbeweis abgeschlossen. Da die Ordnung  $\succ_\sigma$  wohlfundiert ist, muß es ein  $n$  geben, so daß  $t_m\sigma = t_n\sigma$  und  $\sigma$  den Constraint  $c_n$  erfüllt (für alle  $m \geq n$ ). Also ist  $\mathbf{t}_n$   $\sigma$ -persistent, und weil die (Vervollständigungs- und) Normalisierungsfolge fair ist (Def. 4.5.4 bzw. 4.6.5), gibt es auch einen Term

$$\mathbf{t}_\infty = (\forall \bar{x}_\infty)(t_\infty \ll c_{t_\infty}) \in \mathcal{T}^\infty ,$$

der die drei oben genannten Eigenschaften hat, es gilt also

1.  $\sigma$  erfüllt  $c_\infty$ ,
2.  $t_0\sigma \stackrel{*}{\leftrightarrow}_{E\sigma} t_\infty\sigma$ .

Falls es mehrere Terme mit Constraint  $\mathbf{t}_\infty \in \mathcal{T}^\infty$  gibt, die diese Eigenschaften haben, sei ein bezüglich der Ordnung  $\succ_\sigma$  minimaler ausgewählt.

Ein Term  $\mathbf{s}_\infty = (\forall \bar{y}_\infty)(s_\infty \ll c_{s_\infty})$  in einer fairen Menge von Normalformen  $\mathcal{S}^\infty$  von  $\mathcal{S}^0 = \{(s \ll \epsilon)\}$  sei in gleicher Weise bestimmt. Da  $\sigma$  ein Unifikator von  $\langle E, s, t \rangle$  ist, gilt

$$t_\infty\sigma \stackrel{*}{\leftrightarrow}_{E\sigma} t_0\sigma = t\sigma \stackrel{*}{\leftrightarrow}_{E\sigma} s\sigma = s_0\sigma \stackrel{*}{\leftrightarrow}_{E\sigma} s_\infty\sigma .$$

Also ist  $\sigma$  Lösung des  $E$ -Unifikationsproblems  $\langle E, s_\infty, t_\infty \rangle$ . Aus Satz 4.8.15 folgt die Existenz von Termen mit Constraint

$$\mathbf{u}_1 = (\forall \bar{x})(r_1 \ll c_{u,1}) \quad \text{und} \quad \mathbf{u}_2 = (\forall \bar{y})(r_2 \ll c_{u,2}) ,$$

so daß:

1.  $(s_\infty \ll \epsilon) \stackrel{*}{\Rightarrow}_{\mathcal{R}^\infty} \mathbf{u}_1$  und  $(t_\infty \ll \epsilon) \stackrel{*}{\Rightarrow}_{\mathcal{R}^\infty} \mathbf{u}_2$ ;
2.  $r_1$  und  $r_2$  sind unifizierbar mit einem MGU  $\mu$ ;
3.  $\sigma$  erfüllt den Constraint  $c_1 \sqcup c_2 \sqcup \langle \mu, \text{true} \rangle$ .

Da  $\sigma$  die Constraints  $c_{s,\infty}$ ,  $c_{t,\infty}$ ,  $c_{u,1}$  und  $c_{u,2}$  erfüllt, gilt auch

$$\mathbf{s}_\infty \xrightarrow{*}_{\mathcal{R}^\infty} \mathbf{u}_1 \quad \text{und} \quad \mathbf{t}_\infty \xrightarrow{*}_{\mathcal{R}^\infty} \mathbf{u}_2 .$$

Gilt  $\mathbf{s}_\infty \xrightarrow{\dagger}_{\mathcal{R}^\infty} \mathbf{u}_1$ , gibt es wegen der Fairneß der Normalisierungsfolge auch einen Term  $\mathbf{u}'_1 \in \mathcal{T}^\infty$  (die mögliche Regelanwendung muß ausgeführt worden sein, es gilt  $\{\mathbf{u}_1\}\langle\sigma\rangle = \{\mathbf{u}'_1\}\langle\sigma\rangle$ ), so daß  $\mathbf{s}_\infty \xrightarrow{\dagger}_{\mathcal{R}^\infty} \mathbf{u}'_1$ , also  $\mathbf{s}_\infty \succ_{\sigma} \mathbf{u}'_1$  und  $\sigma$  den Constraint des Terms  $\mathbf{u}'_1$  erfüllt. Das aber widerspricht der geforderten Minimalität von  $\mathbf{s}_\infty$ . Also gilt  $\mathbf{s}_\infty = \mathbf{u}_1$ , und in gleicher Weise folgt  $\mathbf{t}_\infty = \mathbf{u}_2$ . Dann liegt aber gemäß Definition 4.6.1 der Constraint  $c_1 \sqcup c_2 \sqcup \langle\mu, \text{true}\rangle$ , der von  $\sigma$  erfüllt wird in der Menge  $\mathcal{C}_\infty(\langle E, s, t \rangle)$  und also  $\sigma$  in  $\text{Sat}(\mathcal{C}_\infty(\langle E, s, t \rangle))$ .

Aus  $\mathbf{s}_\infty \in \mathcal{S}^\infty$  und  $\mathbf{t}_\infty \in \mathcal{T}^\infty$  folgt, daß es ein (hinreichend großes)  $n \geq 0$  geben muß, so daß  $\mathbf{s}_\infty \in \mathcal{S}^n$  und  $\mathbf{t}_\infty \in \mathcal{T}^n$ . Also liegt der Constraint  $c_1 \sqcup c_2 \sqcup \langle\mu, \text{true}\rangle$  auch in  $\mathcal{C}^n$  und also in  $\mathcal{C}$ .

Da  $\sigma$  ein beliebig gewählter minimaler Grund-Unifikator ist, und es zu jedem Grund-Unifikator  $\tau$  einen minimalen Grund-Unifikator  $\sigma$  gibt mit  $\sigma \leq_E \tau$  (Lemma 4.8.11), ist der Satz damit bewiesen. ■

## 4.9 Schwache Konfluenz der Vervollständigung

Die Ableitbarkeitsrelation  $\xrightarrow{*}_{\mathcal{R}^\infty}$  ist nicht konfluent. Wie das folgende Beispiel zeigt, ist es auch nicht sinnvoll, das zu verlangen.

### Beispiel 4.9.1

$\mathcal{R}^\infty$  enthalte die Reduktionsregeln

$$f(a) \rightsquigarrow a \ll \epsilon \quad \text{und} \quad f(b) \rightsquigarrow b \ll \epsilon .$$

Dann gilt für  $\mathbf{s} = (f(x) \ll \epsilon)$

$$\mathbf{s} \Rightarrow_{\mathcal{R}^\infty} (a \ll \langle\{x/a\}, \text{true}\rangle) = \mathbf{t}_1 \quad \text{und} \quad \mathbf{s} \Rightarrow_{\mathcal{R}^\infty} (b \ll \langle\{x/b\}, \text{true}\rangle) = \mathbf{t}_2 .$$

Wäre  $\Rightarrow_{\mathcal{R}^\infty}$  konfluent, müßte es einen Term geben, der sowohl aus  $\mathbf{t}_1$  als auch aus  $\mathbf{t}_2$  ableitbar ist. Das würde aber keinen Sinn ergeben und sogar der Korrektheit widersprechen, denn daraus wäre auch die Gleichheit von  $a$  und  $b$  ableitbar, die aber keinesfalls aus einer Menge  $E$  von Gleichungen, deren Vervollständigung  $\mathcal{R}^\infty$  ist, folgerbar ist. □

Die Ableitbarkeitsrelation  $\xrightarrow{*}_{\mathcal{R}^\infty}$  hat aber, wenn sie auch im allgemeinen nicht konfluent ist, eine „schwache“ Konfluenzeigenschaft, ohne die das gesamte Verfahren nicht beweiskonfluent wäre, und der Vollständigkeitsatz 4.8.15 nicht gelten würde.

### Satz 4.9.2

Sei  $\mathcal{R}^\infty$  eine faire Vervollständigung,  $\mathbf{s}$ ,  $\mathbf{t}_1$  und  $\mathbf{t}_2$  seien Terme mit Constraint, es gelte

$$\mathbf{s} \xrightarrow{*}_{\mathcal{R}^\infty} \mathbf{t}_1 \quad \text{und} \quad \mathbf{s} \xrightarrow{*}_{\mathcal{R}^\infty} \mathbf{t}_2 ,$$

und die Constraints  $c_1$  und  $c_2$  der Terme  $\mathbf{t}_1$  und  $\mathbf{t}_2$  seien kompatibel, d. h. es sei

$$c_1 \sqcup c_2 \neq \top .$$

Dann gibt es Terme mit Constraint  $\mathbf{u}_1$  und  $\mathbf{u}_2$ , so daß

$$\mathbf{t}_1 \xrightarrow{\star}_{\mathcal{R}^\infty} \mathbf{u}_1 \quad \text{und} \quad \mathbf{t}_2 \xrightarrow{\star}_{\mathcal{R}^\infty} \mathbf{u}_2$$

und  $\mathbf{u}_1$  und  $\mathbf{u}_2$  eine gemeinsame Instanz besitzen.<sup>32</sup>

**Beweis:**

Sei  $\mathcal{R}^\infty$  eine Vervollständigung der Menge  $E$  von Gleichungen. Aus dem Lemma über die Korrektheit der Vervollständigung (Lemma 4.7.5) folgt, daß es einen minimalen Grund-Unifikator  $\sigma$  von  $\langle E, s, t_1 \rangle$  und von  $\langle E, s, t_2 \rangle$  und also von  $\langle E, t_1, t_2 \rangle$  geben muß (dabei sind  $s, t_1, t_2$  die Terme in  $\mathbf{s}, \mathbf{t}_1, \mathbf{t}_2$ ).

Dann folgt aber aus dem Vollständigkeitsatz 4.8.15 die Behauptung. ■

Die (entsprechend definierte) schwache lokale Konfluenz der Relation  $\xrightarrow{\star}_{\mathcal{R}^\infty}$  ist schon daraus ableitbar, daß  $\mathcal{R}^\infty$  keine kritischen Paare mehr enthält. Da aber  $\xrightarrow{\star}_{\mathcal{R}^\infty}$  im allgemeinen nicht wohlfundiert ist, folgt die schwache Konfluenz nicht ohne weiteres mit Lemma 2.1.5 (bzw. dem Analogon zu diesem Lemma für schwache Konfluenz), sondern sie muß auf anderem Wege bewiesen werden — wie im Beweis zu Satz 4.8.15 geschehen.

Daß die Relation  $\xrightarrow{\star}_{\mathcal{R}^\infty}$  nicht wohlfundiert ist, ist — unabhängig von der Frage der (schwachen) Konfluenz — von Nachteil. Das führt nämlich dazu, daß das auf ihr basierende Verfahren für die gemischte  $E$ -Unifikation nur ein Semi-Entscheidungsverfahren für die Existenz eines Unifikators ist, und zwar — anders als beim Unfailing Knuth-Bendix-Algorithmus — auch dann, wenn die Vervollständigung  $\mathcal{R}^\infty$  endlich ist.

Es ist ein noch offenes Problem, Kriterien für die Wohlfundiertheit von  $\xrightarrow{\star}_{\mathcal{R}^\infty}$  anzugeben. Es scheint, als wäre  $\xrightarrow{\star}_{\mathcal{R}^\infty}$  für rein starre Probleme stets wohlfundiert. Zumindest ist das bei allen mit der Implementierung des Verfahrens behandelten Beispielen der Fall. Dafür spricht auch, daß die rein starre  $E$ -Unifikation tatsächlich entscheidbar ist [Gallier *et al.*, 1992].

## 4.10 Lösung simultaner $E$ -Unifikationsprobleme

Will man neben einfachen auch simultane gemischte  $E$ -Unifikationsprobleme

$$\{\langle E_1, s_1, t_1 \rangle, \dots, \langle E_n, s_n, t_n \rangle\} \quad (n > 1)$$

lösen — für die Gleichheitsbehandlung im Tableauekalkül ist das unerläßlich —, reicht es nicht aus, die Mengen  $\mathcal{C}(\langle E_k, s_k, t_k \rangle)$  ( $1 \leq k \leq n$ ) für die Einzelprobleme zu bestimmen, denn  $\text{Sat}^*(\mathcal{C}(\langle E_k, s_k, t_k \rangle))$  ist nur bezüglich der Subsumtionsrelation  $\leq_{E_i}$  grund-vollständig, nicht aber bezüglich  $\leq$ .

<sup>32</sup> Das heißt es gibt eine Substitution  $\sigma$ , so daß  $\{\mathbf{u}_1\}[\sigma] \cap \{\mathbf{u}_2\}[\sigma] \neq \emptyset$ .

Wie schon in Abschnitt 3.3 ausgeführt, ist es zur Berechnung einer (grund-)vollständigen Menge von Unifikatoren des simultanen Problems erforderlich, daß zu gegebenen Substitutionen  $\sigma_1, \sigma_2$  die Menge aller  $\tau$ , so daß

$$\sigma_1 \leq \tau \quad \text{und} \quad \sigma_2 \leq \tau ,$$

einfach zu bestimmen ist (dabei bezeichnet  $\leq$  die jeweils verwendete Subsumtionsrelation).<sup>33</sup> Ein entsprechendes einfaches Verfahren existiert für die Subsumtionsrelation  $\leq_E$  im allgemeinen jedoch nicht.<sup>34</sup>

### Beispiel 4.10.1

Zu lösen sei das aus den beiden  $E$ -Unifikationsproblemen

$$\begin{aligned} \langle E_1, s_1, t_1 \rangle &= \langle f(f(a)) \approx a, x, f(a) \rangle \\ \langle E_2, s_2, t_2 \rangle &= \langle f(f(f(a))) \approx a, x, f(f(a)) \rangle \end{aligned}$$

bestehende simultane Problem.

Alle Substitutionen der Form  $\{x/f^{2n+1}(a)\}$  ( $n \geq 0$ ) sind, wie man leicht nachvollzieht, Lösung des Problems  $\langle E_1, s_1, t_1 \rangle$ , und alle Substitutionen  $\{x/f^{3m+2}(a)\}$  ( $m \geq 0$ ) sind Lösung des Problems  $\langle E_2, s_2, t_2 \rangle$ .

Die *minimalen* Unifikatoren der beiden Probleme sind

$$\tau_1 = \{x/f(a)\} \quad \text{und} \quad \tau_2 = \{x/f(f(a))\} .$$

Beide sind keine Unifikatoren des jeweils anderen Problems.

Die kleinste ganzzahlige Lösung der Gleichung  $k = 2n + 1 = 3m + 2$  ist  $k = 5$ . Der minimale Unifikator des simultanen Problems ist daher  $\tau_2 = \{x/f^5(a)\}$ .  $\square$

Es ist also notwendig, aus den bezüglich  $\leq_E$  grund-vollständigen Mengen

$$\mathcal{U}_k = \text{Sat}^*(\mathcal{C}(\langle E_k, s_k, t_k \rangle))$$

Mengen  $\hat{\mathcal{U}}_k$  zu bestimmen, die grund-vollständig sind bezüglich  $\leq$ . Die folgende Überlegung zeigt, wie dies geschehen kann:

Ist  $\sigma$  ein beliebiger Grund-Unifikator eines (nicht simultanen) gemischten  $E$ -Unifikationsproblems  $\langle E, s, t \rangle$ , dann muß es in der Menge  $\mathcal{U} = \text{Sat}^*(\mathcal{C}(\langle E, s, t \rangle))$  wegen deren Grund-Vollständigkeit einen Unifikator  $\tau$  geben, so daß  $\tau \leq_E \sigma$ , d. h.,  $\sigma$  ist aus  $\tau$  mit Gleichungen in  $E\sigma$  ableitbar (gemeint ist die Ableitbarkeit, wie sie in Definition 3.3.2 definiert ist). Wie der Beweis zu Satz 4.8.17 zeigt, muß es insbesondere auch ein solches  $\tau \in \mathcal{U}$  mit  $\tau \leq_E \sigma$  geben, das einen Grund-Unifikator  $\tau'$  als Instanz hat — also  $\tau \leq \tau'$  — für den gilt:

1.  $\tau' \leq_E \sigma$ ,

<sup>33</sup> In Abschnitt 3.3 ist diese Eigenschaft der Subsumtionsrelation auf den Substitutionen mit (S4) bezeichnet.

<sup>34</sup> Letzten Endes ist jedes Verfahren, das — wie auch immer — Lösungen simultaner gemischter  $E$ -Unifikationsprobleme aufzählt, ein Verfahren zur Bestimmung solcher Kombinationen  $\tau$ , insbesondere auch das im folgenden beschriebene. Alle diese Methoden können aber schwerlich als „einfach“ bezeichnet werden.

2.  $\tau'$  ist minimal bezüglich der Ordnung  $\succ_x$  auf den Substitutionen.

Die Minimalität von  $\tau'$  überträgt sich in gewisser Weise — da  $\tau \leq \tau'$  gilt, und nicht etwa nur  $\tau \leq_E \tau'$  — auch auf den Unifikator  $\tau$ . Daraus kann man nun schließen, daß  $\sigma$  aus  $\tau$  nicht nur mit Gleichungen aus  $E\sigma$  ableitbar ist, sondern auch mit Regeln aus  $\mathcal{R}^\infty$  ( $\mathcal{R}^\infty$  ist die faire Vervollständigung von  $E$ , die zur Berechnung von  $\mathcal{C}(\langle E, s, t \rangle)$  verwendet wurde). Benutzt man Regeln aus der Inversen  $(\mathcal{R}^\infty)^-$  von  $\mathcal{R}^\infty$ , muß es umgekehrt auch möglich sein,  $\sigma$  aus  $\tau$  zu bestimmen. Damit bei solchen inversen Regelanwendungen nicht auch Substitutionen abgeleitet werden, die keine Unifikatoren sind, muß man darauf achten, daß die neu entstehende Substitution die Constraints der zur Ableitung verwendeten Regeln erfüllt. Satz 4.10.3 zeigt, daß diese Überlegung richtig ist, daß es möglich ist, eine bezüglich  $\leq$  grund-vollständige Menge von Unifikatoren zu bestimmen, indem man das inverse Reduktionssystem  $(\mathcal{R}^\infty)^-$  auf die Substitutionen in der bezüglich  $\leq_E$  grund-vollständigen Menge  $\mathcal{U}$  anwendet.

In der folgenden Definition wird zunächst präzisiert, was unter der Anwendung einer Reduktionsregel mit Constraint auf eine Substitution zu verstehen ist:

**Definition 4.10.2** (Die Ableitbarkeitsrelation  $\Rightarrow_{\mathcal{R}}$  auf Substitutionen)

Sei  $R$  ein Reduktionssystem. Die Relation  $\Rightarrow_{\mathcal{R}}$  wird wie folgt auf die Menge der Substitutionen erweitert: Für Substitutionen  $\sigma$  und  $\tau$  gilt

$$\sigma \Rightarrow_{\mathcal{R}} \tau$$

genau dann, wenn es eine Variable  $x$  gibt, so daß:

1. Für alle Variablen  $y \neq x$  gilt  $\sigma(y) = \tau(y)$ , d. h., auf  $\mathcal{V} \setminus \{x\}$  stimmen  $\sigma$  und  $\tau$  überein.

2. Es gilt

$$(\sigma(x) \ll \langle \sigma, \text{true} \rangle) \Rightarrow_{\mathcal{R}} (\tau(x) \ll c)$$

(hier ist  $\Rightarrow_{\mathcal{R}}$  die Ableitbarkeitsrelation auf Termen mit Constraint, Def. 4.3.6).

3.  $\tau$  erfüllt den Constraint  $c$ .

**Satz 4.10.3**

Sei  $\langle E, s, t \rangle$  ein gemischtes  $E$ -Unifikationsproblem, und die Menge

$$\mathcal{U} = \text{Sat}^*(\mathcal{C}(\langle E, s, t \rangle))$$

mit Hilfe einer fairen Vervollständigung von  $E$  bestimmt (Def. 4.6.6).<sup>35</sup> Dann ist die Menge  $\hat{\mathcal{U}}(\langle E, s, t \rangle)$  von Substitutionen, die die kleinste Menge ist, die

- $\mathcal{U}$  als Teilmenge enthält,
- abgeschlossen ist bezüglich der Ableitbarkeitsrelation  $\overset{*}{\Rightarrow}_{(\mathcal{R}^\infty)^-}$  (Def. 4.10.2),

<sup>35</sup> Es ist auch möglich  $\mathcal{U} = \text{Sat}(\mathcal{C}_\infty(\langle E, s, t \rangle))$  (Def. 4.6.1) zu verwenden.

eine Menge von Unifikatoren von  $\langle E, s, t \rangle$ , die grund-vollständig ist bezüglich der Subsumtionsrelation  $\leq$ .

**Beweis:**

Zunächst ist zu zeigen, daß  $\hat{\mathcal{U}}(\langle E, s, t \rangle)$  eine Menge von Unifikatoren ist. Da  $\mathcal{U}$  gemäß Satz 4.7.7 eine Menge von Unifikatoren ist, genügt es zu zeigen, daß  $\Rightarrow_{(\mathcal{R}^\infty)^-}$ - Teilrelation der Subsumtionsrelation  $\sqsubseteq_E$  ist, die nach Lemma 3.3.5 abgeschlossen ist bezüglich der Eigenschaft Unifikator zu sein.

Seien also  $\sigma$  und  $\tau$  beliebige Substitutionen mit  $\sigma \Rightarrow_{(\mathcal{R}^\infty)^-} \tau$ . Sei  $x$  die Variable, für die

$$(\sigma(x) \ll \langle \sigma, \text{true} \rangle) \Rightarrow_{\mathcal{R}} (\tau(x) \ll c)$$

gilt. Da  $\mathcal{R}^\infty$  korrekt ist (Lemma 4.7.5), ist auch  $(\mathcal{R}^\infty)^-$  korrekt bezüglich  $E$ . Da die Substitution  $\tau$  den Constraint  $c$  erfüllt (Def. 4.10.2), ist sie ein Unifikator von  $\langle E, \sigma(x), \tau(x) \rangle$ .  $\tau$  ist auch Unifikator von  $\langle E, \sigma(y), \tau(y) \rangle$  für alle Variablen  $y \in (\mathcal{V} \setminus \{x\})$ , denn auf diesen stimmen  $\sigma$  und  $\tau$  überein. Also gilt  $\sigma \sqsubseteq_E \tau$  (Def. 3.3.2).

Daß  $\hat{\mathcal{U}}$  grund-vollständig ist, kann man durch wohlfundierte Induktion bezüglich der Ordnung  $\succ_{\langle E, s, t \rangle}$  zeigen.

*Induktionsanfang:* Sei  $\sigma$  ein minimaler Grund-Unifikator. Dann erfüllt  $\sigma$ , wie in dem Beweis zu Satz 4.8.17 gezeigt wurde, einen Constraint in der Menge  $\mathcal{C}(\langle E, s, t \rangle)$ , liegt also in  $\mathcal{U} \subset \hat{\mathcal{U}}$ .

*Induktionsschluß:* Sei  $\sigma$  ein Grund-Unifikator, der nicht minimal ist. Es gebe also einen Grund-Unifikator  $\tau$  mit  $\tau \leq_E \sigma$  und  $\sigma \succ_{\langle E, s, t \rangle} \tau$ . Dann muß es aber auch einen Grund-Unifikator  $\sigma'$  geben,

1. der sich von  $\sigma$  nur in der Belegung *einer* Variablen  $x$  unterscheidet,
2. für den  $\sigma' \leq_E \sigma$  und  $\sigma \succ_{\langle E, s, t \rangle} \sigma'$  gilt, und
3. wegen der Vollständigkeit von  $\mathcal{R}^\infty$  (Satz 4.8.17)  $(\sigma(x) \ll \epsilon) \Rightarrow_{\mathcal{R}^\infty} (\sigma'(x) \ll c')$  gilt.

Auf  $\sigma'$  ist wegen  $\sigma \succ_{\langle E, s, t \rangle} \sigma'$  die Induktionsannahme anwendbar. Es gibt also eine Substitution  $\sigma'' \in \mathcal{U}$  mit  $\sigma'' \leq \sigma'$ . Es gilt nun entweder auch  $\sigma'' \leq \sigma$ , oder aber es folgt aus  $(\sigma(x) \ll \epsilon) \Rightarrow_{\mathcal{R}^\infty} (\sigma'(x) \ll c')$  (da  $\sigma'$  Grundsubstitution ist)

$$(\sigma''(x) \ll \epsilon) \Rightarrow_{(\mathcal{R}^\infty)^-} (\sigma'''(x) \ll c) .$$

Dabei ist  $\sigma'''$  eine Substitution, die mit  $\sigma''$  auf allen Variablen außer  $x$  übereinstimmt.  $\sigma'''$  erfüllt  $c$ , liegt also in  $\hat{\mathcal{U}}$ , und es gilt  $\sigma''' \leq \sigma$ . ■

**Beispiel 4.10.4**

Das folgende ist ein Beispiel dafür, daß die dritte Bedingung in Definition 4.10.2, daß nämlich die abgeleitete Substitution  $\tau$  den Constraint  $c$  erfüllen muß, tatsächlich notwendig ist.

Gegeben sei das  $E$ -Unifikationsproblem  $\langle E, s, t \rangle = \langle \{x \approx b\}, a, x \rangle$ . Es hat die Lösung  $\sigma = \{x/a\}$ .

Mit Hilfe der in der Inversen  $(\mathcal{R}^\infty)^-$  einer Vervollständigung von  $E$  liegenden Reduktionsregel  $x \rightsquigarrow b \ll \langle id, b \succ a \rangle$  kann aus dem Term  $a \ll \langle x/a, \text{true} \rangle$  der Term  $b \ll \langle \{x/a\}, b \succ a \rangle$  abgeleitet werden. Dennoch gilt für  $\tau = \{x/b\}$ , da  $\tau$  den Constraint  $\langle x/a, \{b \succ a\} \rangle$  nicht erfüllt,  $\sigma \Rightarrow_{(\mathcal{R}^\infty)^-} \tau$  nicht. Das entspricht der Tatsache, daß  $\tau$  kein Unifikator ist.  $\square$

Mit Satz 4.10.3 ist es möglich, bezüglich  $\leq$  grund-vollständige Mengen von Unifikatoren für die Einzelprobleme eines simultanen gemischten  $E$ -Unifikationsproblems zu berechnen (bzw. aufzuzählen). Damit kann nun auch eine grund-vollständige Menge von Unifikatoren eines simultanen gemischten Problems aufgezählt werden:

**Satz 4.10.5**

Sei  $\{\langle E_1, s_1, t_1 \rangle, \dots, \langle E_n, s_n, t_n \rangle\}$  ein simultanes gemischtes  $E$ -Unifikationsproblem, und seien die Mengen

$$\hat{\mathcal{U}}(\langle E_k, s_k, t_k \rangle) \quad (1 \leq k \leq n)$$

wie in Satz 4.10.3 definiert. Dann ist

$$\vec{\mathcal{U}} = \{\sigma_1 \sqcup \dots \sqcup \sigma_n : \sigma_k \in \hat{\mathcal{U}}(\langle E_k, s_k, t_k \rangle) \ (1 \leq k \leq n)\} \setminus \{\top\}$$

eine grund-vollständige Menge von Unifikatoren des Problems.

**Beweis:**

Sei  $\sigma \in \vec{\mathcal{U}}$  beliebig. Dann gibt es Substitutionen  $\sigma_1, \dots, \sigma_n$ , so daß  $\sigma$  Spezialisierung aller dieser Substitutionen ist und  $\sigma_k \in \hat{\mathcal{U}}(\langle E_k, s_k, t_k \rangle)$  Unifikator von  $\langle E_k, s_k, t_k \rangle$  ( $1 \leq k \leq n$ ). Daraus folgt, daß auch  $\sigma$  Unifikator von  $\langle E_k, s_k, t_k \rangle$  ist (Satz 3.2.6); und da dies für  $1 \leq k \leq n$  gilt, ist  $\sigma$  ein Unifikator des simultanen Problems.

Sei nun  $\sigma$  ein beliebiger Grund-Unifikator des simultanen Problems. Dann ist  $\sigma$  auch Unifikator der Einzelprobleme  $\langle E_k, s_k, t_k \rangle$  ( $1 \leq k \leq n$ ). Da  $\hat{\mathcal{U}}(\langle E_k, s_k, t_k \rangle)$  gemäß Satz 4.10.3 grund-vollständig ist bezüglich  $\leq$ , muß es  $\sigma_k \in \hat{\mathcal{U}}(\langle E_k, s_k, t_k \rangle)$  geben mit  $\sigma_k \leq \sigma$ . Dann gilt aber für die Substitution  $\sigma' = \sigma_1 \sqcup \dots \sqcup \sigma_n$

1.  $\sigma' \in \vec{\mathcal{U}}$ ,
2.  $\sigma' \leq \sigma$ .

Also ist  $\vec{\mathcal{U}}$  grund-vollständig.  $\blacksquare$

Da man normalerweise nur einen Unifikator eines simultanen Problems benötigt (insbesondere ist das bei der Gleichheitsbehandlung im Tableauekalkül der Fall, siehe Abschnitt 5.2.3), wird man statt  $\vec{\mathcal{U}}$  zunächst die Menge  $\vec{\mathcal{U}}' \subset \vec{\mathcal{U}}$  betrachten, die definiert ist durch

$$\vec{\mathcal{U}}' = \{\sigma_1 \sqcup \dots \sqcup \sigma_n : \sigma_k \in \mathcal{U}(\langle E_k, s_k, t_k \rangle) \ (1 \leq k \leq n)\} \setminus \{\top\}$$

Zur Berechnung von  $\vec{\mathcal{U}}'$  verwendet man also die nur bezüglich  $\leq_{E_k}$ , nicht aber bezüglich  $\leq$  grund-vollständigen Mengen  $\mathcal{U}(\langle E_k, s_k, t_k \rangle)$ . Nur wenn  $\vec{\mathcal{U}}'$  leer ist,<sup>36</sup>

<sup>36</sup> In Fällen, in denen  $\vec{\mathcal{U}}'$  nur aufzählbar aber nicht berechenbar ist, dann, wenn man nach einer gewissen Rechenzeit kein Element der Menge gefunden hat.



wird man beginnen, weitere Unifikatoren der Einzelprobleme, die nicht schon in der Menge  $\mathcal{U}(\langle E_k, s_k, t_k \rangle)$  liegen, abzuleiten. Durch die Verwendung von Regeln aus der Inversen  $(\mathcal{R}^\infty)^-$  kann eine sehr große Zahl von Unifikatoren abgeleitet werden. Das führt zu einer Explosion des Suchraums, und jeder Vorteil der *vervollständigungs-basierten E*-Unifikation geht verloren. Dann wird ein äquivalenzklassenbasiertes Verfahren — in ineffizienter Weise — simuliert. Glücklicherweise hat sich aber gezeigt, daß in der Praxis nur für sehr wenige simultane Probleme in  $\vec{\mathcal{U}}$  kein Unifikator gefunden werden kann. Bei allen Experimenten trat dies nur bei eigens konstruierten Problemen auf (wie dem aus Beispiel 4.10.4, siehe auch Abschnitt 7.1.4).

## 4.11 Beziehung zu Conditional Rewriting

Beim Conditional Rewriting<sup>37</sup> werden die Reduktionsregeln — ähnlich wie bei dem in diesem Kapitel vorgestellten Verfahren — mit Bedingungen versehen. Diese Bedingungen sind aber grundlegend verschieden von den hier verwendeten Constraints. Meist werden Bedingungen der allgemeinen Form

$$(s_1 \approx t_1) \wedge \dots \wedge (s_n \approx t_n) \quad (n \geq 0)$$

verwendet.<sup>38</sup>

Auf der einen Seite sind diese Bedingungen sehr viel ausdrucksstärker als Constraints, denn das in ihnen enthaltene Prädikatensymbol  $\approx$  wird nicht als syntaktische Gleichheit auf den Termen interpretiert, sondern als Gleichheit bezüglich des Reduktionssystems bzw. bezüglich der zugrunde gelegten Menge  $E$  von Gleichungen. Eine Substitution  $\sigma$  erfüllt also die Bedingung  $s \approx t$  nicht nur, wenn  $s\sigma = t\sigma$ , sondern auch schon dann, wenn  $\sigma$  ein  $E$ -Unifikator von  $s$  und  $t$  ist. Auf der anderen Seite lassen sich aber die Ordnungsbedingungen, die Teil der Constraints sind, mit diesen Bedingungen nicht ausdrücken. Ließe man die obige Einschränkung der Form von Bedingungen fallen, ginge man also zu beliebigen Formeln der Prädikatenlogik mit Gleichheit über, wären die Bedingungen nicht mehr handhabbar; vor allem wäre ihre Erfüllbarkeit nicht mehr entscheidbar. Zudem müßte man dann die verwendete Reduktionsordnung auf der Objektebene axiomatisieren, was völlig ineffizient wäre.

Nichtsdestotrotz steht das vorgestellte Verfahren für die gemischte  $E$ -Unifikation in einem engen Zusammenhang mit Conditional Rewriting, und es könnte von Vorteil sein, eine Kombination beider Methoden zu entwickeln.

<sup>37</sup> Eine Einführung gibt [Kaplan & Rémy, 1989].

<sup>38</sup> Oft werden zusätzliche Einschränkungen gemacht, die die Handhabbarkeit der Bedingungen verbessern.

# 5 Der Tableauealkül mit freien Variablen und Gleichheit

---

**Tableau** [ta'blo:] das; -s, -s;  
*wirkungsvoll gruppiertes Bild*

— DUDEN, Die Rechtschreibung der deutschen Sprache

## 5.1 Der Tableauealkül mit freien Variablen

### 5.1.1 Einführung

In seinen Ursprüngen geht der Tableauealkül zurück auf die Arbeiten von G. Gentzen in den dreißiger und E. W. Beth und K. J. J. Hintikka in den fünfziger Jahren. R. Smullyan [Smullyan, 1968] entwickelte die — im wesentlichen unverändert — bis heute verwendete Version des Tableauealküls, die auch den Kern des im folgenden verwendeten Kalküls bildet. Auf ihn geht die Einteilung der Tableauregeln in  $\alpha$ -Regeln (für konjunktive Formeln),  $\beta$ -Regeln (für disjunktive Formeln),  $\gamma$ -Regeln (für universell quantifizierte Formeln) und  $\delta$ -Regeln (für existentiell quantifizierte Formeln) zurück.

Erst durch in den letzten Jahren entwickelte Verbesserungen allerdings wurde eine effiziente Implementierung des Tableauealküls möglich, von denen die wichtigste die Verwendung freier Variablen ist. Da ein wesentliches Ziel dieser Arbeit eine effiziente Implementierung ist, wird ihr ein solch verbesserter Kalkül zugrunde gelegt. Die im weiteren entwickelten Methoden zur Behandlung von Gleichheit lassen sich aber ohne weiteres auch auf andere Versionen des Tableauealküls übertragen (siehe Abschnitt 5.2.5).

Bevor formale Definitionen gegeben werden, seien die wesentlichen Eigenschaften des verwendeten Kalküls, die ihn von anderen Versionen unterscheiden, hier kurz zusammengefaßt. Sie sind auch die wesentlichen Merkmale des automatischen Beweisers  $\mathcal{I}^{\mathcal{A}P}$ , auf dem die Implementierung (Kapitel 6) aufsetzt.

**Signierte Formeln** Der Kalkül verwendet signierte Formeln, die mit einem der Vorzeichen T (für wahr) und F (für falsch) versehen sind.

**Freie Variablen** Die Verwendung freier Variablen [Fitting, 1990] ist eine wichtige Optimierung. Anstatt schon bei der Anwendung einer  $\gamma$ -Regel einen Grundterm zu „raten“ und für die gebundene Variable einzusetzen, wird diese durch eine freie Variable ersetzt. Die freien Variablen werden erst später mit Termen instantiiert, die erkennbar für den Abschluß eines Astes sinnvoll sind.

**Liberalisierte  $\delta$ -Regel** Um die Korrektheit des Kalküls zu erhalten, müssen bei Verwendung freier Variablen auch die  $\delta$ -Regeln, nämlich der Aufbau der durch sie eingeführten Skolemterme, verändert werden. Unglücklicherweise wird dadurch auch die Zahl möglicher Abschlüsse verringert. Es ist allerdings möglich, die in [Fitting, 1990] angegebene  $\delta$ -Regel für freie Variablen etwas zu liberalisieren [Hähnle & Schmitt, 1993, Beckert *et al.*, 1993], so daß eine größere Zahl von Abschlüssen möglich ist, und die Korrektheit doch erhalten bleibt.

**Universelle Formeln**  $\gamma$ -Formeln werden oft mehrfach zum Abschluß eines Astes benötigt (das trifft ganz besonders häufig auf Gleichungen zu). Normalerweise werden hierzu mehrere Instanzen einer solchen Formel durch Anwenden der  $\gamma$ -Regel erzeugt. Bei den „universellen“ Formeln ist das nicht nötig; sie können — ohne die Korrektheit des Kalküls zu beeinträchtigen — mehrfach, mit verschiedenen Instantiierungen verwendet werden [Beckert, 1991, Beckert & Hähnle, 1992].

Ein Tableauekalkül mit freien Variablen — allerdings ohne universelle Formeln und die liberalisierte  $\delta$ -Regel — ist auch in allen Einzelheiten in [Fitting, 1990] beschrieben, und wird dort um ein einfaches Verfahren zur Behandlung von Gleichheit erweitert (siehe Abschnitt 5.2.2).

Eine aktuelle Darstellung des automatischen Beweisens mit dem Tableauekalkül, insbesondere auch neuer, effizienter Verfahren zur Implementierung, findet sich in [Hähnle, 1993].

### 5.1.2 Syntax und Semantik

In diesem Abschnitt wird Syntax und Semantik des Tableauekalküls mit freien Variablen beschrieben — zunächst ohne Gleichheit, d. h. dem Gleichheitsprädikat  $\approx$  wird keine Sonderrolle zuerkannt.

Wie schon erwähnt, werden signierte, d. h. mit Vorzeichen T und F versehene, Formeln verwendet. Sie seien über einer Signatur  $\mathcal{S}$  definiert, die die verwendeten Skolemsymbole schon enthält.<sup>1</sup>

**Definition 5.1.1** (Signierte Formeln)

Sei **Form** die Menge der Formeln über einer Signatur  $\mathcal{S}$ . Dann ist

$$\mathbf{Form}^* = \{T \phi : \phi \in \mathbf{Form}\} \cup \{F \phi : \phi \in \mathbf{Form}\}$$

die Menge der signierten Formeln über  $\mathcal{S}$ .

Erst die Gliederung der Tableauregeln nach den Schemata, denen sie folgen, erlaubt eine elegante Formulierung des Tableauekalküls:

<sup>1</sup> Dies beschränkt die Allgemeinheit nicht; nötigenfalls werde die Signatur um unendlich viele Skolemsymbole für jede Stelligkeit  $n \geq 0$  erweitert.

**Definition 5.1.2** (Tableauregel-Schemata)

In Tabelle 5.1 sind die vier Schemata dargestellt, denen alle Tableaureweiterungsregeln folgen. Oberhalb der waagerechten Linien sind die Prämissen angegeben, unterhalb die Konklusionen, die sich in durch eine senkrechte Linie getrennte Extensionen gliedern können.<sup>2</sup>

$\alpha$	$\beta$	$\gamma$	$\delta$
$\alpha_1$	$\beta_1$	$\beta_2$	$\gamma_1(y)$
$\alpha_2$			$\delta_1(f(x_1, \dots, x_n))$
wobei $y$ eine freie Variable ist.			wobei $f$ eine neue Skolemfunktion ist, und $\{x_1, \dots, x_n\}$ die Menge der in $\delta$ frei auftretenden Variablen.

**Tabelle 5.1:** Die Tableauregel-Schemata.

Für jede Kombination von Vorzeichen und logischem Operator ist genau eine Tableauregel definiert; jede dieser Regeln folgt einem der Schemata aus Tabelle 5.1.

**Definition 5.1.3** (Tableauregeln)

Auf jede nicht-atomare signierte Formeln ist genau eine Tableauregel anwendbar. Die Tabellen 5.2 (a) bis (d) geben für jeden Formeltyp die entsprechende Tableauregel an und welchem Schema (Def. 5.1.2) sie folgt.

$\alpha$	$\alpha_1$	$\alpha_2$
$\top (\phi \wedge \psi)$	$\top \phi$	$\top \psi$
$\text{F} (\phi \vee \psi)$	$\text{F} \phi$	$\text{F} \psi$
$\text{F} (\phi \supset \psi)$	$\top \phi$	$\text{F} \psi$
$\top \neg \phi$	$\text{F} \phi$	$\text{F} \phi$
$\text{F} \neg \phi$	$\top \phi$	$\top \phi$

(a)

$\beta$	$\beta_1$	$\beta_2$
$\top (\phi \vee \psi)$	$\top \phi$	$\top \psi$
$\text{F} (\phi \wedge \psi)$	$\text{F} \phi$	$\text{F} \psi$
$\top (\phi \supset \psi)$	$\text{F} \phi$	$\top \psi$

(b)

$\gamma$	$\gamma_1(y)$
$\top (\forall x)\phi(x)$	$\top \phi(y)$
$\text{F} (\exists x)\phi(x)$	$\text{F} \phi(y)$

(c)

$\delta$	$\delta_1(f(x_1, \dots, x_n))$
$\text{F} (\forall x)\phi(x)$	$\text{F} \phi(f(x_1, \dots, x_n))$
$\top (\exists x)\phi(x)$	$\top \phi(f(x_1, \dots, x_n))$

(d)

**Tabelle 5.2:** (a)  $\alpha$ -Regeln; (b)  $\beta$ -Regeln; (c)  $\gamma$ -Regeln; (d)  $\delta$ -Regeln.

Es ist möglich, die  $\delta$ -Regel noch weiter zu liberalisieren, und die Einschränkung fallen zu lassen, daß das bei der Anwendung der  $\delta$ -Regel eingeführte Skolemsymbol stets neu sein muß [Beckert *et al.*, 1993]. Der Tableaurekalkül bleibt korrekt, wenn man für  $\delta$ -Formeln, die bis auf Variablenumbenennung gleich sind, jeweils das gleiche Skolemsymbol verwendet.

<sup>2</sup> Implizit sind die Extensionen disjunktiv, die Formeln innerhalb einer Extension konjunktiv verknüpft.

Alles folgende gilt ohne jede Einschränkung auch für diese noch weiter liberalisierte  $\delta$ -Regel<sup>3</sup> (im Zweifel sei jedoch auch weiter die  $\delta$ -Regel aus Definition 5.1.2 gemeint).

Tableaus können charakterisiert werden als diejenigen endlichen binären Bäume, deren Knoten signierte Formeln sind,<sup>4</sup> und die entsprechend der Tableauregeln aus Definition 5.1.2 und 5.1.3 aufgebaut sind:

**Definition 5.1.4** (Tableau, Ast eines Tableaus)

Sei  $\mathbf{Form}^*$  die Menge signierter Formeln über einer Signatur  $\mathcal{S}$ . Tableaus über  $\mathcal{S}$  sind genau diejenigen endlichen binären Bäume, die entsprechend einer der folgenden Regeln aufgebaut sind:

1. Ein linearer Baum  $T$ , dessen Knoten geschlossene Formeln aus  $\mathbf{Form}^*$  sind, ist ein Tableau (über  $\mathcal{S}$ ). (Ist  $\Phi$  die Menge der Formeln, die Knoten von  $T$  sind, dann heie  $T$  ein Tableau für  $\Phi$ .)
2. Ist  $T$  ein Tableau (über  $\mathcal{S}$ ) und  $\phi$  ein Knoten in  $T$ , dann ist ein Baum  $T'$  ein Tableau (über  $\mathcal{S}$ ), der dadurch aus  $T$  entsteht, daß an einen maximalen Pfad in  $T$  (einen Ast von  $T$ ) so viele lineare Unterbäume angeheftet werden, wie die auf  $\phi$  anwendbare Tableauregel (Def. 5.1.2 und 5.1.3) Extensionen hat, wobei die Knoten der Unterbäume die Formeln aus den Extensionen sind. (Ist  $T$  ein Tableau für eine Menge  $\Phi$  von Formeln, dann ist auch  $T'$  ein Tableau für  $\Phi$ .)

Die maximalen Pfade in einem Tableau  $T$  heißen die Äste von  $T$ . Ein Ast wird häufig auch mit der Menge der auf ihm liegenden Formeln identifiziert.

**Definition 5.1.5** (Komplementäre Formeln, geschlossener Ast, geschlossenes Tableau)

Zwei Formeln  $\top \phi$  und  $\mathbf{F} \phi$ , die sich nur in ihrem Vorzeichen unterscheiden, heißen komplementär.

Ein Ast  $A$  eines Tableaus heie geschlossen, wenn er komplementäre Formeln enthält. Gibt es eine Substitution  $\sigma$ , so daß  $A\sigma$  geschlossen ist, so heie  $A$  geschlossen unter  $\sigma$ .

Sei  $T$  ein Tableau. Gib es eine Grundsubstitution  $\sigma$ , so daß alle Äste von  $T$  unter  $\sigma$  geschlossen sind, dann heie  $T$  geschlossen (unter  $\sigma$ ).

Die Forderung, daß  $\sigma$  eine Grundsubstitution sei, kann man fallen lassen. Allerdings müssen auch dann die Äste von  $T$  tatsächlich unter  $\sigma$  geschlossen sein, und nicht etwa unter (verschiedenen) Spezialisierungen von  $\sigma$ .

Zu überprüfen, ob ein Tableau nach Definition 5.1.5 geschlossen ist, ist — im Unterschied zum Abschluß ohne freie Variablen — ein keineswegs triviales Problem, da eine das Tableau abschließende Substitution bestimmt werden muß.<sup>5</sup>

<sup>3</sup> Sie wird in [Beckert *et al.*, 1993] als  $\delta^{++}$ -Regel bezeichnet.

<sup>4</sup> Die Formeln werden also nicht als an die Knoten des Tableaus angeheftet angesehen, sondern mit den Knoten identifiziert.

<sup>5</sup> Nichtsdestoweniger ist es besser, freie Variablen zu verwenden. Auch ohne sie tritt das Problem, günstige Belegungen für die Variablen zu finden, auf, dann nämlich schon früher: bei der Anwendung der  $\gamma$ -Regeln.

Um zu beweisen, daß eine Formel  $G$  eine Tautologie ist, beginnt man mit einem Tableau, dessen einziger Knoten (seine Wurzel) die Formel  $\neg G$  ist. Die Erweiterung dieses Tableaus entspricht einer systematischen Suche nach einem Modell für  $\neg G$ . Dabei kann jeder Ast  $A$  als partielle Definition eines Modells angesehen werden, in dem die Formeln auf  $A$  den ihrem Vorzeichen entsprechenden Wahrheitswert haben. Da komplementäre Formeln nicht zugleich in einem Modell Gültigkeit haben können, kann ein geschlossener Ast kein Modell darstellen. Ein geschlossenes Tableau für  $\neg G$  zeigt, daß die Suche nach einem Modell vergeblich war, und also  $G$  eine Tautologie ist.

**Definition 5.1.6** (Tableaubeweis)

*Ein Tableau  $T$  ist ein Tableaubeweis für eine geschlossene Formel  $\phi$ , wenn*

1.  $T$  geschlossen ist, und
2.  $T$  ein Tableau ist für  $\{\neg \phi\}$ .

Die Suche nach einem Tableaubeweis für eine Formel ist ein in höchstem Maße indeterministischer Prozeß, da es in der Regel viele verschiedene Möglichkeiten gibt, ein gegebenes Tableau zu erweitern. Für eine Implementierung ist es jedoch notwendig, festzulegen, wie die jeweils nächste Tableauerweiterung auszuwählen ist. Es ist möglich, deterministische Auswahlkriterien (Konstruktionsregeln) anzugeben, mit deren Hilfe für eine Tautologie stets ein Tableaubeweis gefunden werden kann. Verwendet man eine solche Konstruktionsregel, ist der Tableaurekalkül beweiskonfluent, d. h., es ist kein Backtracking notwendig.

**Definition 5.1.7** (Faire Tableau-Konstruktionsregel)

*Eine Tableau-Konstruktionsregel legt fest, welche von mehreren möglichen Erweiterungsregeln als nächste auf ein Tableau anzuwenden ist. Sie stützt sich dabei auf das Tableau selbst, sowie auf eine Menge von zusätzlichen Informationen. Die Regel gibt an, wie diese zusätzlichen Informationen aufzubauen sind.*

*Eine Tableau-Konstruktionsregel heie fair, wenn jedes Tableau  $T_n$  ( $n \geq 0$ ) in einer gem der Regel konstruierten Folge  $T_1, T_2, \dots$  von Tableaus folgende Eigenschaften hat:*

1. Auf jede Formel in  $T_n$ , die kein Atom ist, wird in jedem Ast, in dem sie auftritt, die auf sie passende Tableau-Erweiterungsregel schlielich angewendet.
2. Auf jede  $\gamma$ -Formel in  $T_n$  wird in jedem Ast, in dem sie auftritt, die  $\gamma$ -Regel beliebig oft angewendet.

Ein Beispiel fur eine faire Tableau-Konstruktionsregel ist in [Fitting, 1990] angegeben. Die Existenz fairer Konstruktionsregeln ist also sichergestellt.<sup>6</sup>

---

<sup>6</sup> Die von dem Tableaubeweiser  $\mathcal{I}^{\mathcal{A}\mathcal{P}}$  verwendete faire Konstruktionsregel ist in [Hahnle *et al.*, 1992] beschrieben.

Es ist möglich, neben diesen notwendigen Anforderungen an eine faire Konstruktionsregel zusätzliche Einschränkungen zu machen, die das konstruierte Tableau verkleinern und damit auch die Suche nach einem Abschluß vereinfachen.

Beispielsweise ist es keinesfalls sinnvoll, auf Formeln, die keine  $\gamma$ -Formeln sind, die entsprechende Tableauregel auf dem gleichen Ast eines Tableaus mehr als einmal anzuwenden.<sup>7</sup> Bei komplizierteren Beschränkungen dagegen, wie beispielsweise solchen, die Links zwischen Formeln berücksichtigen, wäre im Einzelfall zu prüfen, ob sie auch im Zusammenspiel mit der Gleichheitsbehandlung die Vollständigkeit erhalten.

Für Implementierungen ist es häufig auch sinnvoll, die Zahl der Regelanwendungen auf  $\gamma$ -Formeln durch eine Grenze  $q$  zu beschränken. Auf dem selben Ast wird die  $\gamma$ -Regel dann höchstens  $q$  mal auf eine  $\gamma$ -Formel angewendet. Da die Vollständigkeit des Tableaurekalküls nur erhalten bleibt, wenn die Grenze  $q$  jeweils hinreichend hoch<sup>8</sup> gewählt wird, muß  $q$ , wenn zunächst kein Tableaubeweis gefunden wird, schrittweise erhöht werden.

**Definition 5.1.8** (Erschöpfter Ast, erschöpftes Tableau)

*Ein Ast  $A$  (bzw. ein Tableau  $T$ ) heie erschöpft, wenn  $A$  (bzw.  $T$ ) nicht erweitert werden kann, ohne gegen eine für die jeweils verwendete Version des Tableaurekalküls geltende Beschränkung zu verstoen, insbesondere*

- *gegen die verwendete Tableaurekonstruktionsregel,*
- *gegen die Einschränkung, da auf eine Formel, die keine  $\gamma$ -Formel ist, die entsprechende Tableauregel auf jedem Ast nur einmal angewendet werden darf,*
- *gegen eine Beschränkung der  $\gamma$ -Regelanwendungen durch eine Grenze  $q$ .*

Hier nun der zentrale Satz dieses Abschnittes, der die Korrektheit und Vollständigkeit des beschriebenen Kalküls zum Inhalt hat:

**Satz 5.1.9** (Korrektheit und Vollständigkeit des Tableaurekalküls mit freien Variablen)

*Sei  $\phi$  eine geschlossene Formel.*

*Korrektheit: Gibt es einen Tableaubeweis für  $\phi$ , dann ist  $\phi$  eine Tautologie.*

*Vollständigkeit: Ist  $\phi$  eine Tautologie, dann gibt es einen Tableaubeweis für  $\phi$ . Insbesondere gibt es auch einen nach einer beliebigen fairen Konstruktionsregel und unter Beachtung einer Grenze  $q$  für die Anwendung der  $\gamma$ -Regeln aufgebauten Tableaubeweis für  $\phi$ , vorausgesetzt die Grenze  $q$  ist hinreichend hoch gewählt.*

**Beweis:**

*Korrektheit:* Siehe [Hähnle & Schmitt, 1993].

*Vollständigkeit:* Siehe [Fitting, 1990]. ■

<sup>7</sup> Statt diese Einschränkung zu beachten, kann man auch Formeln (auer  $\gamma$ -Formeln), auf die eine Tableauregel angewendet wurde, von dem dadurch erweiterten Ast löschen (wobei man den Ast als Menge von Formeln betrachtet).

<sup>8</sup> Wann eine Grenze „hinreichend hoch“ ist, hängt von der zu beweisenden Formel ab und lät sich nicht im voraus bestimmen (das widerspräche der Unentscheidbarkeit der Prädikatenlogik).

Der Beweis der Korrektheit des Tableaurekalküls beruht darauf, daß die Anwendung einer Tableauregel auf ein Tableau seine Erfüllbarkeit erhält:

**Definition 5.1.10** (Erfüllbares Tableau)

*Ein Tableau  $T$  heie erfllbar, wenn es ein Modell  $\mathcal{M}$  gibt, so da  $\mathcal{M}$  fr jede Grundsubstitution  $\sigma$  Modell eines Astes  $A\sigma$  von  $T\sigma$  ist, d.h. Modell aller Formeln auf  $A\sigma$ .*

In [Fitting, 1990] ist die Erfllbarkeit eines Tableaus anders definiert, nmlich in der Weise, da zu jeder Grundsubstitution die Existenz einer Interpretation  $\mathcal{M}_\sigma$  gefordert wird, die Modell eines Astes von  $T\sigma$  ist. In Definition 5.1.10 dagegen wird die Existenz *einer* Interpretation  $\mathcal{M}$  gefordert, die die gewnschte Eigenschaft fr *alle* Grundsubstitutionen  $\sigma$  hat.

Die Erfllbarkeit nach Fitting kann bei der Anwendung der hier verwendeten *liberalisierten*  $\delta$ -Regel verloren gehen. Darum kann der Korrektheitsbeweis aus [Fitting, 1990] nicht bernommen werden. Die Erfllbarkeit nach Definition 5.1.10 dagegen bleibt erhalten.<sup>9</sup>

### 5.1.3 Universelle Formeln<sup>10</sup>

$\gamma$ -Formeln werden zum Abschlu eines Tableaus oft mehrfach bentigt, mit unterschiedlichen Instantiierungen fr die freien Variablen, die sie enthalten; besonders hufig gilt dies fr Gleichungen.

Ein typisches Beispiel ist das Assoziativittsaxiom

$$As = (\forall x)(\forall y)(\forall z)((x \cdot y) \cdot z \approx x \cdot (y \cdot z))$$

z.B. der Gruppentheorie. In der Regel mu es mehrfach angewendet werden, mit verschiedenen Instantiierungen fr  $x$ ,  $y$  und  $z$ , um selbst einfachste Theoreme der Gruppentheorie zu beweisen.

Normalerweise mu auf solche Formeln die  $\gamma$ -Regel mehrfach angewendet werden, um die notwendige Zahl von Instanzen zu erzeugen. Das hat mehrere Nachteile: Zum einen, mu eine Grenze  $q$  fr die Anwendung der  $\gamma$ -Regeln so hoch gewhlt sein, da gengend Instanzen erzeugt werden knnen. Zum anderen werden auch Kopien von  $\gamma$ -Formeln erzeugt, die gar nicht bentigt werden. Dadurch wird das erzeugte Tableau und damit der Suchraum nach einem Tableaubeweis aufgeblht.

Hufig ist es jedoch gar nicht notwendig die  $\gamma$ -Regel tatschlich mehrfach anzuwenden, nmlich dann, wenn ohnehin beliebig viele Kopien einer Formel erzeugt werden knnten, ohne da das zu Verzweigungen fhrte oder sonst irgendeinen Einflu auf den Rest des Tableaus htte. Solche Formeln, das Assoziativittsaxiom  $As$  ist ein Beispiel, werden als „universell“ bezeichnet:

<sup>9</sup> In [Beckert *et al.*, 1993] ist bewiesen, da dies auch fr die noch weiter liberalisierte  $\delta^{++}$ -Regel gilt.

<sup>10</sup> Eine ausfhrliche Darstellung des Konzeptes der universellen Formeln findet sich in [Beckert, 1991] und (krzer) in [Beckert & Hhnle, 1992].



**Definition 5.1.11** (Universelle Formel)

Sei  $\phi \in \mathbf{Form}^*$  eine Formel auf einem Ast  $A$  eines Tableaus, und sei  $\phi_u \in \mathbf{Form}$  die zu  $\phi$  äquivalente Formel ohne Vorzeichen, d. h., ist  $\phi = \top G$  für eine Formel  $G$ , dann sei  $\phi_u = G$ ; andernfalls, wenn  $\phi = \text{F } G$ , dann sei  $\phi_u = \neg G$ .

$\phi$  ist universell bezüglich einer Variablen  $x$ , wenn für jedes Modell  $\mathcal{M}$  und für jede Grundsubstitution  $\sigma$  gilt:

$$\text{Wenn } \mathcal{M} \models A\sigma, \text{ dann } \mathcal{M} \models ((\forall x)\phi_u)\sigma .$$

Im allgemeinen ist es unentscheidbar, ob eine Formel universell ist. Ein Teil der universellen Formeln kann jedoch sehr leicht erkannt werden (was unerlässlich ist, will man das Konzept der universellen Formeln für eine effiziente Implementierung ausnutzen).

**Definition 5.1.12** (Methode zum Auffinden universeller Formeln)

Eine Abbildung

$$\Upsilon : \{A : A \text{ der Ast eines Tableaus}\} \times \mathbf{Form}^* \longrightarrow 2^{\mathcal{V}}$$

heiße Methode zum Auffinden universeller Formeln, falls aus

$$x \in \Upsilon(A, \phi)$$

folgt, daß die signierte Formel  $\phi$  auf  $A$  liegt und universell ist bezüglich  $x$ .

Dieser Definition genügt beispielsweise die triviale Methode

$$\Upsilon_0(A, \phi) = \emptyset,$$

die überhaupt keine universelle Formel auffindet.

Leistungsfähiger aber immer noch hinreichend einfach, so daß sie in der Praxis einsetzbar ist, ist die folgende Methode:

**Satz 5.1.13**

Die Funktion  $\Upsilon$  ist eine Methode zum Auffinden universeller Formeln, die definiert ist durch:

$\Upsilon(A, \phi)$  enthalte genau die Variablen  $x$ , für die gilt:  $\phi$  liegt auf  $A$  und ist

1. mit Hilfe einer  $\gamma$ -Regel aus einer  $\gamma$ -Formel abgeleitet, und  $x$  ist die freie Variable, die dabei für die durch den Quantor gebundene Variable eingesetzt wurde, oder
2. mit Hilfe einer  $\alpha$ -Regel oder einer  $\gamma$ -Regel aus einer Formel abgeleitet, die universell ist bezüglich  $x$ .

**Beweis:**

Siehe [Beckert, 1991]. ■

Die im folgenden bewiesenen Sätze und die entwickelten Verfahren sind unabhängig von der tatsächlich gewählten Methode zum Auffinden universeller Formeln. In der Praxis ist aber die Methode aus Satz 5.1.13 die einzig sinnvolle (neben der trivialen Methode  $\Upsilon_0$ , mit der man auf die Verwendung universeller Formeln verzichtet).

Wie schon gesagt ist der Vorteil universeller Formeln — zumindest solcher, die man als universell erkannt hat —, daß sie mehrfach, mit verschiedenen Instantiierungen für die Variablen, bezüglich derer sie universell sind, verwendet werden können. Für den Tableaurekalkül ohne Gleichheit bedeutet das, daß verschiedene Äste mit verschiedenen Instanzen der selben universellen Formeln abgeschlossen werden können:

**Definition 5.1.14** (Abschluß mit universellen Formeln)

Sei  $T$  ein Tableau mit den Ästen  $A_1, \dots, A_k$  und  $\Upsilon$  eine Methode zum Auffinden universeller Formeln. Gibt es eine Grundsubstitution  $\sigma$ , so daß

1. es für  $1 \leq i \leq k$  Formeln  $\phi_i, \psi_i \in A_i$  gibt und eine Grundsubstitution  $\sigma_i$ , so daß  $\phi_i \sigma_i$  und  $\psi_i \sigma_i$  komplementär sind,
2. sich für  $1 \leq i \leq k$  die Substitution  $\sigma_i$  höchstens bei der Belegung solcher Variablen von  $\sigma$  unterscheidet, bezüglich derer  $\phi_i$  und  $\psi_i$  universell sind, d. h.

$$\sigma_i|_{\mathcal{V} \setminus (\Upsilon(A_i, \phi_i) \cap \Upsilon(A_i, \psi_i))} = \sigma|_{\mathcal{V} \setminus (\Upsilon(A_i, \phi_i) \cap \Upsilon(A_i, \psi_i))} ,$$

dann heiÙe  $T$  geschlossen (unter  $\sigma$  und mit  $\Upsilon$ ).

Wie in [Beckert, 1991] bewiesen,<sup>11</sup> ist jedes nach Definition 5.1.14 geschlossene Tableau unerfüllbar, woraus folgt, daß die Korrektheit des Tableaurekalküls durch die „Vereinfachung“ des Abschlusses nicht beeinträchtigt wird. Der Begriff „Vereinfachung“ ist hier tatsächlich angebracht: Zwar ist die Definition des Abschlusses komplizierter geworden, durch die Verwendung universeller Formeln gibt es aber im allgemeinen mehr Substitutionen, unter denen ein Tableau geschlossen ist. Diese sind außerdem auch einfacher zu finden.

Da jedes Tableau, das schon ohne universelle Formeln, d. h. nach Definition 5.1.5, geschlossen ist, erst recht auch geschlossen ist, wenn man universelle Formeln verwendet (Def. 5.1.14), wird dadurch die Vollständigkeit des Tableaurekalküls keinesfalls beeinträchtigt.

Wie die Universalität von Gleichungen bei der Erweiterung des Tableaurekalküls um Gleichheitsbehandlung ausgenutzt werden kann, ist in Abschnitt 5.2.3 beschrieben.

## 5.2 Der Tableaurekalkül mit Gleichheit

### 5.2.1 Einführung

Für den Tableaurekalkül mit Gleichheit, muß man zunächst zu allen Definitionen, die in irgendeiner Weise auf Modellen beruhen, eine Einschränkung auf normale

<sup>11</sup> In [Beckert, 1991] ist der Abschluß mit universellen Formeln zwar etwas anders formuliert — ohne Zuhilfenahme einer einzelnen Grundsubstitution  $\sigma$ . Die Äquivalenz beider Formulierungen ist aber unmittelbar einsichtig. Der Vorteil der Formulierung aus Definition 5.1.14 ist, daß mit ihr auch weiter, wie beim Tableaurekalkül ohne universelle Formeln, von einer (einzelnen) ein Tableau abschließenden Substitution gesprochen werden kann.

Modelle<sup>12</sup> hinzufügen — so ist beispielsweise ein Tableau  $T$  für die Prädikatenlogik mit Gleichheit nur dann erfüllbar, wenn es ein *normales* Modell  $\mathcal{M}$  gibt, das für jede Grundsubstitution  $\sigma$  Modell eines Astes  $A\sigma$  von  $T\sigma$  ist.

Welche Änderungen man vornehmen muß, um den Tableaukalkül um eine Gleichheitsbehandlung zu erweitern, wird deutlich, wenn man den Aufbau eines Tableaus für eine Formel  $\phi$  als Konstruktion der möglichen Modelle dieser Formel auffaßt. Dafür muß der Tableaukalkül folgendes leisten:

1. Formeln, die in einem Modell  $\mathcal{M}$  der Formel  $\phi$  gelten, müssen auf den mit  $\mathcal{M}$  korrespondierenden Ast gebracht werden,
2. Formeln oder Formelmengen, die in keinem Modell gelten können müssen als Abschluß eines Astes erkannt werden.

Diese beiden Anforderungen sind nicht unabhängig voneinander. Je genauer die Formeln auf einem Ast ein Modell  $\mathcal{M}$  darstellen, d. h. je mehr der in  $\mathcal{M}$  gültigen Formeln auf dem Ast vorhanden sind, je einfacher ist es Widersprüche zu erkennen.

In den kanonischen Modellen, die man für die Prädikatenlogik mit Gleichheit betrachtet, müssen nun zum einen mehr Formeln Gültigkeit haben als in einem beliebigen Modell der Prädikatenlogik (gelten beispielsweise in einem normalen Modell  $P(a)$  und  $a \approx b$ , so muß in ihm auch  $P(b)$  gelten). Zum anderen gibt es aber auch zusätzliche Widersprüche; so ist etwa  $\neg(a \approx a)$  in allen kanonischen Modell falsch.

Es gibt nun zwei Möglichkeiten, den Tableaukalkül um Gleichheitsbehandlung zu erweitern. Zum einen kann man die zusätzlich in kanonischen Modellen gültigen Formeln zu den Ästen eines Tableaus hinzufügen. Dazu muß man neue, auf der Ableitbarkeitsrelation  $\leftrightarrow_E$  beruhende Tableauregeln definieren. Es genügt dann, sehr einfache zusätzliche Abschlußregeln zu verwenden. Auf dieser naheliegenden Methode beruhende Verfahren sind im nächsten Abschnitt beschrieben.

Zum anderen kann man aber auch darauf verzichten, neue Tableauregeln einzuführen, und statt dessen eine aufwendigere Abschlußregel definieren, die auf der Lösung bestimmter aus den Formeln eines Astes gebildeter  $E$ -Unifikationsprobleme beruht.

### 5.2.2 Gleichheitsbehandlung durch zusätzliche Tableauregeln

Eine Ausführliche Diskussion der Vor- und Nachteile der verschiedenen auf der Verwendung zusätzlicher Tableauregeln beruhenden Verfahren, die auch Beispiele für ihre Anwendung zeigt, findet sich in [Beckert, 1991, Beckert & Hähnle, 1992]. Auch in [Fitting, 1990] wird auf die dabei auftretenden Probleme näher eingegangen.

Schon in den sechziger Jahren wurden Verfahren für die Gleichheitsbehandlung im Tableaukalkül ohne freie Variablen entwickelt [Jeffrey, 1967, Popplestone, 1967]. R. C. Jeffrey führte als zusätzliche Tableauerweiterungsregel ein, daß die Gleichungen auf einem Ast des Tableaus in beide Richtungen auf Terme des Astes angewendet werden dürfen, um neue Formeln zu generieren (Tabelle 5.3).

<sup>12</sup> Oder kanonische Modelle — was wie Satz 2.2.7 zeigt gleichbedeutend ist.

$\frac{\top (t \approx s) \quad \phi[t]}{\phi[s]}$	$\frac{\top (s \approx t) \quad \phi[s]}{\phi[t]}$
--	--

**Tabelle 5.3:** Zusätzliche Tableaureweiterungsregeln von R. C. Jeffrey zur Behandlung von Gleichheit im Tableaurekalkül ohne freie Variablen.

Neben dieser Erweiterungsregel gibt es eine zusätzliche Abschlußregel. Ein Ast ist nun auch geschlossen, wenn er eine Formel der Form  $\top (t \approx t)$  enthält (wobei  $t$  ein beliebiger Grundterm ist).

Der Nachteil dieses Verfahrens ist die Symmetrie der Erweiterungsregeln. Sie führt zu einem sehr großen Suchraum. Enthält nämlich beispielsweise ein Ast die Formeln  $f(a) \approx a$  und  $P(a)$ , so können die Formeln  $P(f(a))$ ,  $P(f(f(a)))$ ,  $P(f(f(f(a))))$ , ... zu dem Ast hinzugefügt werden.

Eine zu einem kleineren Suchraum führende zusätzliche Tableauregel wurde von S. Reeves [1987] eingeführt. Sie beruht auf der Tatsache, daß in einem kanonischen Modell, in dem Formeln  $\top P(a_1, \dots, a_n)$  und  $\top P(b_1, \dots, b_n)$  gelten, wenigstens eine der Formeln  $\top (a_1 \approx b_1)$ , ...,  $\top (a_n \approx b_n)$  gelten muß (Tabelle 5.4). Auch mit dieser Erweiterungsregel genügt es, als zusätzliche Abschlüsse Formeln der Form  $\top (t \approx t)$  zu verwenden. Diese Regel hat nun aber den Nachteil, daß es zu einer sehr starken, für eine Implementierung nicht mehr beherrschbaren Verzweigung des Tableaus führt.

$\frac{\top P(a_1, \dots, a_n) \quad \top P(b_1, \dots, b_n)}{\top (a_1 \approx b_1 \wedge \dots \wedge a_n \approx b_n)}$	$\frac{\top (f(a_1, \dots, a_n) \approx f(b_1, \dots, b_n))}{\top (a_1 \approx b_1 \wedge \dots \wedge a_n \approx b_n)}$
--	---

**Tabelle 5.4:** Zusätzliche Tableaureweiterungsregeln von S. Reeves zur Behandlung von Gleichheit im Tableaurekalkül ohne freie Variablen.

M. Fitting [1990] erweiterte Jeffreys Ansatz, so daß er für den Tableaurekalkül *mit freien Variablen* geeignet ist. Durch die Anwendung der Tableaureweiterungsregel mit freien Variablen (Tabelle 5.5) wird ein Tableau nicht nur erweitert, sondern insgesamt verändert. Soll nämlich eine Gleichung angewendet werden, und muß dafür eine freie Variable durch einen Term substituiert werden, so muß diese Substitution auf das ganze Tableau erstreckt werden. Auch beim Abschluß eines Astes kann nun eine Unifikation notwendig werden; so ist ein Ast mit der Formel  $\top (f(x) \approx f(a))$  geschlossen, nachdem die Substitution  $\{x/a\}$  (auf das ganze Tableau) angewendet worden ist.

Fitting führte eine weitere Neuerung ein: Bei der in [Fitting, 1990] beschriebenen Implementierung seines Verfahrens ist die Anwendung der Gleichungen nicht in die Anwendung der übrigen Tableaureweiterungsregeln integriert. Zuerst wird die Anwendung der anderen Regeln erschöpft (die Anwendung der  $\gamma$ -Regeln wird durch eine Grenze  $q$  beschränkt), und erst dann werden die Gleichungsregeln angewendet.

Nachdem ein Tableau erschöpft ist, müssen Gleichungen nur noch auf atomare Formeln angewendet werden. Dadurch wird

$\frac{\top (t \approx s)}{\phi[t']}$ $\frac{\phi[t']}{(\phi[s])\mu}$	$\frac{\top (t \approx s)}{\phi[s']}$ $\frac{\phi[s']}{(\phi[t])\mu}$
<p>wobei <math>\mu</math> ein MGU von <math>t</math> und <math>t'</math> ist, der auf das ganze Tableau angewendet wird.</p>	<p>wobei <math>\mu</math> ein MGU von <math>s</math> und <math>s'</math> ist, der auf das ganze Tableau angewendet wird.</p>

**Tabelle 5.5:** Zusätzliche Tableauerweiterungsregeln von M. Fitting für die Behandlung von Gleichheit im Tableaukalkül mit freien Variablen.

1. der Suchraum verkleinert, und die Zahl der möglichen Gleichungsanwendungen stark eingeschränkt,
2. es möglich, sich auf solche atomare Formeln zu beschränken, die potentiell komplementär sind also potentiell den Ast abschließen.

**Definition 5.2.1** (Potentiell komplementäre Atome)

Liegt auf einem Ast eines Tableaus ein Paar

$$\top P(s_1, \dots, s_n) \quad \text{und} \quad \text{F} P(t_1, \dots, t_n)$$

von Atomen mit dem gleichen Prädikatensymbol und verschiedenen Vorzeichen, so heißen diese Atome potentiell komplementär.

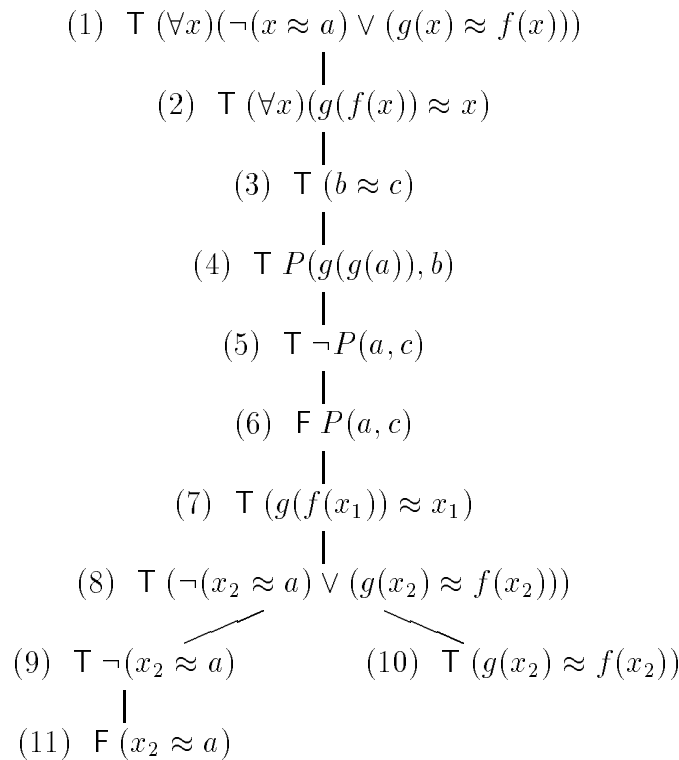
Zwar kann es auch von Nachteil sein, die Anwendung von Gleichungen aufzuschieben bis das Tableau erschöpft ist, nämlich dann, wenn ein Ast durch die Anwendung von Gleichungen auf komplexe Formeln geschlossen werden könnte, bevor durch die Anwendung von Tableauregeln auf diese komplexen Formeln der Ast verzweigt. In der Praxis sind diese Fälle aber selten. Auch ist die Anwendung von Gleichungen sehr viel aufwendiger als die der normalen Tableauerweiterungsregeln.

**Beispiel 5.2.2**

Abbildung 5.1 zeigt ein mit der Grenze  $q = 1$  für die Anwendung der  $\gamma$ -Regeln voll expandiertes Tableau für die Formeln

- (1)  $(\forall x)(\neg(x \approx a) \vee (g(x) \approx f(x)))$
- (2)  $(\forall x)(g(f(x)) \approx x)$
- (3)  $b \approx c$
- (4)  $P(g(g(a)), b)$
- (5)  $\neg P(a, c)$

Dieses Beispiel beinhaltet die wesentlichen Probleme bei der Suche nach einem Beweis in einem Tableaukalkül mit einer auf zusätzlichen Tableauregeln basierenden Gleichheitsbehandlung. In Abbildung 5.2 ist gezeigt, wie das Tableau durch Anwendung der Erweiterungsregeln von Fitting (Tabelle 5.5) geschlossen werden kann.



**Abbildung 5.1:** Das voll expandierte Tableau für die in Beispiel 5.2.2 angegebene Formelmeng e (vor der Erweiterung mit Hilfe von Gleichungen). Es ist folgendermaßen aufgebaut: Die Formeln (1) bis (5) sind vorgegeben; man erhält Formel (6) aus (5), (7) aus (2), (8) aus (1), (9) und (10) aus (8) und (11) aus (9).

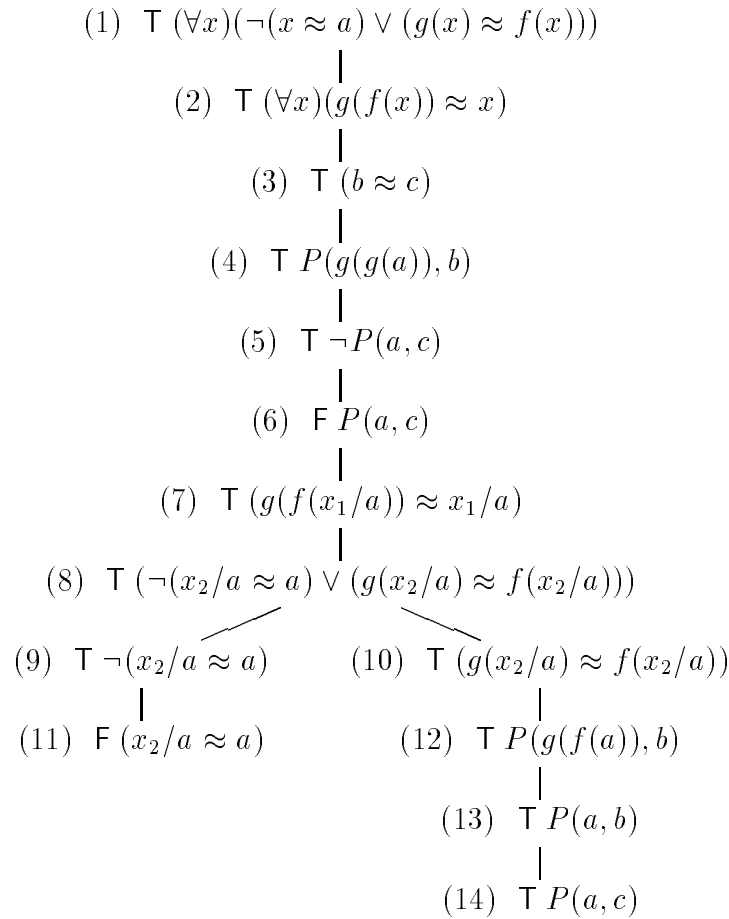
Die für die beiden Gleichungsanwendungen notwendigen Substitutionen  $\{x_1/a\}$  und  $\{x_2/a\}$  sind — wie in Abbildung 5.2 angedeutet<sup>13</sup> — auf das ganze Tableau anzuwenden. Der abgebildete Tableaubeweis ist minimal, d. h. die günstigste Reihenfolge von Gleichungsanwendungen wurde gewählt.  $\square$

### 5.2.3 Gleichheitsbehandlung durch $E$ -Unifikation

Die im vorangegangenen Abschnitt beschriebenen Verfahren kranken alle daran, daß Gleichungen relativ unbeschränkt angewendet werden können. Wegen der Symmetrie der Gleichheit führt das schon bei kleinen Problemen zu einem derart großen Suchraum, daß in einem vernünftigen Zeitrahmen kein Tableaubeweis gefunden werden kann.

Will man komplexere (und vor allem auch effizientere) Verfahren zur Gleichheitsbehandlung im Tableaulkalkül einsetzen, beispielsweise vervollständigungsbasierte,

<sup>13</sup>  $x_1/a$  bedeutet etwa, daß an dieser Stelle im Tableau zunächst die freie Variable  $x_1$  auftritt, diese aber beim weiteren Aufbau des Tableaus durch die Konstante  $a$  ersetzt wird, nämlich bei der Anwendung einer Substitution auf das ganze Tableau.



**Abbildung 5.2:** Der Abschluß des in Abbildung 5.1 dargestellten Tableaus mit Fittings Gleichheitsbehandlung für den Tableaalkül mit freien Variablen (Beispiel 5.2.2). Der linke Ast des Tableaus ist durch Formel (11) nach Anwendung der Substitution  $\{x_2/a\}$  geschlossen. Der rechte Ast wird durch Anwendung der Gleichung (10) auf (4) um Formel (12) erweitert, durch Anwendung der Gleichung (7) auf (12) um Formel (13) — dazu muß die Substitution  $\{x_1/a\}$  ausgeführt werden — und durch Anwendung der Gleichung (3) auf (13) um Formel (14). Die Formel (14) bildet schließlich zusammen mit der Formel (6) den Abschluß des rechten Astes.

stellt man schnell fest, daß es schwierig ist, diese Verfahren in zusätzliche Tableauregeln zu kleiden.

Wie jedoch in [Beckert, 1991] gezeigt wurde, ist es möglich, die Gleichheitsbehandlung im Tableaurekalkül auf die Lösung simultaner gemischter  $E$ -Unifikationsprobleme zu reduzieren; nämlich dadurch, daß man ein Tableau als geschlossen definiert, wenn bestimmte simultane gemischte  $E$ -Unifikationsprobleme eine Lösung besitzen.<sup>14</sup> Zusätzliche Tableauregeln werden dann nicht benötigt. Die gesamte Gleichheitsbehandlung wird in den Abschluß der Äste verlagert.

Die Reduktion auf  $E$ -Unifikationsprobleme gestattet es, beliebige Methoden zu deren Lösung zu verwenden, ohne die Besonderheiten des Tableaurekalküls berücksichtigen zu müssen. In [Beckert, 1991] ist gezeigt, daß die Gleichheitsbehandlung durch Reduktion auf  $E$ -Unifikationsprobleme sehr viel effizienter als die im vorigen Abschnitt beschriebenen auf zusätzlichen Tableauregeln basierenden Verfahren ist, und zwar selbst dann, wenn man das vergleichsweise ineffiziente Verfahren aus [Beckert, 1991] (siehe Abschnitt 3.5) zur Lösung der auftretenden  $E$ -Unifikationsprobleme verwendet.

Die Gleichungsmengen der zu lösenden  $E$ -Unifikationsprobleme enthalten die Gleichungen auf dem jeweils zu schließenden Ast, und als zu unifizierende Terme verwendet man potentiell komplementäre Atome. Kann man ein solches  $E$ -Unifikationsproblem lösen, hat man damit gezeigt, daß es eine Substitution (den Unifikator) gibt, der in jedem Ast ein Paar potentiell komplementärer Atome tatsächlich komplementär macht.

Die auftretenden Unifikationsprobleme sind rein starr oder, wenn man universelle Formeln verwendet, gemischt — jedenfalls aber nicht rein universell. Darum können Verfahren wie der UKBA zur Gleichheitsbehandlung im Tableaurekalkül mit freien Variablen nicht eingesetzt werden.<sup>15</sup>

Die durch den Ast  $A$  eines Tableaus definierte Gleichungstheorie besteht aus den auf  $A$  liegenden Gleichungen, die bezüglich der Variablen all-quantifiziert sind, bezüglich derer sie als universell erkannt werden.

**Definition 5.2.3** (Menge der Gleichungen eines Astes)

Sei  $A$  der Ast eines Tableaus und  $\Upsilon$  eine Methode zum Auffinden universeller Formeln. Dann ist

$$E(A) = \{(\forall x_1) \dots (\forall x_n)(s \approx t) : \top(s \approx t) \in A, \Upsilon(A, \top(s \approx t)) = \{x_1, \dots, x_n\}\}$$

die Menge der Gleichungen des Astes  $A$ .

#### Beispiel 5.2.4

Als Beispiel diene wieder das voll expandierte Tableau aus Abbildung 5.1. Sein linker Ast sei mit  $A_1$  bezeichnet und sein rechter mit  $A_2$ . Zum Auffinden universeller

<sup>14</sup> In [Beckert, 1991] wird der Begriff  $E$ -Unifikation nicht verwendet. Wie man aber leicht feststellt, entsprechen die dort beschriebenen Strukturen gemischten  $E$ -Unifikationsproblemen.

<sup>15</sup> Verzichtet man jedoch auf die Verwendung freier Variablen, enthalten die auftretenden Unifikationsprobleme keine Variablen. Dann sind die klassischen Vervollständigungsverfahren anwendbar. Tatsächlich hat R. J. Browne [1988] ein vervollständigungsbasiertes Verfahren für den Tableaurekalkül ohne freie Variablen formuliert; es ist allerdings mit Hilfe zusätzlicher Tableauregeln formuliert.



Formeln werde die Methode aus Satz 5.1.13 verwendet. Dann enthält  $E(A_1)$  die Gleichungen

$$b \approx c \quad \text{und} \quad (\forall x)(g(f(x)) \approx x) .$$

$E(A_2) \supset E(A_1)$  enthält zusätzlich die Gleichung

$$g(x_2) \approx f(x_2) .$$

□

Die Menge der Unifikationsprobleme  $\mathcal{P}(A)$  eines Astes  $A$  enthält die Unifikationsprobleme, von denen man eines lösen muß, um den Ast abzuschließen.

**Definition 5.2.5** (Menge der Unifikationsprobleme eines Astes)

*Sei  $A$  der Ast eines Tableaus  $T$  und  $\Upsilon$  eine Methode zum Auffinden universeller Formeln. Dann ist die Menge der Unifikationsprobleme  $\mathcal{P}(A)$  von  $A$  die kleinste Menge von Mengen von Term-Paaren, die die folgenden Bedingungen erfüllt (also genau die im folgenden beschriebenen Elemente enthält):*

- *Seien  $\top P(s_1, \dots, s_n)$  und  $\text{F} P(t_1, \dots, t_n)$  ein Paar potentiell komplementärer Atome auf  $A$ , wobei  $P \neq \approx$ .<sup>16</sup>*

$$\{x_1, \dots, x_m\} = \Upsilon(A, \top P(s_1, \dots, s_n)) \cap \Upsilon(A, \text{F} P(t_1, \dots, t_n))$$

*sei die Menge der Variablen bezüglich derer beide Atome als universell erkannt werden. Die Substitution*

$$\sigma = \{x_1/y_1, \dots, x_m/y_m\}$$

*benenne die Variablen  $x_1, \dots, x_m$  um, d. h.,  $y_1, \dots, y_m$  sind Variablen, die im Tableau  $T$  und auch in  $\mathcal{P}(A)$  sonst nicht auftreten. Dann enthält  $\mathcal{P}(A)$  das Element*

$$\{\langle s_1\sigma, t_1\sigma \rangle, \dots, \langle s_n\sigma, t_n\sigma \rangle\} .^{17}$$

- *Sei  $\text{F}(s \approx t)$  eine Ungleichung auf  $A$ .*

$$\{x_1, \dots, x_m\} = \Upsilon(A, \text{F}(s \approx t))$$

*sei die Menge der Variablen bezüglich derer die Ungleichung als universell erkannt wird. Die Substitution*

$$\sigma = \{x_1/y_1, \dots, x_m/y_m\}$$

*benenne die Variablen  $x_1, \dots, x_m$  in neue Variablen um. Dann enthält  $\mathcal{P}(A)$  das Element*

$$\{\langle s\sigma, t\sigma \rangle\} .^{18}$$

<sup>16</sup> Paare von Gleichungen und Ungleichungen sind also ausgeschlossen.

<sup>17</sup> Die Wahl der Variablen  $y_1, \dots, y_m$  bietet einen Freiheitsgrad, der es erlaubt, viele verschiedene solcher Elemente zu bilden. Dieser soll aber nicht ausgenutzt werden; die Wahl der  $y_1, \dots, y_m$  soll eindeutig sein, und für jedes Paar potentiell komplementärer Atome nur ein entsprechendes Element in  $\mathcal{P}(A)$  enthalten sein.

<sup>18</sup> Das in Fußnote 17 gesagte gilt entsprechend: Auch für jede Ungleichung soll nur ein Element in  $\mathcal{P}(A)$  enthalten sein.

**Beispiel 5.2.6**

Seien wieder  $A_1$  der linke und  $A_2$  der rechte Ast des Tableaus in Abbildung 5.1. Dann ist

$$\begin{aligned}\mathcal{P}(A_1) &= \{\{\langle x_2, a \rangle\}, \\ &\quad \{\langle g(g(a)), a \rangle, \langle b, c \rangle\}\} \\ \mathcal{P}(A_2) &= \{\{\langle g(g(a)), a \rangle, \langle b, c \rangle\}\}\end{aligned}$$

□

Die Mengen der Unifikationsprobleme und der Gleichungen eines Astes hängen von der verwendeten Methode zum Auffinden universeller Formeln ab. Man kann beliebige Methoden verwenden. Damit aber im folgenden die Mengen der Unifikationsprobleme und der Gleichungen eines Astes eindeutig bestimmt sind, sei nun eine Methode fest gewählt.

**Definition 5.2.7** (Abschluß eines Tableaus mit freien Variablen, universellen Formeln und Gleichheit)

Ein Tableau  $T$  mit den Ästen  $A_1, \dots, A_k$  ist geschlossen, wenn es in den Mengen  $\mathcal{P}(A_i)$  ( $1 \leq i \leq k$ ) der Unifikationsprobleme der Äste Elemente

$$\{\langle s_{i1}, t_{i1} \rangle, \dots, \langle s_{in_i}, t_{in_i} \rangle\} \in \mathcal{P}(A_i) \quad (1 \leq i \leq k)$$

gibt, so daß das simultane gemischte  $E$ -Unifikationsproblem

$$\begin{aligned}\{\langle E(A_1), s_{11}, t_{11} \rangle, \dots, \langle E(A_1), s_{1n_1}, t_{1n_1} \rangle, \\ \dots, \dots, \dots, \\ \langle E(A_k), s_{k1}, t_{k1} \rangle, \dots, \langle E(A_k), s_{kn_k}, t_{kn_k} \rangle\}\end{aligned}$$

eine Lösung besitzt.

**Beispiel 5.2.8**

Verwendet man die Methode aus Satz 5.1.13 zum Auffinden universeller Variablen, dann ist das Tableau aus Abbildung 5.1 gemäß Definition 5.2.7 geschlossen.

Die Substitution  $\sigma = \{x_2/a\}$  ist nämlich ein Unifikator des simultanen gemischten  $E$ -Unifikationsproblems

$$\begin{aligned}\{\langle E(A_1), x_2, a \rangle, \\ \langle E(A_2), g(g(a)), a \rangle, \\ \langle E(A_2), b, c \rangle\}\end{aligned}$$

□

In [Beckert, 1991] ist bewiesen, daß diese Definition des Abschlusses eines Tableaus zu einem korrekten und Vollständigen Kalkül führt:

**Satz 5.2.9** (Korrektheit und Vollständigkeit)

Sei  $\phi$  eine geschlossene Formel. Verwendet man Definition 5.2.7 für den Abschluß eines Tableaus,<sup>19</sup> dann gilt:

<sup>19</sup> Die Definition des Abschlusses eines Tableaus spielt eine wesentliche Rolle für die Frage, ob ein Tableau ein Tableaubeweis ist.

*Korrektheit:* Gibt es einen Tableaubeweis für  $\phi$ , dann ist  $\phi$  eine Tautologie der Prädikatenlogik mit Gleichheit.

*Vollständigkeit:* Ist  $\phi$  eine Tautologie der Prädikatenlogik mit Gleichheit, dann gibt es einen Tableaubeweis für  $\phi$ . Insbesondere gibt es auch einen nach einer beliebigen fairen Konstruktionsregel und unter Beachtung einer Grenze  $q$  für die Anwendung der  $\gamma$ -Regeln aufgebauten Tableaubeweis für  $\phi$ , vorausgesetzt die Grenze  $q$  ist hinreichend hoch gewählt.

**Beweis:**

Siehe [Beckert, 1991].<sup>20</sup> ■

Ein Charakteristikum der auftretenden  $E$ -Unifikationsprobleme ist, daß mehrere (viele) der Probleme die selbe Gleichungsmenge  $E$  enthalten. Das Verfahren aus Kapitel 4 ist für solche Probleme besonders geeignet, denn die zu unifizierenden Terme gehen nicht in die Vervollständigung  $\mathcal{R}^\infty$  von  $E$  ein. Sie kann darum für alle Probleme, die  $E$  enthalten verwendet werden.

Auch muß bei dem Abschluß gemäß Definition 5.2.7 für jeden Ast ein Unifikationsproblem ausgewählt werden, das Teil des simultanen Problems wird. Verwendete man ein völlig beliebiges Verfahren zur Lösung simultaner  $E$ -Unifikationsprobleme, könnte das bedeuten, daß man versuchen müßte alle simultanen Probleme, die sich als Kombination dieser Einzelprobleme bilden lassen, zu lösen. Das wäre aber völlig unpraktikabel. Bei dem Verfahren aus Kapitel 4 tritt dieses Problem nicht auf, weil zur Lösung simultaner Probleme zunächst die Einzelprobleme gelöst werden, aus denen sie bestehen.

Die Aufteilung der Paare  $\top P(s_1, \dots, s_n)$  und  $\text{F } P(t_1, \dots, t_n)$  potentiell komplementärer Atome in mehrere Unifikationsprobleme  $\langle s_1, t_1 \rangle, \dots, \langle s_n, t_n \rangle$  ist eigentlich unnötig; man könnte auch versuchen, die Paare als ganze zu unifizieren, also das Problem  $\langle P(s_1, \dots, s_n), P(t_1, \dots, t_n) \rangle$  zu lösen. Das wäre aber ineffizient, weil die Unifikation der Einzelterme unabhängig voneinander ausgeführt werden kann.

### 5.2.4 Sukzessiver Abschluß der Äste

Mit Hilfe des Verfahrens für die Lösung simultaner gemischter  $E$ -Unifikationsprobleme wie es in Abschnitt 4.10 beschrieben ist — es ist nach Satz 4.10.5 vollständig und liefert also ein Semi-Entscheidungsverfahren für die Existenz von Unifikatoren —, kann man überprüfen, ob ein Tableau gemäß Definition 5.2.7 geschlossen ist. Damit schließt man ein Tableau sozusagen „auf einen Schlag“ sobald man einen Unifikator findet.

Statt dessen kann man aber die Äste eines Tableaus auch nacheinander betrachten und abschließen. Man wählt dazu jeweils die erste Substitution, die einen Ast  $A_i$  abschließt und wendet sie auf das ganze Tableau an. Kann man danach einen anderen Ast  $A_j$  nicht abschließen, führt man Backtracking aus, und sucht einen weiteren Abschluß des Astes  $A_i$ .

<sup>20</sup> Wie gesagt (Fußnote 14 auf Seite 114), werden in [Beckert, 1991] andere Begriffe verwendet. Das gilt auch für diesen Beweis. Abgesehen von den Bezeichnungen kann er aber unverändert übernommen werden.

In [Beckert, 1991] ist bewiesen, daß auch dieses Verfahren korrekt und vollständig ist — vollständig aber nur, wenn man eine Grenze  $r$  einführt, die die Suche nach einem (weiteren) Abschluß eines Astes beschränkt.<sup>21</sup> Diese Grenze  $r$  muß dann, damit das Verfahren vollständig bleibt, ähnlich wie die Grenze  $q$  für die Anwendung der  $\gamma$ -Regeln, schrittweise erhöht werden, wenn kein Beweis gefunden wird.

### 5.2.5 Übertragung auf andere Versionen des Tableauekalküls

Die Ergebnisse und beschriebenen Verfahren lassen sich auf alle anderen bekannten Versionen des Tableauekalküls übertragen — zumindest auf diejenigen für die Prädikatenlogik mit Gleichheit. Sie alle können als Variante oder sogar Spezialfall des hier verwendeten Kalküls angesehen werden: Kalküle mit nicht-signierten Formeln, solche ohne freie Variablen oder ohne universelle Formeln, mit anderen  $\delta$ -Regeln (insbesondere der nicht liberalisierten), Kalküle mit Verfahren zur Einschränkung des Suchraums<sup>22</sup>, mit Lemmagenerierung usw.

Schwierigkeiten können allenfalls bei Tableauekalkülen für andere Logiken auftreten, beispielsweise wenn die Terme mit Sorten versehen sind, denn dies hat direkte Auswirkungen auf den Begriff der Gleichheit von Termen und damit die Gleichheitsbehandlung.

Je nachdem welchen Tableauekalkül man verwendet, können einfachere Versionen der  $E$ -Unifikation als die gemischte zur Gleichheitsbehandlung ausreichen. Beispielsweise treten bei der Gleichheitsbehandlung im Tableauekalkül ohne universelle Formeln nur rein starre Unifikationsprobleme auf.<sup>23</sup>

---

<sup>21</sup> Bei der Implementierung ist die Zahl der Transformations- und Normalisierungsschritte und die Zahl der jeweils erzeugten Abschlüsse beschränkt (siehe Abschnitt 6.1)

<sup>22</sup> Solche Verfahren sind beispielsweise die Beachtung von *Links* zwischen Formeln und die Einhaltung von Regularitätsbedingungen.

<sup>23</sup> Die Frage, ob ein Ast eines Tableaus ohne universelle Formeln geschlossen ist, ist also entscheidbar — was allerdings nicht von großer Bedeutung ist, da die Frage, ob der Ast so erweitert werden kann, daß die Erweiterung geschlossen ist, unentscheidbar bleibt.

# 6 Implementierung

---

## 6.1 Implementierung als Teil des Tableaubeweiser $\mathcal{I}^A\mathcal{P}$

Das Verfahren zur vervollständigungs-basierten Gleichheitsbehandlung im Tableaurekalkül, wie es in den Kapiteln 4 und 5 beschrieben ist, ist als neues Modul des Tableaubeweisers  $\mathcal{I}^A\mathcal{P}$  [Hähnle *et al.*, 1992] implementiert worden. Es ersetzt das bisher verwendete Modul zur Gleichheitsbehandlung, das eine Implementierung des äquivalenzklassenbasierten Verfahrens aus [Beckert & Hähnle, 1992] ist.

In seiner zweiwertigen<sup>1</sup> Version stellt  $\mathcal{I}^A\mathcal{P}$  — schaltet man Zusatzfunktionen wie Lemma-Generierung u. ä. aus — eine Implementierung des in Kapitel 5 beschriebenen Tableaurekalküls mit freien Variablen, universellen Formeln und der liberalisierten  $\delta$ -Regel dar.

$\mathcal{I}^A\mathcal{P}$  generiert und schließt die Äste eines Tableaus nacheinander. Darum kann auch zum Abschluß mit Gleichheit immer nur ein Ast betrachtet werden; implementiert ist also das Verfahren aus Abschnitt 5.2.4, bei dem die Äste sukzessive geschlossen werden. Wie dort ausgeführt, ist diese Methode nur vollständig, wenn sowohl die Suche nach Unifikatoren, mit denen ein Ast geschlossen ist, begrenzt wird, als auch die Zahl von Abschlüssen, die für einen Ast betrachtet werden. Bei der Implementierung wird dies durch zwei Grenzen  $r$  für die Zahl der Transformationen und Normalisierungen und  $r'$  für die Zahl generierter Unifikatoren erreicht.<sup>2</sup>

Die Schnittstelle zwischen dem Modul zur Gleichheitsbehandlung und den restlichen Modulen konnte unverändert bleiben. An dieser Schnittstelle wird dem Gleichheits-Modul die Menge der Gleichungen  $E(A)$  eines schon voll expandierten, abzuschließenden Astes  $A$  (Def. 5.2.3) und die Menge der Probleme  $\mathcal{P}(A)$  (Def. 5.2.5) übergeben.  $\mathcal{I}^A\mathcal{P}$  verwendet die Methode aus Satz 5.1.13 zum Auffinden universeller Formeln, die Gleichungen in  $E(A)$  können also universell sein bezüglich Variablen, die sie enthalten.

Das Gleichheits-Modul berechnet die Menge  $\vec{U}$  (Satz 4.10.3) für die simultanen gemischten  $E$ -Unifikationsprobleme des Astes  $A$ , die eine bezüglich  $\leq$  grund-vollständige Menge von Unifikatoren ist. Die Unifikatoren in dieser Menge werden benutzt, um den Ast gemäß Definition 5.2.7 abzuschließen.

Die Implementierung ist wie der Rest des Beweisers  $\mathcal{I}^A\mathcal{P}$  in Quintus Prolog, Version 3.1, geschrieben; sie umfaßt etwa 2.500 Zeilen bzw. 85KB Code.

---

<sup>1</sup> Der Beweiser  $\mathcal{I}^A\mathcal{P}$  ist für beliebige mehrwertige Logiken geeignet.

<sup>2</sup> Allerdings werden diese Grenzen nicht — wie es für die Vollständigkeit eigentlich notwendig wäre — automatisch erhöht, wenn kein Beweis gefunden wurde, sondern es bleibt dem Benutzer überlassen, höhere Werte für  $r$  und  $r'$  zu wählen. Standardmäßig sind beide Grenzen so hoch gesetzt, daß sie praktisch nicht erreicht werden können.

## 6.2 Überprüfen der Erfüllbarkeit von Constraints

Verwendet man eine lexikographische Pfadordnung  $\succ_x = \succ_{\text{LPO}}$  (Def. 3.6.7) zur Interpretation des Symbols  $\succ$  in den Ordnungsbedingungen, ist die Konsistenz eines Constraints  $c$  entscheidbar, und die  $O$  erfüllenden Substitutionen sind aufzählbar (Satz 4.2.14). Wie H. Comon gezeigt hat, bleibt die Entscheidbarkeit erhalten, wenn man auch  $\approx$  als Prädikatensymbol in Ordnungsbedingungen zuläßt — was aber, wie schon in Abschnitt 4.2 gesagt, keine wesentlichen Vorteile hätte.

Prinzipiell würde es für die Implementierung des Verfahrens zur gemischten  $E$ -Unifikation aus Kapitel 4 genügen, den in [Comon, 1990] beschriebenen Entscheidungsalgorithmus zu implementieren. Dieser ist allerdings höchst komplex — was nicht anders zu erwarten ist, da es ein NP-hartes Problem ist, zu entscheiden, ob ein Constraint konsistent ist. Man muß nämlich auch solche Inkonsistenzen berücksichtigen und aufdecken, die beispielsweise dadurch entstehen, daß die Ordnungsbedingung  $(b \succ x) \wedge (x \succ a)$  unerfüllbar ist, wenn es kein Konstantensymbol gibt, das — bezüglich der die LPO induzierenden Ordnung  $>_F$  auf den Funktions- und Konstantensymbolen — zwischen  $a$  und  $b$  liegt.

Weil die Konsistenz von Constraints sehr oft überprüft werden muß, nämlich immer dann, wenn man mit Hilfe von Satz 4.2.12 testen will, ob ein Constraint einen anderen subsumiert, ist Comons Verfahren für die Implementierung zu ineffizient und ungeeignet.

Besser ist der folgende pragmatische Ansatz. Er basiert auf der Annahme, daß es kein wirklicher Schaden sein kann, wenn man die Signatur durch Hinzunahme endlich vieler Konstantensymbole ändert. Der Vorteil einer solchen Erweiterung der Signatur ist, daß man ein hinzugenommenes (bzw. hinzugedachtes) Konstantensymbol als — beispielsweise — zwischen  $a$  und  $b$  liegend definieren kann. Für die Annahme, daß dies unschädlich sei, spricht, daß das Verfahren aus Kapitel 4 korrekt und vollständig bleibt, wenn man Inkonsistenzen wie  $(b \succ x) \wedge (x \succ a)$  ignoriert.

Es ist vergleichsweise einfach, zu entscheiden, ob eine Ordnungsbedingung  $O$  oder, darauf aufbauend, ein Constraint  $\langle \sigma, O \rangle$  schwach konsistent ist, konsistent nämlich zumindest dann, wenn man die Signatur in geeigneter Weise ändert. Dazu formt man  $O$  in eine Menge äquivalenter „einfacher“ Ordnungsbedingungen um.<sup>3</sup>

### Definition 6.2.1 (Einfache Ordnungsbedingung)

Eine Ordnungsbedingung  $O$  heiße einfach, wenn sie gleich true oder false ist oder von der Form

$$(s_1 \succ t_1) \wedge \dots \wedge (s_n \succ t_n) \quad (n \geq 1) ,$$

und — faßt man  $O$  als (implizit konjunktiv verknüpfte) Menge auf — für alle  $(s \succ t) \in O$  gilt:

1.  $s \succ t$  ist weder wahr noch falsch.
2.  $s \in \mathcal{V}$  oder  $t \in \mathcal{V}$ .

<sup>3</sup> In etwa entsprechen die einfachen Ordnungsbedingungen der in [Comon, 1990] *solved form* genannten Normalform von Ordnungsbedingungen.

3. Es gibt kein  $(s \succ u) \in O$ , so daß  $u = u[t]$ .
4. Es gibt kein  $(u \succ t) \in O$ , so daß  $s = s[u]$ .
5. Falls  $(u \succ v[s]) \in O$ , dann auch  $(u \succ v[t]) \in O$ .

Eine einfache Ordnungsbedingungen ist also — sofern sie nicht gleich true oder false ist — auf atomarer Ebene konsistent (Punkt 1); je eine Seite ihrer Atome ist eine Variable (Punkt 2); sie enthält keine durch die Montonie von  $\succ$  bezüglich der Termstruktur entstehenden Inkonsistenzen (Punkte 3 und 4); und sie ist in gewissem Sinne transitiv vollständig (Punkt 5).

**Definition 6.2.2** (Schwach konsistente Ordnungsbedingung)

Ein Ordnungsbedingung heie schwach konsistent, wenn sie gem Definition 4.2.4 konsistent ist, nachdem die verwendete Signatur  $\mathcal{S}$  und die zur Interpretation des Symbols  $\succ$  in den Ordnungsbedingungen verwendete LPO  $\succ_{\text{LPO}}$  dadurch verndert werden, da

- endlich viele neue Konstantensymbole zu  $\mathcal{S}$  hinzugefgt werden,
- $\succ_{\text{LPO}}$  geeignet erweitert wird.

### Beispiel 6.2.3

$a$  und  $b$  seien die einzigen Konstantensymbole; es gelte  $b \succ_{\text{LPO}} a$ ; und  $f$  sei ein minimales Funktionssymbol bezglich  $\succ_{\text{LPO}}$ .

Dann sind die folgenden Ordnungsbedingungen schwach konsistent aber nicht konsistent:

- $(b \succ x) \wedge (x \succ a)$ ,
- $(f(a) \succ x) \wedge (x \succ b)$ ,
- $(a \succ x)$ .

Weder konsistent noch schwach konsistent sind

- $x \succ x$ ,
- $(x \succ y) \wedge (y \succ z) \wedge (z \succ x)$ ,
- $x \succ f(x)$ .

□

Eine einfache Ordnungsbedingung ist genau dann schwach konsistent, wenn sie ungleich false ist. Darum gengt es, will man die schwache Konsistenz einer beliebigen Ordnungsbedingung berprfen, diese in eine quivalente einfache Ordnungsbedingung umzuformen — oder, falls eine solche nicht existiert, in eine Disjunktion einfacher Ordnungsbedingungen (Lemma 6.2.4).

Neben der Möglichkeit, ihre Konsistenz in sehr einfacher Weise zu entscheiden, sind die einfachen Ordnungsbedingungen auch sonst sehr viel leichter handhabbar als beliebige Ordnungsbedingungen. Insbesondere lassen sich auch die Negation und die Konjunktionen einfacher Ordnungsbedingungen leicht wieder in eine äquivalente Menge einfacher Ordnungsbedingungen umformen.

Man darf aber natürlich nicht verschweigen, daß eine (wenn auch nur gedachte) Änderung der Signatur sehr wohl ungünstige Auswirkungen hat, insbesondere auf die Effizienz des Verfahrens — sieht man von dem verringerten Aufwand für die Überprüfung der Konsistenz von Ordnungsbedingungen ab. Erkennt man eine Inkonsistenz nicht, weil man sich eine Konstante hinzugedacht hat, kann eine Vervollständigung wenn auch nicht falsche so doch unnötige Reduktionsregeln enthalten. Auch Terme (und Normalformen von Termen), die eigentlich gelöscht werden könnten, bleiben unter Umständen erhalten, weil man nicht merkt, daß ihr Constraint inkonsistent ist oder sie subsumiert werden.

In den Beweisen der Korrektheit und der Vollständigkeit des Verfahrens wird keinerlei Bezug auf die verwendete Signatur genommen. Darum genügt der Test auf schwache Konsistenz, um „schlechte“ Inkonsistenzen aufzudecken, nämlich solche, die, wenn man sie übersieht, die Korrektheit oder die Vollständigkeit beeinträchtigen würden.

Nicht zu jeder Ordnungsbedingung existiert eine äquivalente einfache Ordnungsbedingung — beispielsweise nicht zu  $(x \succ a) \vee (x \succ b)$ . Es ist aber möglich, zu jedem *Constraint*  $c$  eine (implizit disjunktive verknüpfte) äquivalente Menge  $C$  von Constraints anzugeben, die ausschließlich einfache Ordnungsbedingungen enthalten. Mit diesen Mengen  $C$  können die Äquivalenzumformungsregeln (Def. 4.4.4) und 4.5.2) angewendet werden, so daß die Constraints der Reduktionsregeln einer Vervollständigung und der Terme einer Menge von Normalformen nur einfache Ordnungsbedingungen enthalten.

#### Lemma 6.2.4

*Sei  $c$  ein Constraint. Dann gibt es eine Menge*

$$C = \{\langle \sigma_1, O_1 \rangle, \dots, \langle \sigma_m, O_m \rangle\} \quad (m \geq 0)$$

*von Constraints, so daß*

$$\text{Sat}(C) = \text{Sat}(c) \text{ ,}$$

*und die Ordnungsbedingungen  $O_i$  ( $1 \leq i \leq m$ ) einfach sind.*

Eine — gemäß Lemma 6.2.4 zu jedem Constraint  $c$  existierende — äquivalente Menge von Constraints  $C$  mit einfachen Ordnungsbedingungen kann in folgender Weise aus  $c = \langle \sigma, O \rangle$  berechnet werden:<sup>4</sup>

1. Zunächst wird die Ordnungsbedingung  $O$  durch logische Umformungen auf disjunktive Normalform (DNF) gebracht,<sup>5</sup> d. h. auf die Form

$$(O_{1,1} \wedge \dots \wedge O_{1,l_1}) \vee \dots \vee (O_{k,1} \wedge \dots \wedge O_{k,l_k}) \quad (k, l_1, \dots, l_k \geq 1) \text{ ,}$$

<sup>4</sup> Dieses Verfahren zur Berechnung kann zugleich auch als konstruktiver Beweis für Lemma 6.2.4 aufgefaßt werden.

<sup>5</sup> Da nur solche Ordnungsbedingungen auftreten (s. u.), die Negationen, Konjunktionen oder Instanzen einfacher Ordnungsbedingungen sind, genügen wenige Umformungsschritte.



wobei die  $O_{ij}$  atomare Ordnungsbedingungen sind.

2. Damit kann die zu  $c$  äquivalente Menge

$$\begin{aligned} C' &= \{c'_1, \dots, c'_k\} \\ &= \{\langle \sigma, (O_{1,1} \wedge \dots \wedge O_{1,l_1}) \rangle, \dots, \langle \sigma, (O_{k,1} \wedge \dots \wedge O_{k,l_k}) \rangle\} \end{aligned}$$

erstellt werden.

3. Die nur noch  $\wedge$  als logischen Operator enthaltenden Constraints  $c'_i$  ( $1 \leq i \leq k$ ) können in Constraints mit einfachen Ordnungsbedingungen umgeformt werden, indem man Punkt 5 in der Definition einfacher Ordnungsbedingungen (Def. 6.2.1) und die Punkte 1 bis 3 in der Definition der lexikographischen Pfadordnungen (Def. 3.6.7) als Algorithmus (in Form eines logischen Programms) auffaßt. Die Umformung terminiert, wenn entweder eine einfache Ordnungsbedingung erreicht wird, oder wenn eine offensichtliche Inkonsistenz auftritt, d. h. ein Verstoß gegen einen der Punkte 1 bis 4 in der Definition einfacher Ordnungsbedingungen, oder eine atomare Ordnungsbedingungen auftritt, die inkonsistent ist.

Alle auftretenden Constraints werden immer sofort in dieser Weise umgeformt, d. h. nach jeder Anwendung einer Transformations- oder Normalisierungsregel werden die äquivalenten Mengen einfacher Constraints zu den Constraints der neuen Reduktionsregeln und Terme berechnet, die Äquivalenzumformungsregel angewendet, und gegebenenfalls Subsumtionen und Löschungen ausgeführt. Dadurch treten niemals komplexere Ordnungsbedingungen auf als solche, die Negationen, Konjunktionen oder Instanzen von Ordnungsbedingungen sind.

### 6.3 Die Ordnung auf den Funktions- und Konstantensymbolen

Standardmäßig wird diejenige LPO zur Interpretation des Symbols  $\succ$  verwendet, die durch die folgende Ordnung  $>_F$  auf den Funktions- und Konstantensymbolen (bzw. deren transitivem Abschluß) induziert wird: Es gilt  $g >_F f$ , falls

1. die Stelligkeit<sup>6</sup> von  $g$  größer ist als die von  $f$ , oder
2.  $g$  und  $f$  die gleiche Stelligkeit haben, und  $g$  in der alphabetischen Ordnung hinter  $f$  steht.

Diese Ordnung hat sich für die meisten Probleme als geeignet erwiesen (siehe jedoch das Beispiel in Abschnitt 7.2.1).

Der Benutzer kann andere Ordnungen durch Prolog-Fakten wie

```
precedence(f,g).
```

definieren. Symbole, die im transitiven Abschluß dieser Fakten nicht vergleichbar sind, bleiben gemäß dem oben beschriebenen Standard angeordnet.

<sup>6</sup> Konstantensymbole werden als nullstellige Funktionssymbole aufgefaßt.

## 6.4 Die verwendete Vervollständigungs- und Normalisierungsprozedur

Für die Effizienz der Implementierung ist es von entscheidender Bedeutung, welche Vervollständigungs- und Normalisierungsprozedur man verwendet. Dazu ist eine Heuristik implementiert, die die möglichen Regelanwendungen bewertet; auch Vervollständigungs- und Normalisierungsschritte werden miteinander verglichen.

Um die jeweils günstigste Regelanwendung auswählen zu können, werden *alle* möglichen Regelanwendungen, d. h. alle Möglichkeiten, die Kritisches-Paar-, die Kompositions-, die Vereinfachungs- oder die Ableitungsregel anzuwenden, in einer nach ihrer Bewertung sortierten Liste gehalten. Dabei wird nur *eine* solche Liste berechnet für *alle* Vervollständigungs- und Normalisierungsfolgen, die sich aus den verschiedenen zum Abschluß eines Astes zu betrachtenden Unifikationsproblemen ergeben.

Nach jedem Vervollständigungs- oder Normalisierungsschritt werden

1. die Constraints der neuen Terme oder Regeln durch eine Äquivalenzumformung auf Normalform gebracht (Abschnitt 6.2);
2. Terme oder Regeln mit inkonsistenten Constraints und solche, die von anderen subsumiert werden, werden entfernt;
3. die neuen Möglichkeiten, Reduktionsregeln anzuwenden, werden bestimmt, bewertet und in die Liste der möglichen Regelanwendungen einsortiert;
4. Möglichkeiten, die nicht mehr bestehen, weil ein beteiligter Term oder eine beteiligte Reduktionsregel gelöscht worden ist, werden aus der Liste entfernt.

Bezeichne  $M_i$  ( $i = 1, 2$ ) die Möglichkeit, durch Anwendung der Regel  $r_i$  auf den Term  $t_i$  (der auch eine Regel sein kann) neue Terme  $t_i^1, \dots, t_i^k$  abzuleiten (die schon normalisiert sind).  $G(t)$  sei das Gewicht des Terms  $t$ ; <sup>7</sup>  $\text{Index}(t)$  sei der einem Term (oder einer Regel) eindeutig zugeordnete Index, d. h., ist  $\text{Index}(t) = n$ , dann ist  $t$  der  $n$ te Term, der in der Vervollständigungs- und Normalisierungsfolge aufgetreten ist. Mit diesen Bezeichnungen lassen sich die von der implementierten Heuristik für die Bewertung der möglichen Regelanwendungen verwendeten Kriterien folgendermaßen formulieren:<sup>8</sup>

Die Möglichkeit  $M_1$  ist besser zu bewerten als  $M_2$ , wenn:

1.  $\max(\text{Index}(r_1), \text{Index}(t_1)) - \max(\text{Index}(r_2), \text{Index}(t_2)) > d$ .  
Dieses erste Kriterium stellt die Fairneß (Def. 4.6.5) aller zugleich betrachteten Vervollständigungs- und Normalisierungsfolgen sicher. Der Wert  $d$  ist so hoch gewählt, daß das Kriterium gewöhnlich keine Rolle spielt (in der Praxis etwa  $d = 300$ ). Nur wenn eine Möglichkeit *zu* lange nicht ausgewählt wurde, greift es doch.

<sup>7</sup>  $G(t)$  ist die Anzahl aller in dem Term  $t$  (einschließlich seines Constraints) auftretenden Konstanten- und Funktionssymbole, Variablen und logischen Operatoren.

<sup>8</sup> Die Kriterien sind in der Reihenfolge ihrer Wichtigkeit aufgelistet. Nur wenn ein Kriterium keine unterschiedliche Bewertung zur Folge hat, wird das nächste betrachtet.

2.  $M_1$  eine Vereinfachung oder Komposition ist,  $M_2$  aber eine Anwendung der Kritisches-Paar- oder der Ableitungsregel.  
Durch dieses Kriterium wird die Zahl der Regeln und Normalformen klein gehalten, denn nach Vereinfachungen und Kompositionen wird der alte Term (bzw. die Regel) entfernt.
3.  $\max(G(\mathbf{t}_1^1), \dots, G(\mathbf{t}_1^k)) < \max(G(\mathbf{t}_2^1), \dots, G(\mathbf{t}_2^k))$ .
4.  $\max(G(\mathbf{r}_1), G(\mathbf{t}_1)) < \max(G(\mathbf{r}_2), G(\mathbf{t}_2))$ .
5.  $\min(G(\mathbf{r}_1), G(\mathbf{t}_1)) < \min(G(\mathbf{r}_2), G(\mathbf{t}_2))$ .
6.  $\max(\text{Index}(\mathbf{r}_1), \text{Index}(\mathbf{t}_1)) < \max(\text{Index}(\mathbf{r}_2), \text{Index}(\mathbf{t}_2))$ .
7.  $\min(\text{Index}(\mathbf{r}_1), \text{Index}(\mathbf{t}_1)) < \min(\text{Index}(\mathbf{r}_2), \text{Index}(\mathbf{t}_2))$ .

Es hat sich sehr schnell gezeigt, daß es unbedingt notwendig ist, das Termgewicht der beteiligten Regeln und Terme zu berücksichtigen und solche mit einem geringen Gewicht zu bevorzugen. Die oben beschriebenen auf die verschiedenen Termgewichte gestützten Kriterien 3–5 haben sich aber erst durch viele Experimente herausgebildet und als günstig erwiesen.

## 6.5 Ein- und Ausgabe

Da die Implementierung als Modul an den Beweiser  $\mathcal{I}^{\mathcal{A}P}$  angebunden ist, müssen die zu lösenden Probleme in  $\mathcal{I}^{\mathcal{A}P}$ s Eingabesyntax angegeben werden [Hähnle *et al.*, 1992].

Ist die  $\mathcal{I}^{\mathcal{A}P}$ -Variable `eqdebuglevel` von Null verschieden, wird ein Protokoll der ausgeführten Vervollständigungs- und Normalisierungsschritte ausgegeben. Der dieser Variablen zugewiesene Wert legt fest, welche Einzelheiten ausgegeben werden.

Als Beispiel ist hier das Protokoll wiedergegeben, das man erhält, wenn

$$\begin{aligned} E(A) &= \{(\forall x)(\forall y)(p(x, y) \approx p(x, x))\} \\ \mathcal{P}(A) &= \{\{p(c, a), p(c, b)\}\} . \end{aligned}$$

Zunächst werden die Menge  $E(A)$  der Gleichungen des zu schließenden Astes und die zu lösenden  $E$ -Unifikationsprobleme  $\mathcal{P}(A)$  ausgegeben. Quantifizierungen wie  $(\forall x)(\forall y)$  werden in der Form  $[X, Y]$  angegeben. Das Symbol  $\backslash =$  kennzeichnet die zu lösenden Unifikationsprobleme (im Beispiel nur eines):

```
Beginning search for closure with equality of branch b1
Equalities extracted from Branch:
1: [X,Y]   p(X,Y) = p(X,X)
2: [X,Y]   p(X,X) = p(X,Y)

Problems extracted from Branch:
(1,1): p(c,a) \= p(c,b)
```

Das initiale System zu den Gleichungen wird ausgegeben. Regeln mit inkonsistenten Regeln sind schon entfernt worden:

```
Initial System:
1: [X,Y] p(X,Y)=>p(X,X) << <[] and [gr(Y,X)]>
2: [X,Y] p(X,X)=>p(X,Y) << <[] and [gr(X,Y)]>
```

Nun beginnt die Vervollständigung und Normalisierung. Wie man an dem Hinweis „0 left“ erkennt, gibt es nur eine möglich Regelanwendung; diese wird ausgewählt.

```
Chosen possibility (0 left):
critical_pair:
2: [X,Y] p(X,X)=>p(X,Y) << <[] and [gr(X,Y)]>
--- 2: [X,Y] p(X,X)=>p(X,Y) << <[] and [gr(X,Y)]> at [1] ---->
[X,Y,Z,U] p(Z,Y)=>p(Z,U) << <[] and [gr(Z,U),gr(Z,Y)]>
```

Die obige Ausgabe bedeutet, daß die Regel

$$r_2 = (\forall x)(\forall y)(p(x, x) \rightsquigarrow p(x, y) \ll \langle id, x \succ y \rangle)$$

and der Position  $\langle 1 \rangle$  auf sich selbst angewendet werden kann.

Critical pair rule applies

```
New Rules:
3: [X,Y,Z] p(Y,X)=>p(Y,Z) << <[] and [gr(Y,Z),gr(Y,X),gr(X,Z)]>
4: [X,Y,Z] p(Y,Z)=>p(Y,X) << <[] and [gr(Y,Z),gr(Y,X),gr(Z,X)]>
```

Die beiden Regeln, die Ergebnis der Anwendung der Kritisches-Paar-Regel sind, werden ausgegeben. Ihre Constraints haben Normalform, denn die Äquivalenzumformungsregel ist schon angewendet worden.

```
3: [X,Y,Z] p(Y,X)=>p(Y,Z) << <[] and [gr(Y,Z),gr(Y,X),gr(X,Z)]>
subsumed by
4: [X,Y,Z] p(Y,Z)=>p(Y,X) << <[] and [gr(Y,Z),gr(Y,X),gr(Z,X)]>
```

Die erste der neuen Regeln wird also von der zweiten subsumiert. Nun wird das neue Reduktionssystem  $\mathcal{R}^1$  ausgegeben:

```
System:
1: [X,Y] p(X,Y)=>p(X,X) << <[] and [gr(Y,X)]>
2: [X,Y] p(X,X)=>p(X,Y) << <[] and [gr(X,Y)]>
4: [X,Y,Z] p(Y,Z)=>p(Y,X) << <[] and [gr(Y,Z),gr(Y,X),gr(Z,X)]>
0 possibilities left
```

Der Hinweis „0 possibilities left“ zeigt, daß die Vervollständigung abgeschlossen ist.

Als nächstes wird eine Normalisierung ausgeführt. Die neue Regel  $r_4$  wird auf den Term  $p(c, a)$  angewendet. „Non-simplification rule applies“ bedeutet, daß es sich um eine Ableitung, nicht um eine Vereinfachung handelt.

```

Chosen possibility (2 left):
non_simplification:
[1,1,1,0]: [] p(c,a) << <[] and []>
  --- 4: [X,Y,Z] p(Y,Z)=>p(Y,X) <<
          <[] and [gr(Y,Z),gr(Y,X),gr(Z,X)]> at [] ---->
[X,Y,Z] p(c,X) << <[] and [gr(a,X),gr(c,X)]>

Non-simplification rule applies

New Terms:
[1,1,1,1]: [X] p(c,X) << <[] and [gr(a,X),gr(c,X)]>

Terms:
[1,1,1,0]: [] p(c,a) << <[] and []>
[1,1,1,1]: [X] p(c,X) << <[] and [gr(a,X),gr(c,X)]>
2 possibilities left

```

Die Kennzeichnung des neuen Terms — er ist universell bezüglich der Variablen  $x$  — mit [1,1,1,1] bedeutet, daß es sich um die Normalform Nummer 1 des Terms auf der linken Seite des ersten Problems im ersten simultanen Unifikationsproblem handelt.

```

Chosen possibility (2 left):
non_simplification:
[1,1,r,0]: [] p(c,b) << <[] and []>
  --- 4: [X,Y,Z] p(Y,Z)=>p(Y,X) <<
          <[] and [gr(Y,Z),gr(Y,X),gr(Z,X)]> at [] ---->
[X,Y,Z] p(c,X) << <[] and [gr(b,X),gr(c,X)]>

Non-simplification rule applies

New Terms:
[1,1,r,1]: [X] p(c,X) << <[] and [gr(b,X),gr(c,X)]>

Terms:
[1,1,r,0]: [] p(c,b) << <[] and []>
[1,1,r,1]: [X] p(c,X) << <[] and [gr(b,X),gr(c,X)]>
2 possibilities left

```

Nach einer zweiten Ableitung kann nun das Unifikationsproblem gelöst, und der Ast des Tableaus geschlossen werden. Die Constraints  $\langle id, (a \succ x) \wedge (b \succ x) \wedge (c \succ x) \rangle$  und  $\epsilon$  werden gefunden, die Lösungen des Problems definieren.

```

[1,1] closed with constraint(s)
      <[] and [gr(a,X),gr(b,X),gr(c,X)]>

by combination of the normal forms
[1,1,r,1]: [X] p(c,X) << <[] and [gr(b,X),gr(c,X)]>
[1,1,1,1]: [X] p(c,X) << <[] and [gr(a,X),gr(c,X)]>

```

[1,1] closed with constraint(s)  
 <[] and []>

by combination of the normal forms  
 [1,1,r,1]: [X] p(c,X) << <[] and [gr(b,X),gr(c,X)]>  
 [1,1,l,0]: [] p(c,a) << <[] and []>

Der leere Constraint ist hier eine Lösung, weil  $(\forall x)(p(c, x) \ll \langle id, (b \succ x) \wedge (c \succ x) \rangle)$  universell ist bezüglich  $x$ , und die eigentlich zur Unifikation notwendige Belegung von  $x$  mit  $a$  also unerheblich ist.

Der einfachere Constraint  $\epsilon$  und die ihn erfüllende leere Substitution  $id$  werden zum Abschluß des Astes ausgewählt.

Branch closed with instantiation []

----- PROOF -----

# 7 Experimente und Beispiele

---

In diesem Kapitel sind einige Experimente mit der in Kapitel 6 beschriebenen Implementierung des neuen Verfahrens für die Gleichheitsbehandlung im Tableauekalkül beschrieben. Dabei wurden auch Beispiele, bei denen es nur um die Vervollständigung von Reduktionssystemen oder die Lösung einzelner  $E$ -Unifikationsprobleme geht, als Prädikatenlogische Probleme formuliert. Nämlich in der Weise, daß ein Tableau mit genau einem Ast entsteht, zu dessen Abschluß gerade das Problem des jeweiligen Beispiels gelöst werden muß, und darum von dem Tableaubeweiser  $\mathfrak{3TAP}$  an das neue Gleichheitsmodul übergeben wird.

Zum Vergleich sind (teilweise) bei den rein universellen Problemen die Laufzeiten des Gleichheitsbeweisers SbREVE angegeben. Er ist eine effiziente Implementierung des UKBA [Hsiang & Mzali, 1988].

Auch sind zum Vergleich die Laufzeiten der alten Gleichheitsbehandlung des Beweisers  $\mathfrak{3TAP}$ , die auf dem äquivalenzklassenbasierten Verfahren aus [Beckert, 1991] beruht, angegeben, sofern die Probleme mit ihr gelöst werden können (wo keine Zeiten angegeben sind, ist das also nicht möglich).<sup>1</sup>

Hier einige Hinweise zu den in den Statistiken angegebenen Werten:

- Die angegebenen Laufzeiten wurden auf einer SUN SPARC 10 Workstation gemessen. Das gilt auch für die zum Vergleich angegebenen Laufzeiten der alten Gleichheitsbehandlung des Beweisers  $\mathfrak{3TAP}$ . Die angegebenen Laufzeiten des Beweiser SbREVE sind [Hsiang & Mzali, 1988] und [Anantharaman & Bonacina, 1989] entnommen. Sie wurden auf einer langsameren SUN 3/260 Workstation gemessen und sind darum nicht ohne weiteres mit den anderen Zeiten vergleichbar.
- Die angegebenen Laufzeiten schließen immer auch die für die Anwendung der Tableauregeln benötigte Zeit ein; es sind also die für den gesamten Beweis des jeweiligen Theorems benötigten Zeiten und nicht nur die für die Gleichheitsbehandlung.
- Der Wert „Anwendungen“ in den Statistiken ist die Zahl der Transformationen bzw. Normalisierungen, die auf Regelanwendungen beruhen, also Anwendungen der Kritisches-Paar-Regel, der Kompositions- und der Vereinfachungsregel bzw. der Ableitungs- und der Vereinfachungsregel. Nicht dazu gehören Äquivalenzumformungen, Subsumtionen und Löschungen. Zu den „möglichen Anwendungen“ gehören zusätzlich solche Anwendungen dieser Transformationsregeln, die nicht ausgeführt wurden, weil entweder vorher schon eine Lösung

---

<sup>1</sup> In den Statistiken sind diese Zeiten mit „Laufzeit  $\mathfrak{3TAP}$ “ bezeichnet.

gefunden wurde, oder aber die an der Anwendung beteiligten Regeln bzw. Terme nicht persistent waren, und durch andere Transformationen wieder entfernt wurden.

- Die Zahl der Äquivalenzumformungen ist nicht angegeben. Die Implementierung bringt alle neu entstehenden Regeln und Terme (bzw. ihre Constraints) auf die in Abschnitt 6.2 beschriebene Normalform. Die Zahl der Äquivalenzumformungen ist also gleich der Zahl der berechneten Regeln und Terme (oder auch kleiner, wenn Regeln oder Terme entstehen, deren Constraints schon Normalform haben).
- Wird nur die faire Vervollständigung einer Gleichungsmenge berechnet, ohne daß ein zu lösendes  $E$ -Unifikationsproblem gegeben ist, schließen die angegebenen Daten immer auch den Aufwand ein, der entsteht, nachdem schon alle Regeln der endlichen Vervollständigung berechnet sind. Zu diesem Zeitpunkt ist die Vervollständigung in der Regel noch nicht fair. Weitere Transformationen sind auszuführen. Diese letzte Phase der Vervollständigung kann auch als Test dafür aufgefaßt werden, daß das berechnete Reduktionssystem tatsächlich vollständig ist — was der Fall ist, wenn alle noch entstehenden Regeln durch Löschung oder Subsumtion wieder entfernt werden können.
- Der mit „Probleme“ bezeichnete Wert ist die Anzahl der jeweils zum Abschluß des Tableaus zu lösenden  $E$ -Unifikationsprobleme.
- Die Bezeichnung einer Regel mit  $\mathbf{r}_n$  in der Auflistung eines berechneten Reduktionssystems bedeutet, daß  $\text{Index}(\mathbf{r}_n) = n$ , daß also  $\mathbf{r}_n$  die  $n$ te während der Vervollständigung auftretende Regel ist.

Im folgenden Abschnitt sind zunächst die Beweise einiger (mehr oder weniger) einfacher Probleme dargestellt, die die besonderen Eigenschaften des neuen Verfahrens und seiner Implementierung aufzeigen.

## 7.1 Testprobleme

### 7.1.1 Ein einfaches rein universelles Problem

Zu beweisen ist die Tautologie

$$T_1 = (\forall x)(x \approx g(x)) \supset (g(g(g(g(a)))) \approx a) .$$

Nachdem das entsprechende Tableau erstellt ist, ist das rein universelle  $E$ -Unifikationsproblem

$$\langle E, s, t \rangle = \langle \{(\forall x)(x \approx g(x))\}, g(g(g(g(a)))) \rangle, a \rangle$$

zu lösen. Dafür sind keine Transformationen des initialen Reduktionssystems notwendig. Fünf vereinfachende Normalisierungen der Terme genügen (siehe Statistik in Tabelle 7.1).



Vervollständigung	Kritisches Paar	0	Normalisierung		
Anwendungen	Komposition	0	Anwendungen	5	Ableitung
von möglichen	Vereinfachung	0	von möglichen	15	Vereinfachung
Regeln in $\mathcal{R}^0$	Löschung	1			Löschung
Regeln in $\mathcal{R}^n$	Subsumtion	0			Subsumtion
Laufzeit [s]	0,15	Probleme	1	Backtr.	0
				Zeit ${}_3T^AP$ [s]	0,13

Tabelle 7.1: Statistik für den Beweis der Tautologie  $T_1$ .

### 7.1.2 Ein einfaches rein starres Problem

Zu beweisen ist die Tautologie

$$T_2 = (\forall x)((f(a) \approx a) \wedge (g(g(x)) \approx f(a)) \wedge \neg(g(g(g(x))) \approx x) \vee (p \wedge \neg p))$$

Das entstehende Tableau hat zwei Äste. Wegen der Verzweigung können die auftretenden Gleichungen mit der Methode aus Satz 5.1.13 nicht als universell erkannt werden. Darum ist das rein starre Problem

$$\langle E, s, t \rangle = \langle \{f(a) \approx a, g(g(x)) \approx f(a)\}, g(g(g(x))), x \rangle$$

zu lösen (siehe auch Beispiel 3.6.19 und 4.6.2). Die Statistik für den Beweis ist in Tabelle 7.2 enthalten.

Vervollständigung	Kritisches Paar	0	Normalisierung		
Anwendungen	Komposition	0	Anwendungen	1	Ableitung
von möglichen	Vereinfachung	0	von möglichen	2	Vereinfachung
Regeln in $\mathcal{R}^0$	Löschung	2			Löschung
Regeln in $\mathcal{R}^n$	Subsumtion	0			Subsumtion
Laufzeit [s]	0,13	Probleme	1	Backtr.	0
				Zeit ${}_3T^AP$ [s]	0,13

Tabelle 7.2: Statistik für den Beweis der Tautologie  $T_2$ .

### 7.1.3 Systeme mit schnell wachsenden Vervollständigungen

Gleichungen, die sich nur mit Hilfe von Constraints orientieren lassen, und die zudem freie Variablen enthalten, die nur in jeweils einer Seite der Gleichung auftreten, haben die unangenehme Eigenschaft, daß sie zu sehr schnell wachsenden Vervollständigungen führen, denn mit ihnen können in der Regel sehr viele verschiedene kritische Paare gebildet werden. Dies verschlimmert sich noch, wenn Gleichungen der Form  $x \approx y$  auftreten, die mit jeder anderen Gleichung (bzw. Reduktionsregel) ein kritisches Paar bilden.

Die Vervollständigung der Gleichungsmenge  $E_1 = \{x_1 \approx y_1\}$  enthält nur die beiden Regeln

$$\begin{aligned} \mathbf{r}_1 &= x_1 \rightsquigarrow y_1 \ll \langle id, x_1 \succ y_1 \rangle \\ \mathbf{r}_2 &= y_1 \rightsquigarrow x_1 \ll \langle id, y_1 \succ x_1 \rangle \end{aligned}$$

Aber schon bei der Vervollständigung von

$$E_2 = \{x_1 \approx y_1, x_2 \approx y_2\}$$

treten acht kritische Paare auf, und  $\mathcal{R}^\infty$  enthält die zwölf in Tabelle 7.3 aufgeführten Regeln.

$\mathbf{r}_1$	=	$x_1 \rightsquigarrow y_1 \ll \langle id, x_1 \succ y_1 \rangle$
$\mathbf{r}_2$	=	$y_1 \rightsquigarrow x_1 \ll \langle id, y_1 \succ x_1 \rangle$
$\mathbf{r}_3$	=	$x_2 \rightsquigarrow y_2 \ll \langle id, x_2 \succ y_2 \rangle$
$\mathbf{r}_4$	=	$y_2 \rightsquigarrow x_2 \ll \langle id, y_2 \succ x_2 \rangle$
$\mathbf{r}_5$	=	$y_2 \rightsquigarrow y_1 \ll \langle \{x_2/x_1\}, (x_1 \succ y_1) \wedge (x_1 \succ y_2) \wedge (y_2 \succ y_1) \rangle$
$\mathbf{r}_6$	=	$y_1 \rightsquigarrow y_2 \ll \langle \{x_2/x_1\}, (x_1 \succ y_1) \wedge (x_1 \succ y_2) \wedge (y_1 \succ y_2) \rangle$
$\mathbf{r}_9$	=	$y_2 \rightsquigarrow x_1 \ll \langle \{x_2/y_1\}, (y_1 \succ x_1) \wedge (y_1 \succ y_2) \wedge (y_2 \succ x_1) \rangle$
$\mathbf{r}_{10}$	=	$x_1 \rightsquigarrow y_2 \ll \langle \{x_2/y_1\}, (x_1 \succ y_2) \wedge (y_1 \succ x_1) \wedge (y_1 \succ y_2) \rangle$
$\mathbf{r}_{13}$	=	$x_2 \rightsquigarrow y_1 \ll \langle \{y_2/x_1\}, (x_1 \succ y_1) \wedge (x_1 \succ x_2) \wedge (x_2 \succ y_1) \rangle$
$\mathbf{r}_{14}$	=	$y_1 \rightsquigarrow x_2 \ll \langle \{y_2/x_1\}, (x_1 \succ y_1) \wedge (x_1 \succ x_2) \wedge (y_1 \succ x_2) \rangle$
$\mathbf{r}_{17}$	=	$x_2 \rightsquigarrow x_1 \ll \langle \{y_2/y_1\}, (y_1 \succ x_1) \wedge (y_1 \succ x_2) \wedge (x_2 \succ x_1) \rangle$
$\mathbf{r}_{18}$	=	$x_1 \rightsquigarrow x_2 \ll \langle \{y_2/y_1\}, (x_1 \succ x_2) \wedge (y_1 \succ x_1) \wedge (y_1 \succ x_2) \rangle$

**Tabelle 7.3:** Vervollständigung von  $E_2$ .

Die Vervollständigung von  $E_3 = \{x_1 \approx y_1, x_2 \approx y_2, x_3 \approx y_3\}$  enthält schon 126 Reduktionsregeln. Ihre Berechnung dauert über 10 Minuten.

Ähnlich problematisch ist es, eine Vervollständigung der beiden auf den ersten Blick recht einfach erscheinenden Gleichungen

$$f(x) \approx g(y) \quad \text{und} \quad f(f(u)) \approx g(g(v))$$

zu berechnen (siehe Statistik in Tabelle 7.4).

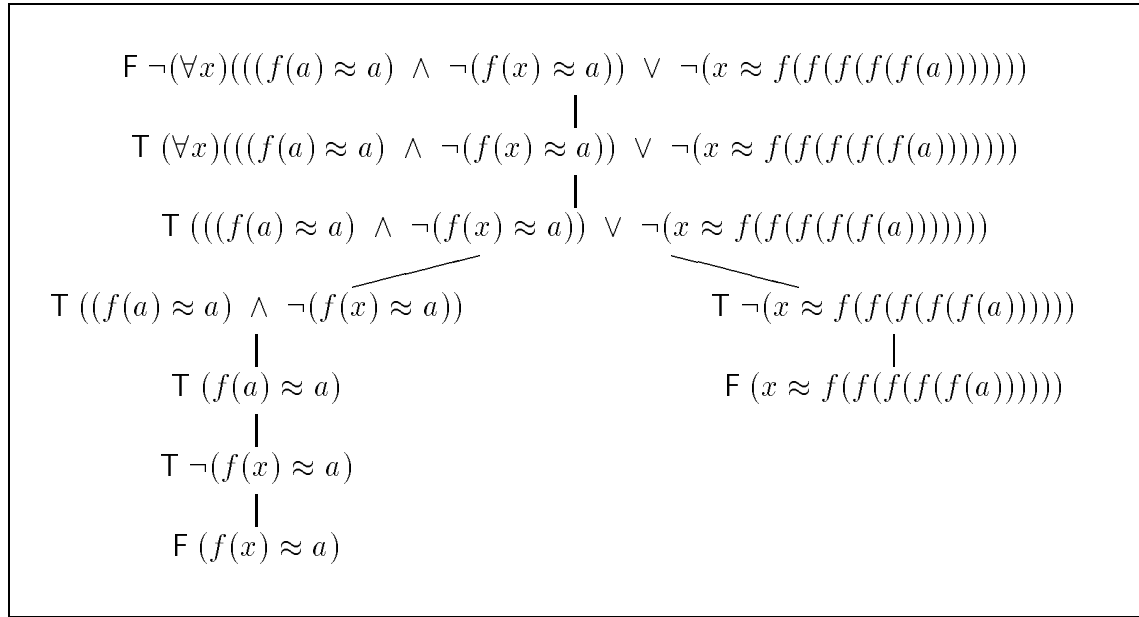
Vervollständigung		Kritisches Paar	19
Anwendungen	31	Komposition	10
von möglichen	50	Vereinfachung	2
Regeln in $\mathcal{R}^0$	4	Löschung	35
Regeln in $\mathcal{R}^\infty$	16	Subsumtion	15
Laufzeit [s]	6,1	Probleme	—

**Tabelle 7.4:** Statistik für die Vervollständigung der Menge von Gleichungen  $E = \{f(x) \approx g(y), f(f(u)) \approx g(g(v))\}$ .

In der Praxis spielt diese Problematik keine große Rolle, weil es nicht nötig ist, eine komplette Vervollständigung solcher Gleichungssysteme zu berechnen, um den Ast eines Tableaus abzuschließen. Reduktionsregeln wie

$$x \rightsquigarrow y \ll \langle id, x \succ y \rangle$$

liefern meist sofort eine Lösung des Unifikationsproblems. Die Komplexität der Vervollständigung ist ein Ausdruck der Tatsache, daß viele verschiedene Unifikatoren existieren, wenn solche Reduktionsregeln vorhanden sind.

Abbildung 7.1: Das voll expandierte Tableau für die Formel  $T_3$ .

#### 7.1.4 Ein Beispiel für die Berechnung weiterer Unifikatoren

Wie in Abschnitt 4.10 beschrieben, ist es in manchen Fällen zur Lösung simultaner  $E$ -Unifikationsprobleme notwendig, weitere Unifikatoren der einfachen Probleme zu berechnen, indem man die Regeln aus der Inversen der Vervollständigung auf die schon gefundenen allgemeinsten Unifikatoren anwendet.

Beispielsweise wird zum Beweis der Tautologie

$$T_3 = \neg(\forall x)((f(a) \approx a) \wedge \neg(f(x) \approx a)) \vee \neg(x \approx f(f(f(f(f(a))))))$$

zunächst das in Abbildung 7.1 dargestellte Tableau generiert. Zum Abschluß seines linken Astes ist das rein starre  $E$ -Unifikationsproblem

$$\langle E, s, t \rangle = \langle f(a) \approx a, f(x), a \rangle$$

zu lösen. Berechnet wird zunächst der bezüglich der Subsumtionsrelation  $\leq_E$  allgemeinste Unifikator  $\{x/a\}$ . Das Tableau kann aber erst geschlossen werden, nachdem aus diesem sukzessive die weiteren Unifikatoren

$$\begin{aligned} &\{x/f(a)\} \\ &\{x/f(f(a))\} \\ &\{x/f(f(f(a)))\} \\ &\{x/f(f(f(f(a))))\} \\ &\{x/f(f(f(f(f(a))))\} \end{aligned}$$

abgeleitet worden sind.<sup>2</sup>

Der Beweis gelingt in 0,3s. Mit der äquivalenzklassenbasierten Gleichheitsbehandlung benötigt  $\exists T^{\mathcal{AP}}$  genauso lange, was nicht überrascht, da hier alle Vorteile der vervollständigungsbasierten Gleichheitsbehandlung verloren gehen.

<sup>2</sup> Die zwar bezüglich  $\leq$  allgemeinste Unifikatoren sind, nicht aber bezüglich  $\leq_E$ .

## 7.2 Beispiele für rein universelle Probleme

### 7.2.1 Gruppentheorie

Die Axiome der Gruppentheorie sind *das* Standardbeispiel für die Vervollständigung rein universeller Systeme. Gibt man die Axiome

$$\begin{aligned} &(\forall x)(p(e, x) \approx x) \\ &(\forall x)(p(i(x), x) \approx e) \\ &(\forall x)(\forall y)(\forall z)(p(x, p(y, z)) \approx p(p(x, y), z)) \end{aligned}$$

vor, so wird die in Tabelle 7.5 angegebene endliche, zehn Regeln enthaltende Vervollständigung berechnet (siehe Statistik in Tabelle 7.6). Die Regeln sind in der Reihenfolge aufgelistet, in der sie während der Vervollständigung auftreten. Dies ist das übliche vollständige Reduktionssystem für die Gruppentheorie, wie es beispielsweise auch von SbREVE gefunden wird (in 6 Sekunden, bei 76 berechneten kritischen Paaren).

$\mathbf{r}_1$	=	$(\forall x)(\forall y)(\forall z)(p(p(x, y), z) \rightsquigarrow p(x, p(y, z))) \ll \epsilon$
$\mathbf{r}_2$	=	$(\forall x)(p(e, x) \rightsquigarrow x \ll \epsilon)$
$\mathbf{r}_3$	=	$(\forall x)(p(i(x), x) \rightsquigarrow e \ll \epsilon)$
$\mathbf{r}_6$	=	$(\forall x)(\forall y)(p(i(x), p(x, y)) \rightsquigarrow y \ll \epsilon)$
$\mathbf{r}_{12}$	=	$i(e) \rightsquigarrow e \ll \epsilon$
$\mathbf{r}_{25}$	=	$(\forall x)(p(x, e) \rightsquigarrow x \ll \epsilon)$
$\mathbf{r}_{30}$	=	$(\forall x)(p(x, i(x)) \rightsquigarrow e \ll \epsilon)$
$\mathbf{r}_{37}$	=	$(\forall x)(i(i(x)) \rightsquigarrow x \ll \epsilon)$
$\mathbf{r}_{41}$	=	$(\forall x)(\forall y)(p(x, p(i(x), y)) \rightsquigarrow y \ll \epsilon)$
$\mathbf{r}_{104}$	=	$(\forall x)(\forall y)(i(p(x, y)) \rightsquigarrow p(i(y), i(x))) \ll \epsilon$

**Tabelle 7.5:** Die Vervollständigung der Gruppentheorie–Axiome.

Vervollständigung		Kritisches Paar	91
Anwendungen	174	Komposition	70
von möglichen	1494	Vereinfachung	13
Regeln in $\mathcal{R}^0$	6	Löschung	181
Regeln in $\mathcal{R}^\infty$	10	Subsumtion	70
Laufzeit [s]	29,4	Probleme	—

**Tabelle 7.6:** Statistik für die Vervollständigung der Gruppentheorie–Axiome.

Das Theorem

$$(\forall x)(p(e, x) \approx x)$$

der Gruppentheorie läßt sich beispielsweise in 5s beweisen.  $\exists T^A P$  benötigt hierzu 8,3s, also nicht viel länger. Um die Eindeutigkeit der Vernüpfung in einer Gruppe, d. h.

$$(\forall x)(\forall y)(\forall z)(p(x, y) \approx p(x, z) \supset (y \approx z)) ,$$

zu beweisen, benötigt  $\exists T^A P$  schon 246s. Mit dem neuen Verfahren jedoch kann dieses Theorem in 19,6s bewiesen werden. Auch komplexere Theoreme der Gruppentheorie, an denen äquivalenzklassenbasierte Verfahren scheitern, können in ca. 30s

bewiesen werden, denn nach dieser Zeit stehen alle Regeln des vollständigen Reduktionssystems zur Verfügung.

Verwendet man statt der üblichen Anordnung  $i >_F p >_F e$  der auftretenden Funktionssymbole die ungünstige Anordnung  $p >_F i >_F e$ ,<sup>3</sup> kann keine *endliche* Vervollständigung berechnet werden. In Tabelle 7.7 sind die ersten 16 persistenten Regeln angegeben, die in diesem Fall abgeleitet werden. Wie man sieht treten nun auch Regeln mit nicht-leeren Constraints auf (alleine schon deswegen wird eine mit Hilfe des UKBA berechnete Vervollständigung andere Regeln enthalten).

$r_1$	$=$	$(\forall x)(\forall y)(\forall z)(p(p(x, y), z) \rightsquigarrow p(x, p(y, z))) \ll \epsilon$
$r_2$	$=$	$(\forall x)(p(e, x) \rightsquigarrow x \ll \epsilon)$
$r_3$	$=$	$(\forall x)(p(i(x), x) \rightsquigarrow e \ll \epsilon)$
$r_6$	$=$	$(\forall x)(\forall y)(p(i(x), p(x, y)) \rightsquigarrow y \ll \epsilon)$
$r_{12}$	$=$	$i(e) \rightsquigarrow e \ll \epsilon$
$r_{30}$	$=$	$(\forall x)(p(x, i(x)) \rightsquigarrow e \ll \epsilon)$
$r_{37}$	$=$	$(\forall x)(i(i(x)) \rightsquigarrow x \ll \epsilon)$
$r_{38}$	$=$	$(\forall x)(p(x, e) \rightsquigarrow x \ll \epsilon)$
$r_{45}$	$=$	$(\forall x)(\forall y)(p(x, p(i(x), y)) \rightsquigarrow y \ll \epsilon)$
$r_{143}$	$=$	$(\forall x)(\forall y)(p(y, i(p(x, y))) \rightsquigarrow i(x) \ll \epsilon)$
$r_{171}$	$=$	$(\forall x)(\forall y)(p(y, i(x)) \rightsquigarrow i(p(x, i(y))) \ll \langle id, y \succ x \rangle)$
$r_{196}$	$=$	$(\forall x)(\forall y)(p(y, x) \rightsquigarrow i(p(i(x), i(y))) \ll \langle id, y \succ i(x) \rangle)$
$r_{210}$	$=$	$(\forall x)(\forall y)(\forall z)(p(i(p(x, y)), p(x, p(y, z))) \rightsquigarrow z \ll \epsilon)$
$r_{289}$	$=$	$(\forall x)(\forall y)(\forall z)(p(y, p(i(p(x, y)), z)) \rightsquigarrow p(i(x), z) \ll \epsilon)$
$r_{330}$	$=$	$(\forall x)(\forall y)(\forall z)(p(y, p(z, i(p(x, p(y, z)))) \rightsquigarrow i(x) \ll \epsilon)$
$r_{332}$	$=$	$(\forall x)(\forall y)(p(x, p(y, i(y))) \rightsquigarrow x \ll \epsilon)$

**Tabelle 7.7:** Die ersten 16 persistenten Regeln, die bei einer Vervollständigung der Gruppentheorie-Axiome auftreten, wenn  $p >_F i$ .

## 7.2.2 Mika-Beispiel

Zu vervollständigen ist das Gleichungssystem<sup>4</sup>

$$\begin{aligned}
 &(\forall x)(m(x, 1) \approx x) \\
 &m(a, b) \approx p(a, 1) \\
 &(\forall x)(\forall y)(m(x, y) \approx m(y, x)) \\
 &(\forall x)(\forall y)(p(x, y) \approx p(y, x))
 \end{aligned}$$

Dieses aus [Hsiang & Mzali, 1988] stammende Beispiel zeigt, daß das Constraints verwendende Verfahren auch bei rein universelle Problemen Vorteile haben und sogar dem UKBA überlegen sein kann. Da die beiden die Kommutativität der Funktionssymbole  $m$  und  $p$  ausdrückenden Gleichungen nicht orientiert werden können, kann der Beweiser SbREVE nur eine noch zwei Gleichungen enthaltene Vervollständigung generieren (Tabelle 7.8); dabei werden in zwei Sekunden 16 kritische Paare berechnet.

<sup>3</sup> Macht der Benutzer keine Vorgaben, wird diese ungünstige Ordnung verwendet (siehe Abschnitt 6.3).

<sup>4</sup>  $m$  kann als Multiplikation und  $p$  als Addition verstanden werden.

$(\forall x)(m(x, 1) \rightsquigarrow x)$
$(\forall x)(m(1, x) \rightsquigarrow x)$
$(\forall x)(\forall y)(m(x, y) \approx m(y, x))$
$(\forall x)(\forall y)(p(x, y) \approx p(y, x))$
$m(b, a) \approx p(1, a)$

**Tabelle 7.8:** Die vom Beweiser SbREVE berechnete Vervollständigung für das Mika-Beispiel.

Die Implementierung des Verfahrens aus Kapitel 4 dagegen findet die in Tabelle 7.9 dargestellte, nur aus Reduktionsregeln bestehende und darum sehr viel günstigere Vervollständigung  $\mathcal{R}^\infty$ , die noch dazu in kürzerer Zeit und mit geringerem Aufwand (zwei kritische Paare) berechnet werden kann (siehe Statistik in Tabelle 7.10).<sup>5</sup>

$\mathbf{r}_3$	$=$	$(\forall x)(\forall y)(p(y, x) \rightsquigarrow p(x, y) \ll \langle id, y \succ x \rangle)$
$\mathbf{r}_4$	$=$	$(\forall x)(m(x, 1) \rightsquigarrow x \ll \epsilon)$
$\mathbf{r}_6$	$=$	$(\forall x)(\forall y)(m(y, x) \rightsquigarrow m(x, y) \ll \langle id, y \succ x \rangle)$
$\mathbf{r}_7$	$=$	$m(a, b) \rightsquigarrow p(1, a) \ll \epsilon$
$\mathbf{r}_8$	$=$	$(\forall x)(m(1, x) \rightsquigarrow x \ll \langle id, x \succ 1 \rangle)$

**Tabelle 7.9:** Die von der Implementierung des neuen Verfahrens berechnete Vervollständigung für das Mika-Beispiel.

Vervollständigung		Kritisches Paar	2
Anwendungen	3	Komposition	0
von möglichen	3	Vereinfachung	1
Regeln in $\mathcal{R}^0$	8	Löschung	4
Regeln in $\mathcal{R}^\infty$	5	Subsumtion	3
Laufzeit [s]	0,4	Probleme	—

**Tabelle 7.10:** Statistik für die Vervollständigung der Axiome des Mika-Beispiels.

### 7.2.3 Transitivitäts-Beispiel

Ein weiteres Beispiel dafür, daß es auch für rein universelle Probleme von Vorteil sein kann, Constraints zu verwenden, ist die Vervollständigung des aus zwei Gleichungen bestehenden Systems

$$\begin{aligned} &(\forall x)(\forall y)(\forall z)(h(p(x, y), p(y, z)) \approx h(p(x, y), p(x, z))) \\ &(\forall x)(\forall y)(h(x, y) \approx h(y, x)) \end{aligned}$$

Der Beweiser SbREVE generiert (mit Hilfe des UKBA) eine aus elf Gleichungen bestehende Vervollständigung. Er benötigt dazu über zehn Minuten (!) und berechnet 264 kritische Paare.

<sup>5</sup> Zur Vervollständigung wurde die Ordnung  $m \succ_F p \succ_F b \succ_F a \succ_F 1$  auf den Funktions- und Konstantensymbolen verwendet.

Die Anwendung des Verfahrens aus Kapitel 4 dagegen liefert die in Tabelle 7.11 dargestellte, aus nur vier Reduktionsregeln bestehende Vervollständigung in wenigen Sekunden (siehe Statistik in Tabelle 7.12):

$\mathbf{r}_2$	$=$	$(\forall x)(\forall y)(h(y, x) \rightsquigarrow h(x, y) \ll \langle id, y \succ x \rangle)$
$\mathbf{r}_3$	$=$	$(\forall x)(\forall y)(\forall z)(h(p(x, y), p(y, z)) \rightsquigarrow h(p(x, y), p(x, z)) \ll \langle id, y \succ x \rangle)$
$\mathbf{r}_4$	$=$	$(\forall x)(\forall y)(\forall z)(h(p(x, y), p(x, z)) \rightsquigarrow h(p(x, y), p(y, z)) \ll \langle id, x \succ y \rangle)$
$\mathbf{r}_9$	$=$	$(\forall x)(\forall y)(\forall z)(h(p(y, z), p(x, y)) \rightsquigarrow h(p(z, y), p(x, z)) \ll \langle id, (x \succ y) \wedge (x \succ z) \wedge (y \succ z) \rangle)$

**Tabelle 7.11:** Die berechnete Vervollständigung für das Transitivitäts-Beispiel.

Vervollständigung		Kritisches Paar	2
Anwendungen	8	Komposition	2
von möglichen	18	Vereinfachung	4
Regeln in $\mathcal{R}^0$	4	Löschung	4
Regeln in $\mathcal{R}^\infty$	4	Subsumtion	4
Laufzeit [s]	4,1	Probleme	—

**Tabelle 7.12:** Statistik für die Vervollständigung der Axiome des Transitivitäts-Beispiels.

#### 7.2.4 Weitere Beispiele aus [Hsiang & Mzali, 1988]

Für die Vervollständigung der einen *entropic groupoid* definierenden Gleichungen

$$\begin{aligned}
 &(\forall x)(\forall y)(\forall z)(\forall w)(p(p(x, y), p(z, w)) \approx p(p(x, z), p(y, w))) \\
 &(\forall x)(\forall y)(p(p(x, y), x) \approx x)
 \end{aligned}$$

benötigt SbREVE 14 Sekunden, berechnet dabei 53 kritische Paare und generiert eine aus drei Reduktionsregeln und einer Gleichung bestehende Vervollständigung.

Die Constraints verwendende Implementierung berechnet die in Tabelle 7.13 dargestellte Vervollständigung (siehe Statistik in Tabelle 7.14):

$\mathbf{r}_3$	$=$	$(\forall x)(\forall y)(p(p(x, y), x) \rightsquigarrow x \ll \epsilon)$
$\mathbf{r}_{40}$	$=$	$(\forall x)(\forall y)(\forall z)(p(z, p(y, x)) \rightsquigarrow p(z, x) \ll \epsilon)$
$\mathbf{r}_{41}$	$=$	$(\forall x)(\forall y)(\forall z)(\forall u)p(p(x, z), u) \rightsquigarrow p(p(x, y), u) \ll \langle id, z \succ y \rangle)$
$\mathbf{r}_{71}$	$=$	$(\forall x)(\forall y)(\forall z)(\forall u)(p(p(z, y), x), u) \rightsquigarrow p(z, u) \ll \epsilon)$

**Tabelle 7.13:** Vervollständigung der einen *entropic groupoid* definierenden Gleichungen.

Gelten die drei Gruppenaxiome (wie in Abschnitt 7.2.1), das die Funktion  $h$  definierende Axiom

$$(\forall x)(\forall y)(h(x, y) \approx p(x, p(y, p(i(x), i(y))))))$$

Vervollständigung		Kritisches Paar	91
Anwendungen	112	Komposition	9
von möglichen	658	Vereinfachung	12
Regeln in $\mathcal{R}^0$	4	Löschung	106
Regeln in $\mathcal{R}^\infty$	4	Subsumtion	87
Laufzeit [s]	22,5	Probleme	—

**Tabelle 7.14:** Statistik für die Vervollständigung der Axiome für einen *entropic groupoid*.

und zudem

$$(\forall x)(p(x, p(x, x)) \approx e)$$

dann kann man das Theorem

$$T_h = (\forall x)(\forall y)(h(h(x, y), y) \approx e)$$

beweisen. SbREVE benötigt dazu eine Minute und berechnet 133 kritische Paare. Auch die Implementierung des neuen Verfahrens mit Constraints kann das Theorem beweisen<sup>6</sup> — allerdings ist damit die Leistungsgrenze für rein universelle Probleme praktisch erreicht. Tabelle 7.15 enthält die Statistik für den Beweis.

Vervollständigung		Kritisches Paar	199	Normalisierung			
Anwendungen	478	Komposition	205	Anwendungen	36	Ableitung	0
von möglichen	6567	Vereinfachung	74	von möglichen	58	Vereinfachung	36
Regeln in $\mathcal{R}^0$	10	Löschung	463			Löschung	0
Regeln in $\mathcal{R}^n$	16	Subsumtion	87			Subsumtion	0
Laufzeit [s]			197	Probleme			1

**Tabelle 7.15:** Statistik für den Beweis des Theorems  $T_h$ .

Auch alle weiteren in [Hsiang & Mzali, 1988] gegebenen Beispiele für rein universelle Probleme können mit der Implementierung des neuen Verfahrens gelöst werden.

### 7.3 Aussagen über Łukasiewicz Logiken

Das folgende Beispiel stammt aus [Anantharaman & Bonacina, 1989]. Gegeben ist eine Axiomatisierung<sup>7</sup>

$$\begin{aligned}
&(\forall x)(i(true, x) \approx x) \\
&(\forall x)(\forall y)(\forall z)(i(i(x, y), i(i(y, z), i(x, z)))) \approx true) \\
&(\forall x)(\forall y)(i(i(x, y), y) \approx i(i(y, x), x)) \\
&(\forall x)(\forall y)(i(i(n(x), n(y)), i(y, x)) \approx true)
\end{aligned}$$

<sup>6</sup> Mit der Ordnung  $h >_F i >_F p >_F e$  auf den Funktions- und Konstantensymbolen.

<sup>7</sup>  $i$  steht für die Implikation,  $n$  für die Negation.



der Lukasiewicz Logiken, aus der sich unter anderem die folgenden Theoreme ableiten lassen:<sup>8</sup>

$$\begin{aligned} L_1 &= (\forall x)(i(x, x) \approx true) \\ L_3 &= (\forall x)(i(x, true) \approx true) \\ L_4 &= (\forall x)(\forall y)(i(x, i(y, x)) \approx true) \end{aligned}$$

Tabelle 7.17 enthält Statistiken von Beweisen dieser drei Theoreme. Bei diesen Beweisen ist nur die Vervollständigung des Reduktionssystems aufwendig. Sobald die notwendigen Regeln zur Verfügung stehen, genügt eine einzige Anwendung der Vereinfachungsregel. Darum enthalten die Statistiken keine Daten über die Normalisierung. Tabelle 7.16 zeigt das von der Implementierung generierte Reduktionssystem  $\mathcal{R}^n$ , mit dem der Beweis aller drei Theoreme gelingt.

Anantharaman und Bonacina [1989] zufolge benötigt der den UKBA verwendende Theorembeweiser SBR3, der eine Erweiterung von SbREVE ist, zwischen etwa zwei und dreißig Sekunden für die Beweise.

$\mathbf{r}_1$	$=$	$(\forall x)(i(true, x) \rightsquigarrow x \ll \epsilon)$
$\mathbf{r}_2$	$=$	$(\forall x)(\forall y)(\forall z)(i(i(x, y), i(i(y, Z), i(x, Z))) \rightsquigarrow true \ll \epsilon)$
$\mathbf{r}_4$	$=$	$(\forall x)(\forall y)(i(i(y, x), x) \rightsquigarrow i(i(x, y), y) \ll \langle id, y \succ x \rangle)$
$\mathbf{r}_5$	$=$	$(\forall x)(\forall y)(i(i(n(x), n(y)), i(y, x)) \rightsquigarrow true \ll \epsilon)$
$\mathbf{r}_6$	$=$	$(\forall x)(i(i(n(x), n(true)), x) \rightsquigarrow true \ll \epsilon)$
$\mathbf{r}_{15}$	$=$	$(\forall x)(\forall y)(i(x, i(i(x, y), y)) \rightsquigarrow true \ll \epsilon)$
$\mathbf{r}_{17}$	$=$	$(\forall x)(i(x, x) \rightsquigarrow true \ll \epsilon)$
$\mathbf{r}_{46}$	$=$	$(\forall x)(i(x, true) \rightsquigarrow true \ll \epsilon)$
$\mathbf{r}_{67}$	$=$	$(\forall x)(\forall y)(i(y, i(x, y)) \rightsquigarrow true \ll \epsilon)$

**Tabelle 7.16:** Das von der Implementierung generierte, neun Regeln umfassende Reduktionssystem, mit dem der Beweis der Theoreme  $L_1$ ,  $L_3$  und  $L_4$  über Lukasiewicz Logiken gelingt.

## 7.4 Pelletiers Probleme

[Pelletier, 1986] enthält eine Sammlung vieler verschiedener Probleme zum Testen automatischer Theorembeweiser. Tabelle 7.18 enthält Statistiken von Beweisen für Pelletiers 48., 49., 56., 58. und 61. Problem.<sup>9</sup> Die angegebenen Daten zur Vervollständigung und Normalisierung sind Durchschnittswerte für alle zum Abschluß des jeweiligen Tableaus gelösten  $E$ -Unifikationsprobleme. Für die Probleme 51–55 wird kein Beweis gefunden; bei diesen tritt das Problem auf, daß zwar jeder Ast für

<sup>8</sup> Die Bezeichnung der Theoreme ist die gleiche wie in [Anantharaman & Bonacina, 1989]. Diese und eine Reihe weiterer Theoreme über die Lukasiewicz Logiken tauchen als Lemmata in einem schrittweise durchgeführten automatischen Beweis für Lukasiewicz' Fünfte Vermutung auf, daß nämlich  $(\forall x)(\forall y)(or(i(x, y), i(y, x)) \approx true)$  ein Theorem ist, falls das das Funktionssymbol  $or$  definierende Axiom  $(\forall x)(\forall y)(or(x, y) \approx i(i(x, y), y))$  hinzugenommen wird.

<sup>9</sup> Pelletier stufte die Probleme nach ihrer (von ihm subjektiv beurteilten) Schwierigkeit auf einer Punkteskala von 1 (einfach) bis 10 (schwierig) ein. Diese Einstufung ist in die Statistiken aufgenommen.

Vervollständigung		Kritisches Paar	9
Anwendungen	12	Komposition	3
von möglichen	124	Vereinfachung	0
Regeln in $\mathcal{R}^0$	8	Löschung	15
Regeln in $\mathcal{R}^n$	11	Subsumtion	3
Laufzeit [s]	2,9	Probleme	1

(b)

Vervollständigung		Kritisches Paar	13
Anwendungen	17	Komposition	3
von möglichen	137	Vereinfachung	1
Regeln in $\mathcal{R}^0$	8	Löschung	21
Regeln in $\mathcal{R}^n$	12	Subsumtion	4
Laufzeit [s]	3,5	Probleme	1

(a)

Vervollständigung		Kritisches Paar	48
Anwendungen	72	Komposition	22
von möglichen	453	Vereinfachung	2
Regeln in $\mathcal{R}^0$	8	Löschung	83
Regeln in $\mathcal{R}^n$	9	Subsumtion	34
Laufzeit [s]	12,0	Probleme	1

(c)

**Tabelle 7.17:** Statistiken für Beweise der Theoreme (a)  $L_1$ , (b)  $L_3$  und (c)  $L_4$  über Lukasiewicz Logiken.

sich ohne weiteres geschlossen werden kann, bei der Suche nach einer das gesamte Tableau abschließenden Substitution aber Backtracking auftritt, und der Suchraum zu groß wird. Die übrigen der Probleme 1 bis 61 in [Pelletier, 1986] stammen nicht aus der Prädikatenlogik *mit Gleichheit*.

Vervollständigung	Kritisches Paar	0	Normalisierung		
Anwendungen	Komposition	0,3	Anwendungen	2	Ableitung
von möglichen	Vereinfachung	0	von möglichen	2,7	Vereinfachung
Regeln in $\mathcal{R}^0$	Löschung	2,7			Löschung
Regeln in $\mathcal{R}^n$	Subsumtion	0			Subsumtion
Laufzeit [s]	0,35	Probleme	3	Backtr.	0
		Punkte	3	Zeit ${}_3T^{\mathcal{AP}}$ [s]	0,33

(a)

Vervollständigung	Kritisches Paar	0,1	Normalisierung		
Anwendungen	Komposition	0,2	Anwendungen	3,9	Ableitung
von möglichen	Vereinfachung	0,1	von möglichen	5,8	Vereinfachung
Regeln in $\mathcal{R}^0$	Löschung	1,6			Löschung
Regeln in $\mathcal{R}^n$	Subsumtion	0			Subsumtion
Laufzeit [s]	3,3	Probleme	13	Backtr.	4
		Punkte	5	Zeit ${}_3T^{\mathcal{AP}}$ [s]	0,6

(b)

Vervollständigung	Kritisches Paar	0	Normalisierung		
Anwendungen	Komposition	0	Anwendungen	0,5	Ableitung
von möglichen	Vereinfachung	0	von möglichen	0,5	Vereinfachung
Regeln in $\mathcal{R}^0$	Löschung	1			Löschung
Regeln in $\mathcal{R}^n$	Subsumtion	0			Subsumtion
Laufzeit [s]	0,15	Probleme	2	Backtr.	0
		Punkte	4	Zeit ${}_3T^{\mathcal{AP}}$ [s]	0,3

(c)

Vervollständigung	Kritisches Paar	0	Normalisierung		
Anwendungen	Komposition	0	Anwendungen	3	Ableitung
von möglichen	Vereinfachung	0	von möglichen	4	Vereinfachung
Regeln in $\mathcal{R}^0$	Löschung	0			Löschung
Regeln in $\mathcal{R}^n$	Subsumtion	0			Subsumtion
Laufzeit [s]	0,25	Probleme	1	Backtr.	0
		Punkte	3	Zeit ${}_3T^{\mathcal{AP}}$ [s]	0,23

(d)

Vervollständigung	Kritisches Paar	0	Normalisierung		
Anwendungen	Komposition	0	Anwendungen	0,5	Ableitung
von möglichen	Vereinfachung	0	von möglichen	0,5	Vereinfachung
Regeln in $\mathcal{R}^0$	Löschung	1			Löschung
Regeln in $\mathcal{R}^n$	Subsumtion	0			Subsumtion
Laufzeit [s]	0,17	Probleme	2	Backtr.	0
		Punkte	6	Zeit ${}_3T^{\mathcal{AP}}$ [s]	0,28

(e)

**Tabelle 7.18:** Statistiken für die Beweise von: (a) Pelletiers 48. Problem; (b) Pelletiers 49. Problem; (c) Pelletiers 56. Problem; (d) Pelletiers 58. Problem; (e) Pelletiers 61. Problem.

# 8 Fazit und Ausblick

---

*Any technology distinguishable from magic  
is insufficiently advanced.*

— ALAN BRAGGINS

## 8.1 Fazit

Mit der Reduktion der Gleichheitsbehandlung im Tableaukalkül (und auch anderen Gentzen-artigen Kalkülen) auf die gemischte  $E$ -Unifikation ist die Suche nach einem effizienten Verfahren zwar nicht beendet, es ist aber geklärt, wo genau die Schwierigkeiten liegen, und welches die zu lösenden Probleme sind.

Das in Kapitel 4 vorgestellte Verfahren für die gemischte  $E$ -Unifikation ist zumindest ein vielversprechender Ansatz, diese Probleme zu meistern. Zugleich ist es das erste vervollständigungsbasierte Verfahren für die gemischte  $E$ -Unifikation überhaupt und das erste *implementierte* vervollständigungsbasierte Verfahren für die rein starre  $E$ -Unifikation.

Wie die Beispiele in Abschnitt 7 zeigen, ist das Verfahren durchaus auch für rein universelle Probleme geeignet und in manchen Fällen sogar dem Unfailing Knuth-Bendix-Algorithmus überlegen.

Als Methode für die Gleichheitsbehandlung im Tableaukalkül ist die Suche nach Lösungen gemischter  $E$ -Unifikationsprobleme den bisher verwendeten Verfahren überlegen, wenn auch die Implementierung noch daran krankt, daß die Äste nacheinander geschlossen werden. Es zeigt sich, daß die Suche nach dem Abschluß eines Tableaus praktisch niemals daran scheitert, daß ein einzelner Ast nicht geschlossen werden könnte, sondern immer daran, daß die Suche nach *einer* alle Äste zugleich abschließenden Substitution zu aufwendig ist.

## 8.2 Mögliche Verbesserungen

Es wäre sicherlich sinnvoll und würde die Effizienz der Implementierung entscheidend verbessern, wenn die Äste des Tableaus nicht nacheinander sondern parallel betrachtet würden. Die gesamte in den verschiedenen Ästen enthaltene Information würde zugleich zur Verfügung stehen. Damit könnte unnötiges Backtracking vermieden und der Suchraum deutlich eingeschränkt werden. Beispielsweise ist es häufig möglich, zu erkennen, daß ein Ast nur einen Abschluß hat. Dieser könnte sofort ausgeführt werden.

Eine weitere sinnvolle Erweiterung wäre, die Gleichungen eines Astes — zumindest teilweise — schon während des Tableaubaufbaus zu vervollständigen. Wenn eine Verzweigung auftritt, wäre es dann möglich, die schon berechnete Vervollständigung auf beide neuen Äste zu übertragen. Zumindest bei Ästen, die zum großen Teil übereinstimmen, wird auch die Gleichungsmenge weitgehend dieselbe sein, und die Vervollständigung der gemeinsamen Gleichungen müßte so nur einmal berechnet werden.

Weitere Verbesserungen wären sicher auch durch die Verwendung anderer Reduktionsordnungen und anderer Kriterien für die Auswahl des jeweils nächsten Vervollständigungs- oder Normalisierungsschrittes zu erzielen. Sie könnten auf bestimmte Problemklassen zugeschnitten werden.

# Literaturverzeichnis

---

- [Anantharaman & Bonacina, 1989] Siva Anantharaman & Maria Paola Bonacina. Automated proofs in Lukasiewicz logic. Technical report, Department of Computer Science, SUNY at Stony Brook, New York, November 1989.
- [Bachmair *et al.*, 1986] Leo Bachmair, Nachum Dershowitz, & Jieh Hsiang. Orderings for equational proofs. In *Proceedings, Symposium on Logic in Computer Science (LICS), Boston*, pages 346–357, 1986.
- [Bachmair *et al.*, 1989] Leo Bachmair, Nachum Dershowitz, & David A. Plaisted. Completion without failure. In H. Ait-Kaci & M. Nivat, editors, *Resolution of Equations in Algebraic Structures, Volume 2*, chapter 1. Academic Press, 1989.
- [Beckert & Hähnle, 1992] Bernhard Beckert & Reiner Hähnle. An improved method for adding equality to free variable semantic tableau. In D. Kapur, editor, *Proceedings, 11th Conference on Automated Deduction (CADE), Albany/NY*, volume 607 of *LNCS*, pages 507–521. Springer, 1992.
- [Beckert *et al.*, 1993] Bernhard Beckert, Reiner Hähnle, & Peter H. Schmitt. The even more liberalized  $\delta$ -rule in free variable semantic tableaux. In *Proceedings, 3rd Kurt Gödel Colloquium (KGC), Brno, Czech Republic*, volume 713 of *LNCS*. Springer, 1993. To appear.
- [Beckert, 1991] Bernhard Beckert. Konzeption und Implementierung von Gleichheit für einen tableau-basierten Theorembeweiser. IKBS Report 208, Science Center, Institute for Knowledge Based Systems, IBM Germany, 1991. Zusammenfassung in [Beckert & Hähnle, 1992].
- [Beckert, 1993] Bernhard Beckert. A completion-based method for adding equality to free variable semantic tableaux. In D. Basin, R. Hähnle, B. Fronhöfer, J. Posegga, & C. Schwind, editors, *Proceedings, 2nd Workshop on Theorem Proving with Analytic Tableaux and Related Methods, Marseille/France, Saarbrücken*, MPI-I-92-213, March 1993. Max-Planck-Institut für Informatik.
- [Browne, 1988] Randall Jeffrey Browne. Ground term rewriting in semantic tableaux systems for first-order logic with equality. Technical Report UMIACS-TR-88-44, College Park, MD, 1988.
- [Comon, 1990] Hubert Comon. Solving inequations in term algebras. In *Proceedings, Fifth Annual IEEE Symposium on Logic in Computer Science (LICS), Philadelphia, PA*. IEEE Computer Society Press, 1990.
- [Dershowitz & Jouannaud, 1990] Nachum Dershowitz & Jean-Pierre Jouannaud. *Handbook of Theoretical Computer Science*, volume B, chapter 6: Rewrite Systems, pages 244–320. Elsevier Science Publishers North-Holland, 1990.
- [Dershowitz, 1987] Nachum Dershowitz. Termination of rewriting. *Journal of Symbolic Computation*, 3(1):69–115, November 1987.

- [Dershowitz, 1989] Nachum Dershowitz. Completion and its applications. In H. Aït-Kaci & M. Nivat, editors, *Resolution of Equations in Algebraic Structures, Volume 2*, chapter 2. Academic Press, 1989.
- [Fitting, 1990] Melvin C. Fitting. *First-Order Logic and Automated Theorem Proving*. Springer, 1990.
- [Gallier *et al.*, 1987] Jean H. Gallier, Stan Raatz, & Wayne Snyder. Theorem proving using rigid  $E$ -unification, equational matings. In *Proceedings, Symposium on Logic on Logic in Computer Science (LICS), Ithaca/NY*, pages 338–346. The Computer Society of the IEEE, 1987.
- [Gallier *et al.*, 1988] Jean H. Gallier, Wayne Snyder, Paliath Narendran, & David A. Plaisted. Rigid  $E$ -unification is NP-complete. In *Proceedings, Third Annual Symposium on Logic in Computer Science (LICS), Edinburgh*, pages 218–227. IEEE Computer Society, 1988.
- [Gallier *et al.*, 1990] Jean H. Gallier, Paliath Narendran, David A. Plaisted, & Wayne Snyder. Rigid  $E$ -unification: NP-completeness and application to equational matings. *Information and Computation*, pages 129–195, 1990.
- [Gallier *et al.*, 1992] Jean H. Gallier, Paliath Narendran, Stan Raatz, & Wayne Snyder. Theorem proving using equational matings and rigid  $E$ -unification. *Journal of the ACM*, 39(2):377–429, April 1992.
- [Göbel, 1987] Richard Göbel. Ground confluence. In P. Lescanne, editor, *International Conference on Rewriting Techniques and Applications (RTA), Bordeaux*, volume 256 of *LNCS*, pages 156–167. Springer, 1987.
- [Goubault, 1993] Jean Goubault. A rule-based algorithm for rigid  $E$ -unification. Unpublished, February 1993.
- [Hähnle & Schmitt, 1993] Reiner Hähnle & Peter H. Schmitt. The liberalized  $\delta$ -rule in free variable semantic tableaux. *Journal of Automated Reasoning*, 1993. To appear.
- [Hähnle *et al.*, 1992] Reiner Hähnle, Bernhard Beckert, Stefan Gerberding, & Werner Kernig. The many-valued tableau-based theorem prover  $\mathcal{Z}^{AP}$ . IKBS Report, Science Center, Institute for Knowledge Based Systems, IBM Germany, 1992.
- [Hähnle, 1993] Reiner Hähnle. *Automated Proof Search in Multiple Valued Logics*. Oxford University Press, forthcoming, 1993.
- [Herrman *et al.*, 1991] M. Herrman, Claude Kirchner, & Hélène Kirchner. Implementation of term rewriting systems. *Computer Journal*, 34(1):20–33, 1991.
- [Hsiang & Mzali, 1988] Jieh Hsiang & Jalel Mzali. SbREVE user's guide. Technical report, LRI, Université de Paris-Sud, 1988.
- [Jeffrey, 1967] Richard C. Jeffrey. *Formal Logic: Its Scope and Limits*. McGraw Hill, 1967.
- [Kaplan & Rémy, 1989] Stephane Kaplan & Jean-Luc Rémy. Completion algorithms for conditional rewriting systems. In H. Aït-Kaci & M. Nivat, editors, *Resolution of Equations in Algebraic Structures, Volume 2*, chapter 5. Academic Press, 1989.
- [Kirchner, 1990] Claude Kirchner, editor. *Unification*. Academic Press, 1990.

- [Knuth & Bendix, 1970] Donald E. Knuth & P. B. Bendix. Simple word problems in universal algebras. In J. Leech, editor, *Computational Problems in Abstract Algebras*, pages 263–297. Pergamon Press, Oxford, 1970.
- [Lipson, 1981] John D. Lipson. *Elements of Algebra and Algebraic Computing*. Benjamin/Cummings Publishing Company, 1981.
- [Martelli & Montanari, 1982] A. Martelli & U. Montanari. An efficient unification algorithm. *ACM Transactions on Programming Languages and Systems*, 4(2):258–282, April 1982.
- [Nelson & Oppen, 1980] G. Nelson & D. C. Oppen. Fast decision procedures based on congruence closure. *Journal of the ACM*, 27(2):356–364, April 1980.
- [Nutt *et al.*, 1989] Werner Nutt, P. Réty, & Gert Smolka. Basic narrowing revisited. *Journal of Symbolic Computation*, 7(3/4):295–318, 1989.
- [Pelletier, 1986] Francis Jeffrey Pelletier. Seventy-five problems for testing automatic theorem provers. *Journal of Automated Reasoning*, 2:191–216, 1986.
- [Petermann, 1993] Uwe Petermann. A framework for integrating equality reasoning into the extension procedure. In D. Basin, R. Hähnle, B. Fronhöfer, J. Posegga, & C. Schwind, editors, *Proceedings, 2nd Workshop on Theorem Proving with Analytic Tableaux and Related Methods, Marseille/France*, Saarbrücken, MPI-I-92-213, March 1993. Max-Planck-Institut für Informatik.
- [Peterson, 1990] Gerald E. Peterson. Complete sets of reductions with constraints. In M. E. Stickel, editor, *Proceedings, 10th International Conference on Automated Deduction (CADE)*, Kaiserslautern, volume 449 of LNCS, pages 381–395. Springer, 1990.
- [Poplestone, 1967] R. J. Poplestone. Beth-tree methods in automatic theorem proving. In *Machine Intelligence*, volume 1, pages 31–46. Oliver and Boyd, 1967.
- [Reeves, 1987] Steve V. Reeves. Adding equality to semantic tableau. *Journal of Automated Reasoning*, 3:225–246, 1987.
- [Shostak, 1978] Robert E. Shostak. An algorithm for reasoning about equality. *Communications of the ACM*, 21(7):583–585, July 1978.
- [Siekman, 1989] Jörg H. Siekman. Universal unification. *Journal of Symbolic Computation*, 7(3/4):207–274, 1989. Auch in [Kirchner, 1990]; frühere Version in *Proceedings, 7th International Conference on Automated Deduction (CADE)*, Napa/FL, volume 170 of LNCS, Springer, 1984.
- [Smullyan, 1968] Raymond Smullyan. *First-Order Logic*. Springer, 1968.



# Symbolverzeichnis

---

KIRK: *Analysis, Mr. Spock?*  
SPOCK: *Captain, it doesn't appear  
in the symbol table.*  
KIRK: *Then it's of external origin?*  
SPOCK: *Affirmative.*  
— STARTREK

Im folgenden sind die verwendeten Symbole beschrieben. In der Regel ist jeweils die Seite angegeben, auf der ein Symbol zuerst auftritt, und falls es in einer Definition eingeführt wurde, die Nummer dieser Definition.

$\alpha, \beta, \gamma, \delta$	Tableauregel–Schemata, S. 100
$\epsilon$	Der leere Constraint, Def. 4.2.6, S. 47
$\Phi, \Psi$	Mengen von (signierten oder unsignierten) Formeln, S. 5
$\phi, \psi$	Signierte Formeln, S. 101
$\sigma _V$	Einschränkung der Substitution $\sigma$ auf die Menge der Variablen $V$ , Def. 2.3.1, S. 8
$\sigma \circ \tau$	Komposition der Substitutionen $\sigma$ und $\tau$ , Def. 2.3.1, S. 8
$\sigma, \tau, \mu, \nu$	Substitutionen, Def. 2.3.1, S. 7
$t\sigma$	Anwendung der Substitution $\sigma$ , Def. 2.3.4, S. 9
$\Upsilon$	Methode zum Auffinden universeller Formeln, Def. 5.1.12, S. 107
$[t]_E$	Äquivalenzklasse von $t$ bezüglich $\overset{*}{\leftrightarrow}_E$ , Def. 3.5.1, S. 26
$\langle t \rangle_E$	Menge, die alle Äquivalenzklassen von $t$ darstellt, Def. 3.5.5, S. 28
$\langle \sigma, O \rangle$	Constraint, Def. 4.2.6, S. 46
$\langle p_1, \dots, p_n \rangle$	Position, Def. 2.3.16, S. 12
$\langle \rangle$	Leere Position, Def. 2.3.16, S. 12
$p \circ p'$	Verkettung von Positionen, Def. 2.3.16, S. 12
$\Phi \models F$	$F$ folgerbar aus $\Phi$ , Def. 2.2.4, S. 6
$\Phi \cong F$	$F$ folgerbar aus $\Phi$ in der Prädikatenlogik mit Gleichheit, Def. 2.2.4, S. 6
$\Phi \equiv F$	$F$ starr folgerbar aus $\Phi$ , Def. 2.2.5, S. 6
$\Phi \cong\equiv F$	$F$ starr folgerbar aus $\Phi$ in der Prädikatenlogik mit Gleichheit, Def. 2.2.5, S. 7
$\models F$	$F$ ist Tautologie, Def. 2.2.4, S. 6
$\cong F$	$F$ ist Tautologie der Prädikatenlogik mit Gleichheit, Def. 2.2.4, S. 6
$\mathcal{M} \models F$	$\mathcal{M}$ Modell von $F$ , Def. 2.2.4, S. 6
$\mathcal{M} \cong F$	$\mathcal{M}$ normales Modell von $F$ , Def. 2.2.4, S. 6
$(\forall \bar{x})$	Kurzschreibweise für $(\forall x_1) \dots (\forall x_n)$ , Def. 3.1.1, S. 13

$\sim$	Ableitbarkeitsrelation der Objektebene, Def. 3.6.1, S. 30
$\forall, \exists$	Quantoren, S. 5
$\approx$	Gleichheitssymbol der Objektebene, S. 5
$\neg, \wedge, \vee, \supset$	Logische Operatoren, S. 5
$\leq$	Subsumtionsrelation auf der Menge der Constraints, Def. 4.2.8, S. 47
$\leq$	Spezialisierungsrelation auf den Substitutionen, Def. 2.3.6, S. 9
$\triangleright$	„Ein Unterterm ist Instanz von“-Relation, Def. 2.3.18, S. 12
$\succ_F$	Die eine lexikographische Pfadordnung induzierende Ordnung auf den Funktionssymbolen, Def. 3.6.7, S. 33
$\vdash_{\text{UKBA}}$	Transformationsrelation des UKBA auf rein universellen Reduktionssystemen, Def. 3.6.13, S. 36
$\succ_X$	Ordnung auf $\mathcal{R}$ - $\sigma$ -Beweisen, Def. 4.8.7, S. 78
$\succ_{\text{LPO}}$	Durch $\succ_{\text{LPO}}$ lexikographisch induzierte Ordnung auf Tupeln von Termen, Def. 3.6.7, S. 33
$\succ_{(E,s,t)}$	Ordnungsrelation auf Grundsubstitutionen, Def. 4.8.10, S. 81
$\succ_{\text{LPO}}$	Lexikographische Pfadordnung, Def. 3.6.7, S. 33
$\succ_X$	Die zur Interpretation des Prädikatsymbols $\succ$ in den Ordnungsbedingungen verwendete Reduktionsordnung, Def. 4.2.2, S. 44
$\sqsubseteq_E$	(Subsumtions-)Relation auf den Substitutionen, Def. 3.3.2, S. 22
$\leq_E$	(Subsumtions-)Relation auf den Substitutionen, Def. 3.3.2, S. 22
$s \succ t$	Atomare Ordnungsbedingung, Def. 4.2.1, S. 44
$\Rightarrow_{\mathcal{R}}$	„Umkehrbare“ Ableitbarkeitsrelation auf Termen mit Constraint, Def. 4.3.6, S. 54
$\Rightarrow_{\mathcal{R}}$	Ableitbarkeitsrelation auf Substitutionen, Def. 4.10.2, S. 96
$\Rightarrow_{\mathcal{R}}$	Ableitbarkeitsrelation auf Termen mit Constraint, Def. 4.3.6, S. 54
$\rightarrow_{\mathcal{R}}$	Hilfswise zum Beweis definierte Ableitbarkeitsrelation auf Termen mit Constraint, Def. 4.8.1, S. 75
$\leftarrow$	Inverse der Relation $\rightarrow$ , Def. 2.1.2, S. 3
$\rightarrow_R$	Ableitbarkeitsrelation auf Termen bezüglich des Reduktionssystems $R$ , Def. 3.6.3, S. 31
$\leftrightarrow_E$	Ableitbarkeitsrelation auf Termen bezüglich der Gleichungsmenge $E$ , Def. 3.2.1, S. 16
$\rightarrow^*$	Transitiver und reflexiver Abschluß der Relation $\rightarrow$ , Def. 2.1.2, S. 3
$\leftrightarrow^*$	Transitiver, reflexiver und symmetrischer Abschluß der Relation $\rightarrow$ , Def. 2.1.2, S. 3
$\leftrightarrow$	Symmetrischer Abschluß der Relation $\rightarrow$ , Def. 2.1.2, S. 3
$\vdash$	Transitiver Abschluß der Relation $\rightarrow$ , Def. 2.1.2, S. 3
$\sim$	Äquivalenzrelation auf den Substitutionen, Def. 2.3.8, S. 9
$\sim$	Äquivalenzrelation auf der Menge der Constraints, Def. 4.2.8, S. 47
$\top$	Top-Element im Verband der Substitutionen modulo $\sim$ , S. 10
$\vdash$	Transformationsrelation auf den Reduktionssystemen, Def. 4.4.9, S. 60
$\vdash_{\mathcal{R}}$	Normalisierungsrelation auf Mengen von Termen mit Constraint bzgl. des Reduktionssystems $\mathcal{R}$ , Def. 4.5.3, S. 63
$\sqcup$	Kombination von Constraints, Def. 4.2.9, S. 48
$\sqcup$	Vereinigung von Substitutionen, Def. 2.3.12, S. 11

$\sqcup$	Supremum (im Verband der Substitutionen modulo $\sim$ ), S. 10
$\ll$	Der einen Constraint an einen Term knüpfende „wenn“-Operator, Def. 4.3.1, S. 50
$x/t$	Instantiierung der Variablen $x$ mit dem Term $t$ , Def. 2.3.1, S. 8
$A$	Ast eines Tableaus, S. 103
$a, b, c, d$	Konstantensymbole, S. 5
$\text{Bound}(G)$	Menge der in $G$ gebunden auftretenden Variablen, Def. 2.2.1, S. 5
$\mathcal{C}(\langle E, s, t \rangle)$	Menge von durch Vermischung von Vervollständigung und Normalisierung berechneten Constraints, Def. 4.6.6, S. 68
$\mathcal{C}^i(\langle E, s, t \rangle)$	Mengen von durch Vermischung von Vervollständigung und Normalisierung berechneten Constraints, die $\mathcal{C}(\langle E, s, t \rangle)$ approximieren, Def. 4.6.6, S. 68
$\mathcal{C}_i\langle E, s, t \rangle$	Mengen von Constraints, die von Lösungen des Problems $\langle E, s, t \rangle$ erfüllt werden, Def. 4.6.1, S. 65
$\mathcal{C}_i^{j,k}\langle E, s, t \rangle$	Mengen von Constraints, die $\mathcal{C}_i(\langle E, s, t \rangle)$ approximieren, Def. 4.6.1, S. 65
$c^{-1}$	Negation des Constraints $c$ , Def. 4.2.9, S. 47
$c$	Constraint, S. 46
$\text{Dom}(\sigma)$	Definitionsbereich der Substitution $\sigma$ , Def. 2.3.1, S. 7
$\mathcal{D}$	Grundmenge eines Modells, S. 5
$E(A)$	Menge der Gleichungen des Astes $A$ , Def. 5.2.3, S. 114
$E$	Menge von Gleichungen, S. 14
false	Falsche Ordnungsbedingung, Def. 4.2.1, S. 44
$\mathcal{F}$	Menge der Funktionssymbole, S. 5
<b>Form*</b>	Menge der signierten Formeln, Def. 5.1.1, S. 101
<b>Form</b>	Menge der Formeln, S. 5
$\text{Free}(G)$	Menge der in $G$ frei auftretenden Variablen, Def. 2.2.1, S. 5
$F, G$	Formeln, S. 5
<b>F</b>	Vorzeichen für signierte Formeln (falsch), S. 100
$f, g$	Funktionssymbole, S. 5
$\mathcal{I}$	Interpretation eines Modells, S. 5
$id$	Leere Substitution, S. 8
$\mathcal{K}$	Menge der Konstantensymbole, S. 5
$l, r$	Term, der die linke (bzw. rechte) Seite einer Gleichung oder einer Reduktionsregel ist, S. 13
$\mathcal{L}$	Menge der Lösungen eines $E$ -Unifikationsproblems, S. 21
$\mathcal{M}$	Modell, S. 5
$O$	Ordnungsbedingung, S. 44
$\mathcal{P}(A)$	Menge des Unifikationsprobleme des Astes $A$ , Def. 5.2.5, S. 116
$\mathcal{P}$	Menge der Prädikatensymbole, S. 5
$\mathcal{P}(A)$	Menge der Unifikationsprobleme des Astes $A$ , Def. 5.2.5, S. 115
$\text{Range}(\sigma)$	Wertebereich der Substitution $\sigma$ , Def. 2.3.1, S. 8
$\mathcal{R}^\infty$	Vervollständigung eines Reduktionssystems, Def. 4.4.9, S. 60
$\mathcal{R}^-$	Inverse des Reduktionssystems $\mathcal{R}$ , Def. 4.7.1, S. 69

---

$\mathcal{R}^\pm$	Vereinigung des Reduktionssystems $\mathcal{R}$ mit seiner Inversen $\mathcal{R}^-$ , Def. 4.7.1, S. 69
$\mathcal{R}$	Reduktionssystem mit Constraints, S. 52
$\mathbf{r}^-$	Inverse der Reduktionsregel $\mathbf{r}$ , Def. 4.7.1, S. 69
$\mathbf{r}$	Reduktionsregel mit Constraint, Def. 4.3.1, S. 50
$\mathcal{R}^0$	Das initiale Reduktionssystem zu einer Menge von Gleichungen, Def. 4.3.3, S. 52
$\text{Sat}^*(c)$	Aufzählbare Menge von $c$ erfüllenden Substitutionen, Def. 4.2.13, S. 49
$\text{Sat}(c)$	Menge der $c$ erfüllenden Substitutionen, Def. 4.2.6, S. 47
$\mathcal{R}\langle\sigma\rangle$	Grundinstanz eines Reduktionssystems, Def. 4.3.4, S. 52
$\mathcal{S}$	Signatur, S. 5
$\mathcal{S}, \mathcal{T}$	Mengen von Termen mit Constraint, S. 52
<b>Subst</b> $\sim$	Menge der Substitutionen modulo $\sim$ , Def. 2.3.10, S. 10
<b>Subst</b> *	Menge der idempotenten Substitutionen, Def. 2.3.2, S. 8
<b>Subst</b>	Menge der Substitutionen, Def. 2.3.1, S. 7
$s, t$	Terme, S. 5
$\mathcal{T}[\sigma]$	Instanz der Menge $\mathcal{T}$ von Termen mit Constraint, Def. 4.3.4, S. 52
<b>Term</b>	Menge der Terme, S. 5
true	Wahre Ordnungsbedingung, Def. 4.2.1, S. 44
$t _p$	Unterterm von $t$ an der Position $p$ , Def. 2.3.16, S. 12
$\mathbf{t}$	Term mit Constraint, Def. 4.3.1, S. 50
$t[p/s]$	Term $t$ mit $s$ eingesetzt an Position $p$ , Def. 2.3.16, S. 12
$T$	Tableau, S. 103
$\top$	Vorzeichen für signierte Formeln (wahr), S. 100
$\hat{\mathcal{U}}(\langle E, s, t \rangle)$	Menge von Unifikatoren, die bezüglich $\leq$ vollständig ist, S. 96
$\vec{\mathcal{U}}$	Menge von Lösungen eines simultanen Unifikationsproblems, S. 98
$\mathcal{U}'_\infty(\langle E, s, t \rangle)$	Menge von Lösungen des Problems $\langle E, s, t \rangle$ , S. 66
$\mathcal{U}_\infty(\langle E, s, t \rangle)$	Menge von Lösungen des Problems $\langle E, s, t \rangle$ , S. 65
$\mathcal{U}$	Vollständige Menge allgemeinster Unifikatoren, Def. 3.3.1, S. 21
$\text{Var}(G)$	Menge der in $G$ auftretenden Variablen, Def. 2.2.1, S. 5
$\mathcal{V}$	Menge der Variablen, S. 5
$x, y, z$	Variablen, S. 5