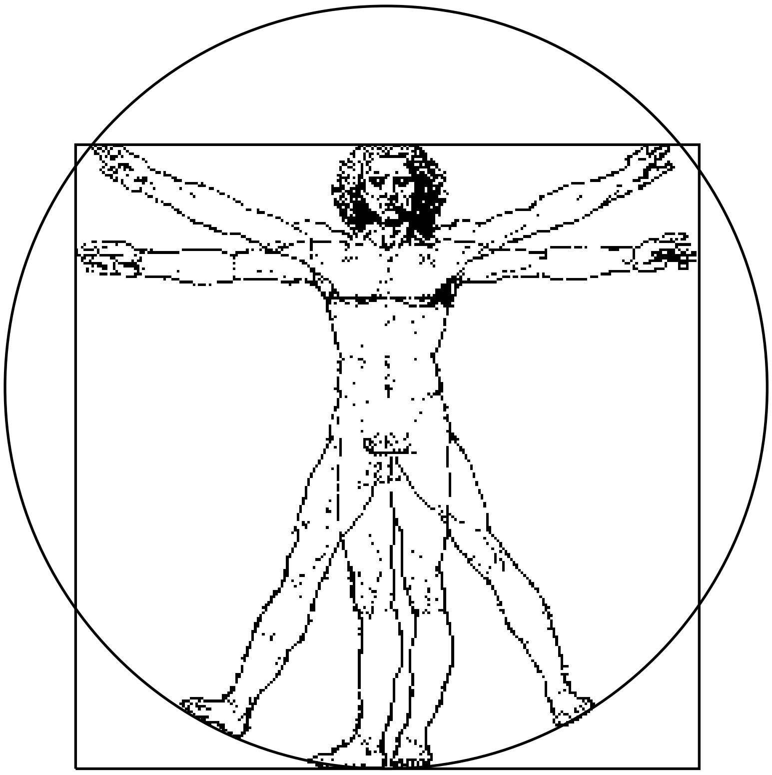
Heuristiken zum Beweisen von Gleichungen

Diplomarbeit von Serge Autexier



Angefertigt nach einem Thema von Prof. Dr. Jörg Siekmann am
Fachbereich 14, Informatik, der Universität des Saarlandes, Saarbrücken.

8. Mai 1996

Hiermit erkläre ich an Eides Statt, daß ich diese Arbeit nur unter der Verwendung der angegebenen Quellen und Hilfsmitteln angefertigt habe.

Saarbrücken, den 8. Mai 1996

A Mamy ...

Inhaltsverzeichnis

Kapitel 1

Einleitung

Der Begriff des Beweisens ist untrennbar damit verbunden, Aussagen treffen zu können, die in einem bestimmten Rahmen wahr sind. Die erste Definition dessen, was einen Beweis ausmacht, findet sich im *Organon* des Aristoteles. Dort heißt es in der *Lehre vom Beweis*, daß ein Beweis auf der Verwendung von Schlußregeln aufbaut. Die Entdeckung solcher allgemeinen Schlußregeln, wie zum Beispiel der *Syllogismus*, ist Aristoteles' Verdienst und dadurch wurde er zum Begründer der formalen Logik:

> »Bei der Reflexion auf die dem Argumentieren zugrundeliegenden Strukturen entdeckte er, daß es Argumentationen gibt, über deren Triftigkeit man ganz unabhängig von ihrem Inhalt allein auf Grund ihrer Form entscheiden kann.«
> (C. M. Wieland, 1733-1833), aus [Hel92].

Diese Eigenschaften des logischen Schließens griff später Leibniz auf, um seine Idee einer neuen Logik zur Mechanisierung und Automatisierung des logischen Schließens aufzustellen ([Lei86]). Dieser Gedanke erfuhr durch Boole (1815-1864) eine erste Konkretisierung in der von ihm entwickelten Boole'schen Logik. Anschließend dauerte es fast 200 Jahre bis der Leibniz'sche Traum durch die Arbeit von Frege [Fre78] eine weitere Konkretisierung erfuhr. Frege entwickelte in seiner "Begriffsschrift" eine formale Logik, die mächtiger war, als die von Boole. Allerdings ließen sich in dieser Logik Antinomien beschreiben, die durch Russell entdeckt wurden. Russel und Whitehead gelang es in ihrer "Principia Mathematica" [RW10], diese Paradoxien zu eliminieren. Dafür belegten sie jedes Objekt mit einem Typ und konnten die Termkonstruktion auf wohlgetypte Terme einschränken. Die von Whitehead und Russel definierte Logik ist heute bekannt als Prädikatenlogik.

Bis dahin schien es, als ließe sich der Traum von Leibniz realisieren. Dieser Traum wurde durch zwei Ereignisse zunichte gemacht: Das erste Ereignis war

Gödel's Beweis, daß für jedes logische System, in dem sich mindestens die Peano-Arithmetik [Pea89] kodieren läßt, kein vollständiger Kalkül angegeben werden kann [Göd31]. Dies bedeutet, daß für ein solches logisches System keine endliche Menge von Schlußregeln angegeben werden kann, so daß alle erfüllbaren Sätze auch tatsächlich abgeleitet werden können. Das zweite Ereignis waren die Beweise von Church [Chu36] und Turing [Tur37], daß schon die Einschränkung der Prädikatenlogik auf Formeln, in denen nur über Variablen erster Ordnung quantifiziert werden darf, prinzipiell unentscheidbar ist. Gerettet wurde ein Teil des Leibniz'schen Traumes durch die Arbeit von Herbrand [Her30]: Herbrand bewies, daß es für die Prädikatenlogik zumindest Kalküle gibt, so daß aus jedem unerfüllbaren Satz ein Widerspruch mit Hilfe dieses Kalküls abgeleitet werden kann. Der erste widerlegungsvollständige Kalkül, der erfolgreich auf einem Rechner implementiert wurde, war der auf der Unifikation aufbauende Resolutionskalkül von Robinson [Rob65]. Dieser besteht nur aus einer einzigen Schlußregel, der Resolutionsregel, bei der es sich um eine Verallgemeinerung des einführends erwähnten Syllogismus handelt. Die Unifikation löst das Problem, eine Instantiierung der Variablen zweier Terme zu finden, so daß die beiden Terme gleich werden. Von da an begann man damit automatische Beweissysteme zu entwickeln, wie zum Beispiel MKRP [OS89] oder OTTER [McC90], die alle auf dem Resolutions-Kalkül von Robinson aufbauten. Diese Systeme waren und sind in der Lage sehr komplexe Sätze zu beweisen. Dabei hat es sich allerdings gezeigt, daß sie stark abhängig sind von der Art der Axiomatisierung und hauptsächlich dann dem Menschen überlegen sind, wenn der Satz aus einem Gebiet stammt, in dem der Mensch so gut wie keine Intuition besitzt.

Parallel dazu entwickelten in den 50er Jahren Newell, Shaw und Simon ihre "Logik-Maschine" [NSS56, NSS63]. Dabei war es ihnen wichtiger, eine kognitiv adäquate Modellierung menschlichen Verhaltens zu erreichen, denn ein widerlegungsvollständiges Verfahren.

Desweiteren wurden Beweisprüfer entwickelt: Mit ihnen sollte es möglich sein, bestehende Beweise auf ihre Korrektheit hin zu überprüfen. Als erstes erfolgreiches Projekt in dieser Richtung ist das AUTOMATH-Projekt [dB80] zu zitieren, mit dem die Beweise eines mathematischen Lehrbuchs von Landau [Lan30] auf ihre Korrektheit hin überprüft wurden [Jut79]. Namhafte Systeme, die aus diesem Ansatz heraus entwickelt wurden, sind zum Beispiel LCF [GMW79], Nuprl [CAB+86], Isabelle [PN90] und KIV [HRS90, HRS91]. In den ersten Systemen dieses Typs war der Benutzer noch gezwungen, jeden einzelnen Beweisschritt anzugeben. In zunehmenden Maß wurden diese Systeme dann anwendungsfreundlicher, indem der Benutzer auf definierte *Taktiken* zurückgreifen konnte, in denen mehrere Beweisschritte zusammengefaßt sind. Gleichzeitig wurden auch automatische Beweiskomponenten integriert, die den Benutzer bei der Beweissuche unterstützen sollten.

Die Prädikatenlogik, wie sie von Russell und Whitehead definiert wurde, wurde um eine feste Semantik für das Gleichheitsprädikat erweitert. Entsprechend wurde der Resolutionskalkül von Robinson um entsprechende Gleichheits-Regeln erweitert: Der bekannteste Vertreter dieser Art von Gleichheitsregeln ist die Paramodulation [RW69]. Parallel dazu wurde für reine Gleichheitsbeweise die Theorie der sogenannten Termersetzungssysteme entwickelt. Die Idee eines Termersetzungssystems ist es, mit Hilfe einer Ordnung allen Gleichungen eine Richtung zu geben, die die Anwendungsrichtung dieser Gleichung ein für alle mal festlegt. Diese gerichteten Gleichungen bilden ein sogenanntes Termersetzungssystem. Mit Hilfe dieser Gleichungen werden die beiden Terme eines Gleichheitsproblems reduziert, bis keine Gleichung des Termersetzungssystems mehr anwendbar ist. Ein Term, auf den keine Gleichung mehr anwendbar ist, wird Normalform genannt. Sind die Normalformen beider Terme gleich, sind auch beide Terme gleich. Dabei ergeben sich zwei Probleme: Zum einen muß das Termersetzungssystem terminierend sein und zum anderen muß es für jeden Term eine *eindeutige* Normalform geben. Mit Hilfe der von Knuth und Bendix entwickelten Vervollständigung [KB70] kann ein solches Termersetzungssystem zu einem terminierenden Termersetzunsgsystem vervollständigt werden, falls dies möglich ist. Die Güte eines Termersetzungssystems hängt im wesentlichen von der gewählten Ordnung ab. Deshalb wurden immer wieder Ordnungen angegeben, wie zum Beispiel in [HO80, ZK88].

Parallel dazu hat sich in den 50er Jahren für das Beweisen von Gleichungen das Paradigam der *Differenz Reduktion* herauskristallisiert. Dieses wird generell Newell, Shaw und Simon [NSS56, NSS63] attributiert und besagt sinngemäß folgendes:

> »Für den Beweis der Gleichheit zweier Terme müssen nach bestimmten Maßstäben die Differenzen und die Gemeinsamkeiten dieser beiden Terme festgestellt werden. Der Beweis ist so zu steuern, daß die Differenzen der beiden Terme kleiner werden, die Gemeinsamkeiten größer.«

Namhafte Kalkülregeln, die nach diesem Paradigma entwickelt wurden, sind die E-Resolution [Mor69], die RuE-Resolution [Dig79] und [BS88]. Im Rahmen des induktiven Beweisens wurden später spezielle Unifikations-Algorithmen entwickelt, die die gemeinsamen Teil-Terme (sogenannte Skelette) zweier Terme berechneten: Diese *Differenz Unifikation* wurde von Basin und Walsh [BW92, BW93] entwickelt. Nach und nach wurden diverse Verfahren für die *Differenz Reduktion* entwickelt, z. B. [Prä92, Hut91].

Das Problem beim automatischen Beweisen ist der Suchraum. Deshalb geht man mehr und mehr dazu über, interaktive Beweissysteme nach dem Vorbild der oben erwähnten Beweisprüfer zu entwickeln. Durch die Benutzer-Interaktion ist man aber gezwungen, die von den automatischen Beweiskomponenten gelieferten Beweise oder Teil-Beweise so aufzubereiten, daß ein menschlicher Benutzer sie verstehen kann. In der Tat sind die meisten Beweise, die von einem widerlegungsvollständigen Verfahren geliefert werden, sehr unübersichtlich. Ansätze, solche Beweise für den Menschen aufzubereiten, sind in [Hua89, Hua95] zu finden. Ein anderer Ansatz basiert darauf, die Strategien in den Beweisern im Sinne von Newell, Shaw und Simon dem menschlichen Problemlösungsverhalten nachzuempfinden. Gleichzeitig müssen aber auch die Möglichkeiten zur Interaktion gewährleistet sein, d. h. zum Beispiel, daß der Benutzer dem System bestimmte zu erreichende Ziele vorgeben können muß. Diese beiden Aspekte implizieren die Verwendung von Abstraktionen. Abstraktion ist ein am menschlichen Problemlösungsverhalten häufig zu beobachtendes Verhalten, um eine Idee oder einen Plan dafür zu erhalten, wie der eigentliche Beweis ablaufen könnte. Deshalb ist die Abstraktion auch ein in der KI häufig verwendetes Konzept: Zum Beispiel in Problemlösungssystemen [NSS63], beim Planen [Sac74], im Theorem-Beweisen und dem logischen Programmieren [GW89, GW92b, Pla81, Pla86, Plu87], beim Lernen [UR89] u. a. Durch die Verwendung einer Abstraktion wird das Augenmerk auf die nach bestimmten Kriterien wichtigen Teile des zu lösenden Problems gelenkt.

Die Motivation für die vorliegende Arbeit ist nun die, das Prinzip der Abstraktion mit dem Paradigma der *Differenz Reduktion* zu kombinieren. Als Erprobungsfeld dieser Idee dient das reine Gleichheitsbeweisen. Die Kombination dieser beiden klassischen KI-Ansätze sieht folgendermaßen aus: Zunächst wird ein zu lösendes Gleichheitsproblem nach bestimmten Kriterien abstrahiert. Anschließend sollen die beiden abstrakten Terme mit Hilfe von *Differenz Reduktions* Techniken angeglichen werden. Der Beweis des abstrakten Gleichheitsproblems stellt dann einen Plan für den Beweis des konkreten Gleichheitsproblems dar.

Das in dieser Arbeit gewählte Abstraktionsprinzip läßt sich folgendermaßen motivieren: Es wird zunächst davon ausgegangen, daß die beiden Terme s und t eines zu lösenden Gleichheitsproblems gemeinsame Unterterme haben. Die Idee ist nun, bei der Angleichung beider Terme das Interesse auf das Umfeld der gemeinsamen Unterterme zu fokussieren. Alle anderen Unterterme, die nicht gemeinsam sind, werden ignoriert. Die Abstraktion ist also so gewählt, daß sie alle nicht gemeinsamen Unterterme ausblendet. Das Ausblenden der nicht gemeinsamen Unterterme wird technisch dadurch realisiert, daß sie durch ein neues konstantes Funktionssymbol @ ersetzt werden. Hierzu ein Beispiel: Gegeben sei eine Spezifikation von Listen über natürlichen Zahlen mit dem Funktionssymbol cons, mit der eine natürliche Zahl an eine Liste angehängt wird, und dem Funktionssymbol app, für die Konkatenation zweier Listen. Sei folgendes Gleichheitsproblem über Listen von natürlichen Zahlen gegeben:

$$\forall \mathsf{n} : \mathsf{Nat}\ \forall \mathsf{l} : \mathsf{List}\ \exists \mathsf{l}_1, \mathsf{l}_2 : \mathsf{List}\ \mathsf{cons}(\mathsf{n}, \mathsf{l}) \equiv \mathsf{app}(\mathsf{l}_1, \mathsf{cons}(\mathsf{n}, \mathsf{l}_2))$$

Der einzige gemeinsame Unterterm der beiden Terme ist die Variable n. Die Idee ist nun, alle anderen Teil-Terme auszublenden, damit der Fokus auf die gemeinsame Variable n gesetzt wird. Dafür werden alle Unterterme, in denen die gemeinsame Variable nicht vorkommt, durch das neue konstante Funktionssymbol @ ersetzt. Dies liefert die folgende abstrakte Gleichung:

$$\forall \mathsf{n} : \mathsf{Nat}\ \mathsf{cons}(\mathsf{n}, @) \equiv \mathsf{app}(@, \mathsf{cons}(\mathsf{n}, @))$$

Die Sorten werden an dieser Stelle aus Gründen der Einfachheit vernachlässigt. Bei der späteren formalen Definition der Abstraktion werden diese dann berücksichtigt.

Jetzt wird ein Kriterium benötigt, nach denen die Gemeinsamkeiten und die Differenzen der beiden abstrakten Terme charakterisiert werden können. Dafür werden die einzelnen Vorkommen der Variablen n betrachtet. In dem rechten Term tritt die Variable n an der Termposition $[2, 1]$ auf, was soviel bedeutet, wie: “Das Vorkommen von n taucht in dem 2-ten Argument von app und in dem ersten Argument von cons auf.” Fügt man die Funktionssymbole zu der Termposition hinzu, erhält man folgende Liste: $[(\mathsf{app}, 2), (\mathsf{cons}, 1)]$. Diese Art von Listen werden im folgenden *Funktionssymbollisten* genannt. Die Funktionssymbolliste für das Vorkommen von n im linken Term ist: $[(\mathsf{cons}, 1)]$. Werden diese beiden Funktionssymbollisten miteinander verglichen, stellt man fest, daß sie die Unterliste $[(\mathsf{cons}, 1)]$ gemeinsam haben. Die Differenz der beiden Listen ist das Paar $(\mathsf{app}, 2)$ in der Funktionssymbolliste $[(\mathsf{app}, 2), (\mathsf{cons}, 1)]$. Damit kann folgendes Beweisziel formuliert werden:

> “Das Paar $(\mathsf{app}, 2)$ muß aus der Funktionssymbolliste $[(\mathsf{app}, 2), (\mathsf{cons}, 1)]$ eliminiert werden. Gleichzeitig sollen aber die Gemeinsamkeiten, nämlich die Unterliste $[(\mathsf{cons}, 1)]$, erhalten bleiben.”

In der Datenbasis muß nun nach Gleichungen gesucht werden, mit deren Hilfe das formulierte Ziel erreicht werden kann. Hierfür werden die Gleichungen aus der Datenbasis nach demselben Verfahren wie das Theorem abstrahiert. So kann zum Beispiel das Axiom

$$\forall \mathsf{l} : \mathsf{List}\ \mathsf{app}(\mathsf{nil}, \mathsf{l}) \equiv \mathsf{l}$$

bezüglich der gemeinsamen Variable l abstrahiert werden. Man erhält somit die entsprechende abstrakte Gleichung

$$\forall \mathsf{l} : \mathsf{List}\ \mathsf{app}(@, \mathsf{l}) \equiv \mathsf{l}$$

Betrachtet man die Funktionssymbollisten $[(\mathsf{app}, 2)]$ und $[]$ zu den Vorkommen der gemeinsamen Variablen l in den beiden abstrakten Termen, läßt sich folgendes feststellen: Wird die abstrakte Gleichung von links nach rechts angewendet, eliminiert sie das Paar $(\mathsf{app}, 2)$ aus der Funktionssymbolliste. Sie erfüllt also das gesteckte Beweisziel und wird deshalb ausgewählt.

Diese Abstraktionsweise von Termen und Gleichungen wird *Abstraktionsebene der abstrakten Terme* genannt. Darüberhinaus gibt es noch eine weitere Abstraktionsebene. Diese ist eine Abstraktion von abstrakten Termen, also hierarchisch über der Abstraktionsebene der abstrakten Terme angesiedelt. Diese Abstraktion blendet zusätzlich die ausgewählten gemeinsamen Unterterme der abstrakten Terme aus und betrachtet nur noch die Funktionssymbollisten, und wird daher sinngemäß *Strukturterm* genannt. Beispiel: Gegeben sei der abstrakte Term $\mathsf{app}(@, \mathsf{cons}(\mathsf{n}, @))$. Der *Strukturterm* zu diesem abstrakten Term besteht aus einer einzigen Funktionssymbolliste $[(\mathsf{app}, 1), (\mathsf{cons}, 1)]$, nämlich die zu dem Vorkommen des Unterterms n. Da mit Hilfe eines solchen *Strukturterms* die Struktur eines Termes bezüglich ausgewählter Unterterme repräsentiert wird, wird diese Abstraktion als *Abstraktionsebene der Strukturterme* bezeichnet. Die Verwendung dieser zweiten Abstraktion wird an folgendem Beispiel motiviert. Gegeben sei die oben erwähnte Spezifikation von Listen über natürlichen Zahlen sowie eine partielle Funktion max, die die größte natürliche Zahl aus einer Liste berechnet, und eine Funktion $\mathsf{reverse}$, die die Reihenfolge der Elemente einer Liste umkehrt. Unter anderem sei das folgende Axiom gegeben:

$$\forall \mathsf{n} : \mathsf{Nat}\ \mathsf{l} : \mathsf{List}\ \mathsf{max}(\mathsf{reverse}(\mathsf{cons}(\mathsf{n}, \mathsf{l}))) \equiv \mathsf{max}(\mathsf{cons}(\mathsf{n}, \mathsf{l}))$$

Das zu beweisende Theorem sei:

$$\forall \mathsf{l}_1, \mathsf{l}_2 : \mathsf{List}\ \mathsf{app}(\mathsf{l}_1, \mathsf{l}_2) \neq \mathsf{nil} \Rightarrow \mathsf{max}(\mathsf{reverse}(\mathsf{app}(\mathsf{l}_1, \mathsf{l}_2))) \equiv \mathsf{max}(\mathsf{app}(\mathsf{l}_1, \mathsf{l}_2))$$

Die Intuition sagt einem, daß, könnte das Axiom auf das Theorem angewendet werden, das Gleichheitsproblem gelöst wäre. Diese Intuition gilt es zu modellieren. Werden die *Strukturterme* des Axioms bezüglich des gemeinsamen Unterterms $\mathsf{cons}(\mathsf{n}, \mathsf{l})$ betrachtet, ist zu sehen, daß dadurch eine Strukturmodifikation modelliert wird. Diese ist

$$[(\mathsf{max}, 1), (\mathsf{reverse}, 1)] \equiv [(\mathsf{max}, 1)]$$

Analog kann dem Theorem bezüglich des gemeinsamen Unterterms $\mathsf{app}(\mathsf{l}_1, \mathsf{l}_2)$ folgendes Strukturproblem zugeordnet werden:

$$[(\mathsf{max}, 1), (\mathsf{reverse}, 1)] \equiv [(\mathsf{max}, 1)]$$

Auf der Ebene der Strukturterme ist jetzt ersichtlich, daß durch die Anwendung der Strukturgleichung auf das Strukturproblem selbiges gelöst werden könnte.

Dies ist eine Modellierung der Intuition, die man als Mensch auf dem konkreten Gleichheitsproblem hatte, und stellt einen Plan für das zu lösende Gleichheitsproblem dar. Die Ausführung dieses Plans, nämlich die Anwendung von links nach rechts der Gleichung $\mathsf{max}(\mathsf{reverse}(\mathsf{cons}(\mathsf{n},\mathsf{l}))) \equiv \mathsf{max}(\mathsf{cons}(\mathsf{n},\mathsf{l}))$ auf den linken Term des Gleichheitsproblems, liefert das Unterproblem

$$\forall \mathsf{l}_1, \mathsf{l}_2 : \mathsf{List}\ \mathsf{app}(\mathsf{u},\mathsf{v}) \neq \mathsf{nil} \Rightarrow \exists \mathsf{n} : \mathsf{Nat}\ \exists \mathsf{l} : \mathsf{List}\ \mathsf{app}(\mathsf{u},\mathsf{v}) \equiv \mathsf{cons}(\mathsf{n},\mathsf{l}).$$

Dieses Unterproblem kann leicht gelöst werden und somit kann der Plan erfolgreich ausgeführt werden.

Die in dieser Arbeit vorgestellten Abstraktionen von Gleichungen erstrecken sich sowohl auf die Axiome als auch auf das zu beweisende Gleichheitsproblem. Dabei ist noch anzumerken, daß die Bedingungen einer bedingten Gleichung bei der Abstraktion wegfallen. Dies hat den Hintergrund, daß am menschlichen Problemlösungsverhalten folgendes zu beobachten ist: Um die Gleichheit zweier Terme zu beweisen, orientiert sich ein Mensch zunächst einmal an den zur Verfügung stehenden Gleichungen, und ignoriert mehr oder weniger die Bedingungen, unter denen diese Gleichungen anwendbar sind. Erst wenn sich herausstellt, daß ein Beweis des Gleichheitsproblems mit Hilfe der Gleichungen möglich ist, versucht er die Bedingungen der verwendeten Gleichungen zu überprüfen. Daß Mathematiker oft Bedingungen weglassen, um komplexe Zusammenhänge einfacher und klarer darstellen zu können, zeigt sich auch in folgendem Auszug aus den ersten Seiten von [Mat86]:

> »When we say, that R has characteristic p, or write $\mathsf{char}\ \mathsf{R} = \mathsf{p}$, we always mean that $\mathsf{p} > 0$ is a prime number.
> In definitions and theorems about rings, it may sometimes happen, that the condition $\mathsf{A} \neq 0$ is omitted even when it is actually necessary.«

Aus den vorangegangenen Überlegungen bezüglich der Verwendung von Abstraktionen ergibt sich folgendes generelles Vorgehen: Der Ausgangspunkt ist eine Menge von Axiomen zu einer bestimmten Signatur. Zu beweisen ist eine Gleichung $\mathsf{s} \equiv \mathsf{r}$ zu dieser Signatur. Entsprechend der Terminologie von Giunchiglia und Walsh [GW92a] sind dies die "*Grund*"-Repräsentationen. Ausgehend davon werden sowohl die Axiome als auch die beiden Terme des Theorems zu sogenannten *abstrakten Gleichungen* bzw. *abstrakten Termen* abstrahiert (vgl. Abbildung 1.1). Die Abstraktion von s sei s', die von r sei r'. Auf dieser ersten Abstraktionsebene, in Anlehnung an [GW92a] auch "*abstrakte*" Repräsentation genannt, wird ein Beweis für das abstrakte Gleichheitsproblem $\mathsf{s}' \equiv \mathsf{r}'$ gesucht. Dabei werden die abstrakten Terme s' und r' mit Hilfe von abstrakten Gleichungen in einen abstrakten Term t' transformiert. In Abbildung 1.1 ist dies exemplarisch für den abstrakten

Term s' dargestellt. Die Idee ist nun, daß dieser Beweis auf der "*abstrakten*" Repräsentation einen Plan für den Beweis auf der "*Grund*"-Repräsentation darstellt; d. h. der Beweis des abstrakten Gleichheitsproblems $\mathsf{s}' \equiv \mathsf{r}'$ wird zu einem Beweis des konkreten Gleichheitsproblems $\mathsf{s} \equiv \mathsf{r}$ verfeinert (vgl. hierzu auch [GW92a]).

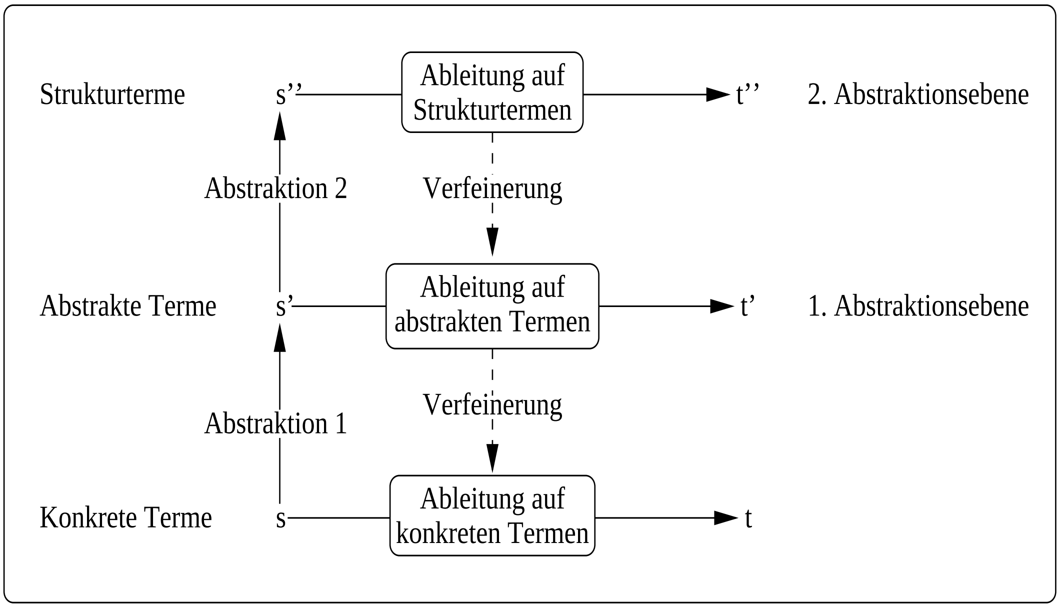


Abbildung 1.1: Generelles Vorgehen

Gelingt der Beweis des abstrakten Gleichheitsproblems nicht auf dieser ersten Abstraktionsebene, kann auf eine höhere Abstraktionsebene gewechselt werden. Diese zweite Abstraktionsebene umfaßt die sogenannten *Strukturterme* und *Strukturgleichungen*. Dabei werden die einzelnen abstrakten Axiome, wie oben erläutert, zu *Strukturgleichungen* abstrahiert und die abstrakten Terme s' und r' des abstrakten Gleichheitsproblems zu *Strukturtermen* s'' und r''. Somit gibt es eine zweite "*abstrakte*" Repräsentation, die hierarchisch über der ersten angesiedelt ist. Auf dieser zweiten Abstraktionsebene wird dann mit Hilfe der *Strukturgleichungen* ein Beweis für die Gleichheit der *Strukturterme* des Gleichheitsproblems gesucht. Die Annahme ist wiederrum, daß dieser Beweis seinerseits einen Plan für den Beweis auf der ersten Abstraktionsebene darstellt (vgl. Abbildung 1.1), welcher durch Verfeinerung des Beweises auf den Strukturtermen erhalten wird. Gelingt diese Verfeinerung des Beweises, hat man einen Beweis auf der Ebene der abstrakten Terme, welcher dann seinerseits, wie oben bereits erläutert, zu einem Beweis auf den konkreten Termen verfeinert werden kann. Gelingt die Verfeinerung eines Beweises der zweiten Abstraktionsebene zu einem Beweis der ersten

Abstraktionsebene nicht, muß versucht werden, diesen Beweis direkt zu einem Beweis auf der "Grund"-Ebene zu verfeinern.

1.1 Überblick über die Arbeit

In Kapitel 2 werden die oben motivierten Abstraktionen und die verwendeten Kalküle formal definiert. Desweiteren werden theoretische Eigenschaften der Abstraktionen untersucht, um eine Einordnung in die von Giunchiglia und Walsh [GW92a] definierten Klassen von Abstraktionen machen zu können. In Kapitel 3 werden dann die Heuristiken auf den beiden Abstraktionsebenen angegeben. Dies sind zum einen die Heuristiken auf der Ebene der abstrakten Terme, die nach dem *Differenz Reduktion* Paradigma vorgehen. Zum anderen sind dies die Heuristiken auf der Ebene der Strukturterme, der zweiten Abstraktionsebene. Kapitel 4 beschäftigt sich mit den Verfeinerungen der Beweise einer höheren Abstraktionsebene in eine tiefere und mit den dabei auftretenden Schwierigkeiten. In Kapitel 5 wird die Terminierung des aus der Kombination aller Heuristiken resultierenden Gesamtverfahrens diskutiert. Die praktische Tauglichkeit der entwickelten Heuristiken wird dann anhand einiger kommentierter Beispiele in Kapitel 6 demonstriert. Im abschließenden Kapitel 7 wird das Ergebnis dieser Arbeit nochmals diskutiert.

Kapitel 2

Abstraktionen

Dieses Kapitel beschäftigt sich mit der formalen Definition der in der Einleitung motivierten Abstraktionsebenen. Dies umfaßt die Definition der Abstraktionen von Termen und Gleichungen, sowie den Kalkül, mit dem auf den Abstraktionen Ableitungen gemacht werden können. Weiterhin werden einige theoretische Eigenschaften der Abstraktionen untersucht.

2.1 Abstraktion: "Abstrakte Terme"

Das Ausblenden bestimmter "nicht interessanter" Unterterme eines gegebenen Gleichheitsproblems kann dazu verwendet werden, eine Beweisskizze für das gegebene Gleichheitsproblem zu finden. Das Ausblenden von Untertermen wird durch die Erweiterung der Signatur um ein neues nullstelliges Funktionssymbol erreicht, welches für die auszublendenden Unterterme eingesetzt wird. Zunächst wird beschrieben, wie dies auf Termen und atomaren Formeln formal gefaßt werden kann. Dann wird noch die Abstraktionsfunktion definiert, die einen Term oder eine abstrakte Formel bezüglich einer vorgegebenen Menge von Untertermen abstrahiert.

Die Abstraktionsebene bettet sich vollständig in die Prädikatenlogik erster Stufe ein. Auf deren formale Definition wird an dieser Stelle verzichtet und auf die entsprechende Fachliteratur verwiesen (siehe [Fit90, EFT86, Men87, Ric78]). Im folgendem werden, zu gegebener Signatur Σ und Variablenmenge $\mathcal{V}$, die Terme mit $\mathsf{T}(\Sigma, \mathcal{V})$ bezeichnet, die atomaren Formeln (Literale) mit $\mathsf{At}(\Sigma, \mathcal{V})$ und die Gesamtheit aller Formeln mit $\mathsf{WFF}(\Sigma, \mathcal{V})$.

2.1.1 Abstrakte Terme

In diesem Abschnitt wird die Menge der abstrakten Terme bezüglich einer Signatur $\Sigma = (\mathsf{F}, \mathsf{P})$ und einer Variablenmenge $\mathcal{V}$ formal beschreiben.

Definition 2.1 (Abstrakte Terme) *Gegeben sei eine Signatur* $\Sigma = (\mathsf{F}, \mathsf{P})$ *und eine Menge von Variablen* $\mathcal{V}$*. Dann ist die Menge* $\mathsf{AT}(\Sigma, \mathcal{V})$ *der* **abstrakten Terme** *bezüglich* Σ *und* $\mathcal{V}$ *definiert als die Menge der Terme zur Signatur* $\Sigma' = (\mathsf{F} \cup \{@\}, \mathsf{P})$ *und der Variablenmenge* $\mathcal{V}$*. Hierbei ist @ ein neues nullstelliges Funktionssymbol; d. h.*

$$\mathsf{AT}(\Sigma, \mathcal{V}) := \mathsf{T}(\Sigma', \mathcal{V})$$

Die Signatur Σ' aus der obigen Definition wird auch als **abstrakte Signatur** der Signatur Σ bezeichnet.

Beispiele:

Sei eine Signatur $\Sigma = (\{\mathsf{0}, \mathsf{s}, \mathsf{p}, \mathsf{plus}\}, \{\equiv\})$ und eine Variablenmenge $\mathcal{V}$ gegeben. Seien $\mathsf{x}, \mathsf{y} \in \mathcal{V}$. Dann sind die folgenden Ausdrücke **abstrakte Terme** zur Signatur Σ und der Variablenmenge $\mathcal{V}$:

$$\mathsf{0}, @, \mathsf{s}(\mathsf{0}), \mathsf{s}(@), \mathsf{plus}(\mathsf{0}, @), \mathsf{s}(\mathsf{x}), \mathsf{plus}(@, \mathsf{y}), \ldots$$

2.1.1.1 Abstrakte atomare Formeln

Die Menge der abstrakten atomaren Formeln zu einer Signatur Σ kann in analoger Art und Weise wie die abstrakten Terme definiert werden.

Definition 2.2 (Abstrakte atomare Formeln) *Gegeben sei eine Signatur* $\Sigma = (\mathsf{F}, \mathsf{P})$ *und eine Menge von Variablen* $\mathcal{V}$*. Dann ist die Menge* $\mathsf{AAt}(\Sigma, \mathcal{V})$ *der* **abstrakten atomaren Formeln** *bezüglich* Σ *und* $\mathcal{V}$ *definiert als die Menge der atomaren Formeln zur Signatur* $\Sigma' = (\mathsf{F} \cup \{@\}, \mathsf{P})$ *und der Variablenmenge* $\mathcal{V}$*. Das Symbol @ ist das gleiche wie in der Definition der abstrakten Terme. Formal:*

$$\mathsf{AAt}(\Sigma, \mathcal{V}) := \mathsf{At}(\Sigma', \mathcal{V})$$

Diese Arbeit beschäftigt sich ausschließlich mit der Steuerung von Gleichheitsbeweisen. Deshalb wird die Menge der nicht atomaren abstrakten Formeln hier nicht formal definiert.

Beispiele:

Sei eine Signatur $\Sigma = (\{\mathsf{0}, \mathsf{s}, \mathsf{p}, \mathsf{plus}\}, \{\equiv\})$ und eine Variablenmenge $\mathcal{V}$ gegeben. Seien $\mathsf{x}, \mathsf{y} \in \mathcal{V}$. Dann sind die folgenden Ausdrücke **abstrakte atomare Formeln** zur Signatur Σ und der Variablenmenge $\mathcal{V}$:

$$\mathsf{s}(\mathsf{x}) + @ \equiv \mathsf{s}(\mathsf{x}), \mathsf{plus}(@, \mathsf{y}) \equiv \mathsf{y}, \mathsf{plus}(\mathsf{0}, @) \equiv @, \ldots$$

2.1.1.2 Anmerkungen

Durch diese Definition der abstrakten Terme und abstrakten atomaren Formeln als Terme und atomare Formeln einer erweiterten Signatur, sind die Begriffe "Substitution" und "Unifikation" in natürlicher Art und Weise auf die sogenannte Ebene der abstrakten Terme übertragbar. Es gibt allerdings noch ein Problem, wenn Signaturen mit Sorten verwendet werden. Das Problem ist das neu eingeführte Funktionssymbol @ : Welche Sorte soll es erhalten? Die einzige Lösung besteht darin, für jede Sorte S ein neues Symbol $@_{\mathsf{S}}$ einzuführen.

2.1.1.3 Gemeinsame Unterterme

In diesem Abschnitt soll kurz der Begriff eines **gemeinsamen Untertermes** einer Formel eingeführt werden. Aufbauend auf der Anschauung, was ein **gemeinsamer Unterterm** zweier Terme ist, läßt sich dieser Begriff zunächst auf Gleichungen ausdehnen. Problematischer wird die Anschauung, wenn von **gemeinsamen Untertermen** einer Formel die Rede ist. Dort ist die Vorstellung in etwa die folgende:

> Ein Term t ist ein gemeinsamer Unterterm einer Formel Φ , wenn
>
> - es in Φ eine Gleichung gibt, von der t ein *gemeinsamer Unterterm* ist und
> - es keine Gleichung in Φ gibt, von der t zwar Unterterm, aber nicht *gemeinsamer Unterterm* ist.

Zum Beispiel: Der Term x ist ein gemeinsamer Unterterm von

$$\mathsf{plus}(\mathsf{x}, 0) \equiv \mathsf{x} \wedge \mathsf{plus}(\mathsf{s}(\mathsf{x}), \mathsf{y}) \equiv \mathsf{s}(\mathsf{plus}(\mathsf{x}, \mathsf{y})).$$

Wohingegen der Term x kein gemeinsamer Unterterm der folgenden Formel ist:

$$\mathsf{times}(0, \mathsf{x}) \equiv 0 \wedge \mathsf{times}(\mathsf{s}(\mathsf{y}), \mathsf{x}) \equiv \mathsf{plus}(\mathsf{times}(\mathsf{y}, \mathsf{x}), \mathsf{x})$$

Formal kann der Begriff des **gemeinsamen Untertermes** einer Formel wie folgt definiert werden:

Definition 2.3 (Gemeinsame Unterterme) *Gegeben sei eine Signatur Σ und eine Menge $\mathcal{V}$ von Variablen zur Signatur Σ. Dann läßt sich der Begriff eines* gemeinsamen Untertermes einer Formel Φ *folgendermaßen induktiv über den Aufbau der Formel definieren:*

- *Falls Φ eine Gleichung $\mathsf{s} \equiv \mathsf{t}$ ist, dann ist ein beliebiger Term u ein* gemeinsamer Unterterm *von $\mathsf{s} \equiv \mathsf{t}$, genau dann wenn entweder u sowohl in s, als in t vorkommt, oder in keinem von beiden.*

- *Falls ϕ Literal, aber keine Gleichung ist, dann ist jeder Term ein Unterterm von ϕ.*

- *Falls Φ der Form $\mathsf{Op}\ \Phi'$ ist, $\mathsf{Op} \in \{\neg, \forall\mathsf{x}, \exists\mathsf{x}\}$, dann ist ein Term* u *ein* gemeinsamer Unterterm *von Φ, genau dann wenn* u gemeinsamer Unterterm *von Φ' .*

- *Falls Φ der Form $\Phi_1\mathsf{Op}\ \Phi_2$, $\mathsf{Op} \in \{\wedge, \vee\}$, dann ist ein Term* u *ein* gemeinamer Unterterm *von Φ , genau dann wenn* u *sowohl* gemeinsamer Unterterm *von Φ_1 , als auch von Φ_2 ist.*

Bemerkung: Diese Definition von gemeinsamen Untertermen einer Formel kategorisiert auch solche Terme als gemeinsame Unterterme einer Formel, wenn diese nicht in der Formel vorkommen. Im weiteren Verlauf werden aber nur solche gemeinsame Unterterme einer Formel betrachtet, die auch in der Formel vorkommen.

2.1.1.4 Abstraktionsverfahren

Im vorherigen Kapitel wurde beschrieben, daß ein Term immer bezüglich einer Menge von Untertermen abstrahiert werden soll. Um dies so allgemein wie möglich zu halten, werden nicht nur gemeinsame Unterterme betrachtet, sondern eine vorgegebene Menge von Vorkommen von gemeinsamen Untertermen. Dies soll ermöglichen, daß andere Teile des Beweisers dem Planverfahren bestimmte Vorkommen von Untertermen vorgeben können, bezüglich derer die Planung erfolgen soll. Die Vorkommen eines Untertermes werden dabei durch deren Termpositionen denotiert. Die Menge aller Termpositionen zu einem Term s sei durch $\Pi(\mathsf{s})$ gegeben.

Die Abstraktionsfunktion erhält somit einen Term s sowie eine Menge von Termpositionen $\{\mathsf{p}_1, \ldots, \mathsf{p}_\mathsf{n}\} \subseteq \Pi(\mathsf{s})$ und berechnet den entsprechenden abstrakten Term. Die Idee des Verfahrens ist wie folgt: Die Unterterme von s an Termpositionen, die nicht durch eine der Termpositionen p_i denotiert sind, werden durch die neue Konstante @ ersetzt.

Definition 2.4 (Abstraktionsfunktion auf Termen) *Sei eine Signatur $\Sigma = (\mathsf{F}, \mathsf{P})$ und eine Menge $\mathcal{V}$ von Variablen gegeben und sei $\Sigma' = (\mathsf{F} \cup \{@\}, \mathsf{P})$, wobei @ nicht aus F. Weiterhin seien $\mathsf{s} \in \mathsf{T}(\Sigma, \mathcal{V})$ und $\{\mathsf{p}_1, \ldots, \mathsf{p}_\mathsf{n}\} \subseteq \Pi(\mathsf{s})$. Dann ist der abstrakte Term $\alpha(\mathsf{s}, \{\mathsf{p}_1, \ldots, \mathsf{p}_\mathsf{n}\})$ zu dem Term s der folgende abstrakte Term:*

- *Für alle $\mathsf{p}_\mathsf{i}, 1 \leq \mathsf{i} \leq \mathsf{n}$ gilt: p_i ist eine Termposition von $\alpha(\mathsf{s}, \{\mathsf{p}_1, \ldots, \mathsf{p}_\mathsf{n}\})$ und es gilt: $\alpha(\mathsf{s}, \{\mathsf{p}_1, \ldots, \mathsf{p}_\mathsf{n}\})_{|\mathsf{p}_\mathsf{i}} = \mathsf{s}_{|\mathsf{p}_\mathsf{i}}$.*

- *Für alle anderen Termpositionen* p *von* $\alpha(\mathsf{s}, \{\mathsf{p}_1, \ldots, \mathsf{p}_n\})$ *gilt:*

$$\alpha(\mathsf{s}, \{\mathsf{p}_1, \ldots, \mathsf{p}_n\})_{|\mathsf{p}} = @.$$

Diese Abstraktionsfunktion wird nun auf Gleichungen erweitert. Dafür erhält sie als Argumente eine Gleichung $\mathsf{s} \equiv \mathsf{t}$ und zwei Mengen von Termpositionen Π_s und Π_t. Dabei muß Π_s eine Menge von Termpositionen von s sein und Π_t eine Menge von Termpositionen von t. Die Abstraktionsfunktion berechnet dann jene abstrakte Gleichung, die aus den Abstraktionen der beiden Terme s und t bezüglich der Mengen Π_s und Π_t besteht.

Definition 2.5 (Abstraktionsfunktion auf Gleichungen) *Sei eine Signatur* $\Sigma = (\mathsf{F}, \mathsf{P} \cup \{\equiv\})$ *und eine Menge* $\mathcal{V}$ *von Variablen gegeben und sei* $\Sigma' = (\mathsf{F} \cup \{@\}, \{\equiv\})$, *wobei* @ *nicht aus* F. *Weiterhin seien* $\mathsf{s}, \mathsf{t} \in \mathsf{T}(\Sigma, \mathcal{V})$ *und* Π_s *bzw.* Π_t *Mengen von Termpositionen von* s *bzw.* t. *Dann wird die Abstraktionsfunktion* α *wie folgt für negierte und unnegierte Gleichungen definiert:*

- $\alpha(\mathsf{s} \equiv \mathsf{t}, (\Pi_\mathsf{s}, \Pi_\mathsf{t})) := \alpha(\mathsf{s}, \Pi_\mathsf{s}) \equiv \alpha(\mathsf{t}, \Pi_\mathsf{t})$
- $\alpha(\neg \mathsf{s} \equiv \mathsf{t}, (\Pi_\mathsf{s}, \Pi_\mathsf{t})) := \neg\alpha(\mathsf{s}, \Pi_\mathsf{s}) \equiv \alpha(\mathsf{t}, \Pi_\mathsf{t})$

Beispiele:

Gegeben sei die Signatur $\Sigma = (\{\mathsf{s}, 0, +\}, \{\equiv\})$ und eine Variablemenge $\mathcal{V}$ mit $\mathsf{x} \in \mathcal{V}$. Die folgenden Ausdrücke sind Beispiele für die Abstraktionsfunktion α :

- $\alpha(\mathsf{s}(\mathsf{x}) + 0, \{[1]\}) = (\mathsf{s}(\mathsf{x}) + @)$
- $\alpha(\mathsf{s}(\mathsf{x}) + 0 \equiv \mathsf{s}(\mathsf{x}), (\{[1]\}, \{[]\})) = \mathsf{s}(\mathsf{x}) + @ \equiv \mathsf{s}(\mathsf{x})$

Die Abstraktionsfunktion α wurde nur auf Termen und Gleichungen definiert. Es bleibt also zu klären, was mit den anderen Formeln aus $\mathsf{WFF}(\Sigma, \mathcal{V})$ passiert.

Aus einer nicht atomaren Formel Φ werden die Gleichungen herausgefiltert, ohne irgendwelche Bedingungen zu berücksichtigen. Dies hat folgenden Grund: Beobachtet man einen Menschen beim Suchen eines Beweises für ein Gleichheitsproblem, orientiert er sich häufig zunächst nur an den Gleichungen und ignoriert gewissermaßen deren Bedingungen. Der andere Grund ist technischer Natur und wird an dem folgendem Beispiel erläutert. Sei eine bedingte Gleichung

$$\mathsf{P}(\mathsf{t}_1, \mathsf{t}_2, \mathsf{t}_3) \Rightarrow \mathsf{f}(\mathsf{t}_1, \mathsf{g}(\mathsf{t}_2, \mathsf{t}_3)) \equiv \mathsf{g}(\mathsf{t}_3, \mathsf{f}(\mathsf{t}_1, \mathsf{t}_2))$$

gegeben. Die ausgewählten Unterterme der Gleichung seien $\{\mathsf{t}_1, \mathsf{t}_2\}$. Durch die Abstraktion wird das Vorkommen des nicht ausgewählten Untertermes t_3 in der

Gleichung durch die neue Konstante @ substituiert. Was passiert mit dem Vorkommen von t_3 in der Bedingung

$$\mathsf{P}(\mathsf{t}_1, \mathsf{t}_2, \mathsf{t}_3)?$$

Das Vorkommen von t_3 nicht durch das Symbol @ zu substituieren widerspräche der zu Grunde liegenden Intuition des Verfahrens. Das Vorkommen zu substituieren würde zu Problemen beim Beweis der Bedingung führen. Wie zuverlässig wäre die Aussage, daß die abstrakte Bedingung nachgeiwesen worden wäre? Abgesehen davon müßten in dieser Arbeit auch ein Kalkül und Heuristiken erarbeitet werden, die die Behandlung allgemeiner abstrakter Formeln beinhalten. Dies ist aber nicht Gegenstand der Arbeit, da es um die Erprobung dieser Art der Planung an einem Spezialfall, den Gleichheitsproblemen, geht. Deshalb werden die Bedingungen von Gleichungen einfach ignoriert. Damit läßt sich die Abstraktionsfunktion wie folgt auf bedingte Gleichungen erweitern.

Definition 2.6 (Erweiterung der Abstraktionsfunktion) *Seien eine Signatur Σ, eine Menge von Variablen $\mathcal{V}$ zur Signatur Σ, zwei Terme $\mathsf{s}, \mathsf{t} \in \mathsf{T}(\Sigma, \mathcal{V})$ und eine Menge von Literalen $\phi_1, \ldots, \phi_\mathsf{n} \in \mathsf{At}(\Sigma, \mathcal{V})$ gegeben. Weiterhin sei Π_s eine Menge von Termpositionen von s und Π_t eine Menge von Termpositionen von t. Dann definiert sich die Erweiterung der Abstraktionsfunktion α auf bedingten Gleichungen oder Ungleichungen wie folgt:*

- $\alpha(\phi_1 \wedge \ldots \phi_\mathsf{n} \Rightarrow \mathsf{s} \equiv \mathsf{t}, (\Pi_\mathsf{s}, \Pi_\mathsf{t})) := \alpha(\mathsf{s} \equiv \mathsf{t}, (\Pi_\mathsf{s}, \Pi_\mathsf{t}))$
- $\alpha(\phi_1 \wedge \ldots \phi_\mathsf{n} \Rightarrow \neg\mathsf{s} \equiv \mathsf{t}, (\Pi_\mathsf{s}, \Pi_\mathsf{t})) := \alpha(\neg\mathsf{s} \equiv \mathsf{t}, (\Pi_\mathsf{s}, \Pi_\mathsf{t}))$

Das Paar $\mathsf{M} := (\Pi_\mathsf{s}, \Pi_\mathsf{t})$ wird in den folgenden Abschnitten der Einfachheit halber als eine Menge von Vorkommen von s und t bezeichnet.

2.1.2 Kalkül für abstrakte Terme

Das angestrebte Verfahren ist ein reines Gleichheitsverfahren auf einer Menge von abstrakten Gleichungen $\mathsf{AAt}(\Sigma, \mathcal{V})$. Angewendet werden nur unbedingte abstrakte Gleichungen. Daher wird nur eine einzige Regel benötigt, die angibt, wie eine Gleichung auf einen Term an einer bestimmten Termposition angewendet wird. Dabei werden für die Anwendung nur Matcher betrachtet.

Definition 2.7 (Anwendung einer Gleichung) *Gegeben sei eine Signatur Σ und eine Menge $\mathcal{V}$ von Variablen zur Signatur Σ. Seien $\mathsf{s}, \mathsf{u}, \mathsf{v} \in \mathsf{T}(\Sigma, \mathcal{V})$, und sei π eine Termposition von s. Falls es eine Substitution σ gibt, so daß $\mathsf{s}_{|\pi} = \sigma(\mathsf{u})$ gilt, dann ist $\mathsf{s}_{|\pi, \leftarrow \sigma(\mathsf{v})}$ der aus der Anwendung der Gleichung $\mathsf{u} \equiv \mathsf{v}$ auf s an der Termposition π resultierende Term.*

Anmerkung: In dieser Arbeit werden nur abstrakte Terme und abstrakte Gleichungen betrachtet. Dennoch ist es der Idee nach denkbar, das vorgestellte Abstraktionsverfahren allgemein auf Formeln zu erweitern. Da die abstrakten Formeln wieder Formeln der Prädikatenlogik erster Stufe sind, ist es möglich jeden Kalkül der Prädikatenlogik erster Stufe auf der “abstrakten Ebene” zur verwenden.

2.1.2.1 Ableitung

Die sukzessive Anwendung von Gleichungen auf einen Term nennt sich eine **Ableitung**. Da dieser Begriff später im Zusammenhang mit dem Plan und dessen Ausführung wichtig sein wird, wird er hier formal definiert.

Definition 2.8 (Ableitung) *Sei* Σ *eine Signatur. Dann seien eine Menge von Termen* $\mathsf{s}_\mathsf{i}, \mathsf{i} = 0 \ldots \mathsf{n}$ *aus* $\mathsf{AT}(\Sigma, \mathcal{V})$ *und eine Menge von Gleichungen* $\mathsf{u}_\mathsf{j} \equiv \mathsf{v}_\mathsf{j}, \mathsf{j} = 1 \ldots \mathsf{n}$ *aus* $\mathsf{AAt}(\Sigma, \mathcal{V})$ *gegeben. Dann ist*

$$\mathsf{s}_0 \vdash_{\mathsf{u}_1 \to \mathsf{v}_1} \mathsf{s}_1 \vdash \ldots \vdash \mathsf{s}_{\mathsf{n}-1} \vdash_{\mathsf{u}_\mathsf{n} \to \mathsf{v}_\mathsf{n}} \mathsf{s}_\mathsf{n}$$

eine Ableitung, *falls es für alle* $\mathsf{s}_\mathsf{i}, 1 \leq \mathsf{i} \leq \mathsf{n}$, *eine Termposition* $\pi_\mathsf{i} \in \Pi(\mathsf{s}_{\mathsf{i}-1})$ *und eine Substitution* σ_i *gibt, so daß gilt:*

$$\mathsf{s}_\mathsf{i} = \mathsf{s}_{\mathsf{i}-1|\pi_\mathsf{i} \leftarrow \sigma_\mathsf{i}(\mathsf{v}_\mathsf{i})}$$

Ein einzelner solcher Schritt nennt sich Ableitungsschritt.

Notation: Im folgenden wird mit $\mathsf{s} \vdash_\Phi \mathsf{s}'$ eine Ableitung abgekürzt, in der der Term s mit Hilfe von Gleichungen aus Φ zu dem Term s' abgeleitet werden kann.

2.1.3 Theoretische Eigenschaften

2.1.3.1 Einführende Bemerkungen

An dieser Stelle soll veranschaulicht werden, was während der Abstraktion und insbesondere während des Beweisens auf der abstrakten Ebene passiert.

Im folgenden wird die Menge der Terme zu einer Signatur Σ und einer Variablenmenge $\mathcal{V}$ benötigt, die alle bestimmte Termpositionen haben und an den durch die Termpositionen denotierten Untertermen den gleichen Unterterm haben. Die Menge $\mathcal{C}_\mathsf{s}(\Pi)$ dieser Terme aus $\mathsf{T}(\Sigma, \mathcal{V})$ wird bezüglich eines gegeben Referenztermes s und einer Menge von Termpositionen $\Pi \subseteq \Pi(\mathsf{s})$ definiert:

Definition 2.9 *Gegeben sei eine Signatur Σ und eine Menge von Variablen $\mathcal{V}$, und ein Term $\mathsf{s} \in \mathsf{T}(\Sigma, \mathcal{V})$. Dann wird für eine Menge Π von Termpositionen mit $\Pi \subseteq \Pi(\mathsf{s})$ die Menge $\mathcal{C}_\mathsf{s}(\Pi) \subseteq \mathsf{T}(\Sigma, \mathcal{V})$ wie folgt definiert:*

$$\mathcal{C}_\mathsf{s}(\Pi) := \{\mathsf{t} \in \mathsf{T}(\Sigma, \mathcal{V}) \mid \Pi \subseteq \Pi(\mathsf{t}) \wedge \forall \pi \in \Pi \ \mathsf{s}_\pi = \mathsf{t}_\pi\}$$

Wie schon beschrieben, werden die beiden Terme eines Gleichheitsproblems bezüglich einer Menge von Vorkommen von gemeinsamen Untertermen abstrahiert. Dies geschieht mit Hilfe der eingeführten Funktion α. Nun kann eine Relation $\equiv_\Pi$ über $\mathsf{T}(\Sigma, \mathcal{V})$ definiert werden derart, daß für alle Terme $\mathsf{s}, \mathsf{t} \in \mathcal{C}_\mathsf{r}(\Pi)$ gilt:

$$\mathsf{s} \equiv_\Pi \mathsf{t} \quad \text{gdw.} \quad \alpha(\mathsf{s}, \Pi) = \alpha(\mathsf{t}, \Pi),$$

d. h. wenn die Abstraktionen von s und t bezüglich Π gleich sind. Formal:

Definition 2.10 (Äquivalenzklasse von Termen) *Gegeben seien eine Signatur Σ, eine Menge von Variablen $\mathcal{V}$ zur Signatur Σ, und sei eine Menge Π von Termpositionen eines Termes $\mathsf{r} \in \mathsf{T}(\Sigma, \mathcal{V})$. Dann wird die Relation $\equiv_\Pi$ wie folgt definiert:*

$$\equiv_\Pi := \{(\mathsf{s}, \mathsf{t}) \in \mathcal{C}_\mathsf{r}(\Pi) \times \mathcal{C}_\mathsf{r}(\Pi) \mid \alpha(\mathsf{s}, \Pi) = \alpha(\mathsf{t}, \Pi)\}$$

Diese Relation $\equiv_\Pi$ ist die durch α und Π induzierte Äquivalenzrelation. Das diese auch eine Art Kongruenzeigenschaft besitzt, wird im folgendem Satz festgehalten:

Satz 2.1 *Gegeben seien eine Menge von Termen $\mathsf{s}_\mathsf{i}, \mathsf{t}_\mathsf{i} \in \mathsf{T}(\Sigma, \mathcal{V}), \mathsf{i} = 1 \ldots \mathsf{n}$, und eine Menge von Termpositionen $\Pi_\mathsf{i}, \mathsf{i} = 1 \ldots \mathsf{n}$, so daß für alle $1 \leq \mathsf{i} \leq \mathsf{n}$ gilt: $\mathsf{s}_\mathsf{i} \equiv_{\Pi_\mathsf{i}} \mathsf{t}_\mathsf{i}$. Weiterhin sei ein n-stelliges Funktionssymbol f gegeben und eine Menge von Termpositionen Π, die wie folgt definiert ist:*

$$\Pi := \bigcup_{\mathsf{i}=1\ldots\mathsf{n}} \{[\mathsf{i}, \pi] \mid \pi \in \Pi_\mathsf{i}\}$$

Dann gilt: $\mathsf{f}(\mathsf{s}_1, \ldots, \mathsf{s}_\mathsf{n}) \equiv_\Pi \mathsf{f}(\mathsf{t}_1, \ldots \mathsf{t}_\mathsf{n})$.

Beweis 2.1 *Per Konstruktion gelten:* $\Pi \subseteq \Pi(\mathsf{f}(\mathsf{s}_1, \ldots, \mathsf{s}_\mathsf{n}))$ *und* $\Pi \subseteq \Pi(\mathsf{f}(\mathsf{t}_1, \ldots \mathsf{t}_\mathsf{n}))$. *Da für alle* $1 \leq \mathsf{i} \leq \mathsf{n}$ $\alpha(\mathsf{s}_\mathsf{i}, \Pi_\mathsf{i}) = \alpha(\mathsf{t}_\mathsf{i}, \Pi_\mathsf{i})$ *gilt, gilt auch:* $\alpha(\mathsf{f}(\mathsf{s}_1, \ldots, \mathsf{s}_\mathsf{n}), \Pi) = \alpha(\mathsf{f}(\mathsf{t}_1, \ldots \mathsf{t}_\mathsf{n}), \Pi)$. *Daraus folgt die Behauptung.* ■

Anhand der Äquivalenzklassen läßt sich nun das Verfahren wie folgt beschreiben: Das Verfahren betrachtet die Äquivalenzklassen der beiden Terme eines Gleichheitsproblems und versucht diese durch abstrakte Gleichungen in dieselbe Äquivalenzklasse zu überführen. Die abstrakten Gleichungen stellen somit Übergänge von einer Äquivalenzklasse in eine andere dar. Gelingt dies, wird dieser Beweis auf den Äquivalenzklassen als Plan für den eigentlichen Beweis verwendet. Das Verfahren sucht also für ein gegebenes Gleichheitsproblem einen Beweis auf einer bestimmten Ebene von Äquivalenzklassen.

2.1.3.2 Theoretische Eigenschaften

In diesem Abschnitt geht es darum, ob es theoretische Verbindungen zwischen einer Formelmenge und deren Abstraktion gibt. Mit dem Ergebnis des vorherigen Abschnittes ist bekannt, daß die Abstraktionen zweier syntaktisch verschiedener Terme syntaktisch gleich sein können. Aber es kann nicht passieren, daß die Abstraktionen zweier syntaktisch gleicher Terme syntaktisch ungleich sind. Angenehm wäre es, wenn ähnliche Eigenschaften auf modelltheoretischer Seite auch gelten würden, wie zum Beispiel, daß die Abstraktion einer (un-)erfüllbaren Formelmenge auch wieder eine (un-)erfüllbare Formelmenge ist. Dies ist leider nicht der Fall, wie in Satz 2.2 belegt wird. Dies entspricht der durch Giunchiglia und Walsh festgestellten Unvermeidbarkeit widersprüchlicher abstrakter Räume [GW92a]. Dennoch gibt es eine angenehme Eigenschaft, nämlich die, daß es zu jedem konkreten beweisbaren Gleichheitsproblem auch ein beweisbares abstraktes Gleichheitsproblem gibt. Dies läßt sich daraus folgern, daß jedes Gleichheitsproblem $\mathsf{s} \equiv \mathsf{t}$ zu dem trivialen Gleichheitsproblem $@ \equiv @$ abstrahieren läßt. Insofern läßt sich also die hier gewählte Abstraktionsweise in die Klasse der TI-Abstraktionen im Sinne von Giunchiglia und Walsh (siehe [GW92a]) einordnen, da es für jedes Theorem des "Grund"-Raumes eine mögliche Abstraktionsweise gibt, die eine TI-Abstraktion im Sinne von Giunchiglia und Walsh ist. Eigentlich ist die hier gewählte Abstraktion aber gar keine Abstraktion im Sinne von Giunchiglia und Walsh. Vielmehr beschreibt diese Abstraktion eine ganze Klasse von Abstraktionen im Sinne von Giunchiglia und Walsh. Daß es für jedes Theorem des "Grund"-Raumes eine solche triviale TI-Abstraktion gibt, ist aber praktisch irrelevant, da aus dem Beweis des abstrakten Gleichheitsproblems $@ \equiv @$ keinerlei Hilfen für die Erstellung eines Beweises für das konkrete Gleichheitsproblem abgeleitet werden können. Deshalb beschäftigt sich der zweite Teil dieses Abschnittes mit der Charakterisierung einer Klasse von beweisbaren Gleichheitsproblemen, für deren Abstraktionen es auch einen Beweis gibt und aus dem sich Rückschlüße für die Erstellung eines konkreten Beweises gewinnen lassen.

Unvermeidbarkeit widersprüchlicher abstrakter Räume:

Satz 2.2 *Gegeben seien eine Signatur Σ und eine Menge $\mathcal{V}$ von Variablen zu dieser Signatur. Weiterhin seien mit $\phi_1, \ldots, \phi_\mathsf{n}$ (bedingte) Gleichungen oder Ungleichungen zu dieser Signatur und dieser Variablenmenge gegeben. Für jedes dieser ϕ_i gebe es eine Menge von Vorkommen $(\Pi^\mathsf{i}_1, \Pi^\mathsf{i}_2)$ von gemeinsamen Untertermen. Dann gilt:*

> *Die Erfüllbarkeit oder die Unerfüllbarkeit von $\alpha(\phi_1, (\Pi^1_1, \Pi^1_2)) \wedge \ldots \wedge \alpha(\phi_\mathsf{n}, (\Pi^\mathsf{n}_1, \Pi^\mathsf{n}_2))$ steht in keinem Verhältnis zur Erfüllbarkeit oder Unerfüllbarkeit von $\phi_1 \wedge \ldots \wedge \phi_\mathsf{n}$.*

Beweis 2.2

- *Sei die unerfüllbare Formel* $\forall \mathsf{x}\ \mathsf{f}(\mathsf{x}) \equiv \mathsf{x} \wedge \neg \mathsf{f}(\mathsf{a}) \equiv \mathsf{b} \wedge \mathsf{a} \equiv \mathsf{b}$. *Die Abstraktion dieser Formel bezüglich* $\{\mathsf{x}\}$ *ist die Formel* $\forall \mathsf{x}\ \mathsf{f}(\mathsf{x}) \equiv \mathsf{x} \wedge @ \equiv @ \wedge @ \equiv @$, *welche erfüllbar ist. Dahingegen ist die Abstraktion der unerfüllbaren Formel* $\forall \mathsf{x}\ \neg \mathsf{x} \equiv \mathsf{x}$ *bezüglich* $\{\mathsf{x}\}$ *die unerfüllbare Formel* $\forall \mathsf{x}\ \neg \mathsf{x} \equiv \mathsf{x}$.

- *Sei die folgende Formel* Φ *gegeben:* $\forall \mathsf{x}\ \mathsf{f}(\mathsf{x},\mathsf{a}) \equiv \mathsf{f}(\mathsf{x},\mathsf{b}) \wedge \neg \mathsf{f}(\mathsf{x},\mathsf{c}) \equiv \mathsf{f}(\mathsf{x},\mathsf{d})$. *Diese Formel ist erfüllbar. Dagegen ist die folgende Abstraktion von* Φ *unerfüllbar:* $\forall \mathsf{x}\ \mathsf{f}(\mathsf{x},@) \equiv \mathsf{f}(\mathsf{x},@) \wedge \neg \mathsf{f}(\mathsf{x},@) \equiv \mathsf{f}(\mathsf{x},@)$.

  Dahingegen ist die Abstraktion $\forall \mathsf{x}\ \mathsf{f}(\mathsf{x},@) \equiv \mathsf{f}(\mathsf{x},@)$ *der Formel* $\forall \mathsf{x}\ \mathsf{f}(\mathsf{x},\mathsf{a}) \equiv \mathsf{f}(\mathsf{x},\mathsf{a})$ *genauso erfüllbar, wie die ursprüngliche Formel.*

■

Aus diesem Satz folgt dann auch, daß die klassischen widerlegungsvollständigen Beweisverfahren auf der abstrakten Ebene nicht verwendbar sind. Der Grund dafür ist, daß die Abstraktion einer erfüllbaren Axiomenmenge Φ widersprüchlich sein kann. Falls also für die Widerlegung einer abstrakten Ungleichung $\neg \mathsf{s} \equiv \mathsf{t}$ widerlegungsvollständige Verfahren verwendet werden, finden diese vielleicht einen Widerspruch, obwohl $\Phi \models \mathsf{s} \equiv \mathsf{t}$ nicht gilt. Der gefundene Widerspruch ist dann nur aus den widersprüchlichen Abstraktionen der Axiome gefolgert worden.

Kategorisierung von beweisbaren Gleichheitsproblemen, deren Abstraktionen auch beweisbar sind

Dem aus den in Kapitel 3 vorgestellten Heuristiken resultierenden Beweisverfahren liegt das Paradigma zu Grunde, daß die für die Bildung der Abstraktion eines Gleichheitsproblems $\mathsf{s} \equiv \mathsf{t}$ betrachteten gemeinsamen Unterterme von s und t im Verlauf des Beweises erhalten bleiben sollen. In diesem Abschnitt wird versucht, eine theoretische Beschreibung der Klasse von konkreten Gleichheitsproblemen zu erhalten, die auf der konkreten Ebene und auf der abstrakten Ebene unter dem genanntem Paradigma beweisbar sind.

Der Beweis der Gleichheit zweier Terme s und t mit Hilfe von (bedingten) Gleichungen Φ kann wie folgt dargestellt werden:

$$\mathsf{s} \vdash^*_{\Phi} \mathsf{t}$$

Es wird nun zunächst definiert, wann für einen einzelnen Ableitungsschritt

$$\mathsf{s} \vdash_{\pi,\sigma(\mathsf{u})\to\sigma(\mathsf{v})} \mathsf{t}$$

ein abstrakter Ableitungsschritt auf einer Abstraktion von s bezüglich einer Menge von Vorkommen von Untertermen $\Pi \subseteq \Pi(\mathsf{s})$ und unter dem genannten Paradigma existiert. Ein solcher Ableitungsschritt wird dann als "repräsentierbar unter der Abstraktion Π" bezeichnet. Ein solcher Ableitungsschritt ist genau dann repräsentierbar, wenn alle durch Π denotierten Unterterme von s (nicht die Vorkommen) auch in t enthalten sind. Dies wird in der folgenden Definition formal beschrieben.

Definition 2.11 ($\alpha\Pi$-repräsentierbare Ableitungsschritte) *Gegeben seien vier Terme $\mathsf{s}, \mathsf{s}', \mathsf{u}, \mathsf{v}$ zu einer Signatur Σ und einer Variablenmenge $\mathcal{V}$. Sei weiterhin $\pi \in \Pi(\mathsf{s})$ und σ eine Substitution, so daß $\mathsf{s}_{|\pi} = \sigma(\mathsf{u})$ gilt, und es gelte $\mathsf{s} \vdash_{\pi,\sigma(\mathsf{u})\to\sigma(\mathsf{v})} \mathsf{s}'$. Dann wird unter dem Paradigma der Erhaltung gemeinsamer Unterterme dieser Ableitungsschritt als $\alpha\Pi$-*repräsentierbar *für eine Menge $\Pi \subseteq \Pi(\mathsf{s})$ von Vorkommen definiert, wenn:*

1. *Entweder π weder Präfix noch Suffix einer Termposition aus Π ist,*
2. *oder π Präfix einer Termposition π' aus Π und*
   (a) *entweder gibt es ein $\pi'' \in \Pi$, mit $\mathsf{s}_{|\pi''} = \mathsf{s}_{|\pi'}$ und π ist kein Präfix von π'',*
   (b) *oder es gibt ein $\pi_\mathsf{u} \in \Pi(\mathsf{u})$, so daß $\pi\pi_\mathsf{u}$ Präfix von π' ist und $\mathsf{u}_{|\pi_\mathsf{u}}$ kommt in v vor.*
3. *oder π Suffix einer Termposition π' aus Π ist und*
   (a) *entweder gibt es ein $\pi'' \in \Pi$, mit $\mathsf{s}_{|\pi''} = \mathsf{s}_{|\pi'}$ und π ist kein Präfix von π'',*
   (b) *oder es gilt $\sigma(\mathsf{u}) = \sigma(\mathsf{v})$.*

Im folgenden wird ein solcher Ableitungsschritt kurz als $\alpha\Pi$-repräsentierbar bezeichnet.

Auf Grund dieser Kategorisierung eines Ableitungsschrittes gilt dann, daß wenn ein Ableitungsschritt $\mathsf{s} \vdash_{\pi,\sigma(\mathsf{u})\to\sigma(\mathsf{v})} \mathsf{s}'$ $\alpha\Pi$-*repräsentierbar* ist, daß dann alle durch Π charakterisierten Unterterme von s auch in s' enthalten sind.

Satz 2.3 ($\alpha\Pi$-Repräsentierbare Ableitungsschritte erhalten die Unterterme) *Sei $\mathsf{s} \vdash_{\pi,\sigma(\mathsf{u})\to\sigma(\mathsf{v})} \mathsf{s}'$ $\alpha\Pi$-repräsentierbar. Dann gibt es ein $\Pi' \subseteq \Pi(\mathsf{s}')$, so daß gilt:*

$$\forall \pi \in \Pi \; \exists \pi' \in \Pi' \; \mathsf{s}_{|\pi} = \mathsf{s}_{|\pi'}$$

Beweis 2.3 *Folgt direkt aus der Definition von $\alpha\Pi$-repräsentierbaren Ableitungsschritten.* ■

Desweiteren kann gezeigt werden, daß jeder $\alpha\Pi$-repräsentierbare Ableitungsschritt $\mathsf{s} \vdash_{\pi,\sigma(\mathsf{u})\to\sigma(\mathsf{v})} \mathsf{s}'$ auf der abstrakten Ebene nachvollzogen werden kann. Dies bedeutet, daß entweder $\alpha(\mathsf{s},\Pi) = \alpha(\mathsf{s}',\Pi)$ gilt, oder es eine Abstraktion $\alpha(\mathsf{u},\Pi_\mathsf{u}) \equiv \alpha(\mathsf{v},\Pi_\mathsf{v})$ der konkreten Gleichung $\mathsf{u} \equiv \mathsf{v}$ bezüglich von Vorkommen gemeinsamer Unterterme gibt, so daß gilt:

$$\alpha(\mathsf{s}',\Pi') = \alpha(\mathsf{s},\Pi)_{|\pi,\leftarrow\sigma(\alpha(\mathsf{v},\Pi_\mathsf{v}))}$$

wobei Π' eine Menge von Termpositionen ist, deren Existenz in Satz 2.3 gezeigt wurde.

Satz 2.4 ($\alpha\Pi$-repräsentierbare Ableitungsschritte sind auf der abstrakten Ebene nachvollziehbar) *Sei ein $\alpha\Pi$-repräsentierbarer Ableitungsschritt*

$$\mathsf{s} \vdash_{\pi\sigma(\mathsf{u})\to\sigma(\mathsf{v})} \mathsf{s}'$$

gegeben. Dann gilt:

- *entweder ist $\alpha(\mathsf{s},\Pi) = \alpha(\mathsf{s}',\Pi)$*
- *oder es gibt eine Menge von Vorkommen $(\Pi_\mathsf{u},\Pi_\mathsf{v})$ von gemeinsamen Untertermen der Gleichung $\mathsf{u} \equiv \mathsf{v}$ und ein Π' wie in Satz 2.3, so daß gilt:*

$$\alpha(\mathsf{s}',\Pi') = \alpha(\mathsf{s},\Pi)_{|\pi,\leftarrow\sigma(\alpha(\mathsf{v},\Pi_\mathsf{v}))}$$

Beweis 2.4

Fälle 1., 2.(a), 3.(a) und 3.(b) aus Definition 2.11: *In diesen Fällen folgt direkt, daß gilt: $\alpha(\mathsf{s},\Pi) = \alpha(\mathsf{s}',\Pi)$.*

Fall 2.(b) aus Definition 2.11: *Hier gilt, daß π Präfix einer Termposition $\pi' \in \Pi$ ist. Weiterhin gilt, daß der konkrete Term $\mathsf{s}_{|\pi'}$ sowohl Unterterm von $\sigma(\mathsf{u})$ als auch von $\sigma(\mathsf{v})$ ist. Es gibt also eine Termpositionen π_1 und mindestens eine Termposition π_2 mit: $\sigma(\mathsf{u})_{|\pi_1} = \sigma(\mathsf{v})_{|\pi_2} = \mathsf{s}_{|\pi'}$. Per Definition gibt es auch eine Termposition π_u, die Präfix ist von π_1, und eine Termposition π_v, die Präfix ist von einem dieser π_2, für die gilt: $\mathsf{u}_{|\pi_\mathsf{u}} = \mathsf{v}_{|\pi_\mathsf{v}}$.*

Wird diese Betrachtung für alle Termpositionen π' von s gemacht, deren Präfix π ist, erhält man eine Reihe Vorkommen von gemeinsamen Untertermen $(\Pi_\mathsf{u},\Pi_\mathsf{v})$, nach denen die Gleichung $\mathsf{u} \equiv \mathsf{v}$ abstrahiert werden muß, so daß gilt:

$$\alpha(\mathsf{s}',\Pi') = \alpha(\mathsf{s},\Pi)_{|\pi,\leftarrow\sigma(\alpha(\mathsf{v},\Pi_\mathsf{v}))}$$

Die Mengen Π_u und Π_v bestehen aus den für alle betroffenen π' oben bestimmten π_u und π_v.

Damit ist gezeigt, daß es eine Abstraktion der Gleichung $u \equiv v$ gibt, mit der der konkrete Ableitungsschritt auf der abstrakten Ebene nachvollzogen werden kann. Die Menge Π' der Vorkommen in s' berechnet sich aus Π, indem die Suffixe π_1 der betroffenen π' ausgetauscht werden gegen die entsprechenden π_2.

■

Mit der Definition, dessen was ein $\alpha\Pi$-repräsentierbarer Ableitungsschritt ist, kann nun definiert werden, was eine *repräsentierbare Ableitung* $s \vdash^* s'$ ist. Dies kann allerdings nur bezüglich einer Menge Π_s von Vorkommen von Termen in s definiert werden. Deshalb wird eine *repräsentierbare Ableitung* immer nur bezüglich einer Anfangsabstraktion Π_s definiert. Eine solche Ableitung wird dann als $\alpha\Pi_s$ *-repräsentierbare Ableitung* bezeichnet.

Definition 2.12 ($\alpha\Pi$-repräsentierbare Ableitung) *Gegeben sei eine Menge von Termpositionen Π_0 zu einem Term s_0. Eine Ableitung $s_0 \vdash s_1 \vdash \ldots \vdash s_n$ ist genau dann $\alpha\Pi_0$-repräsentierbar, wenn*

- *es die leere Ableitung ist*
- *oder der Ableitungsschritt $s_0 \vdash s_1$ $\alpha\Pi_0$-repräsentierbar ist, und die Ableitung $s_1 \vdash \ldots \vdash s_n$ $\alpha\Pi_1$-repräsentierbar ist. Dabei ist Π_1 die Menge von Vorkommen in s_1, die sich aus s_0 nach dem im Beweis von Satz 2.4 skizzierten Verfahren berechnen läßt.*

Aus der Definition einer $\alpha\Pi$-repräsentierbaren Ableitung und Satz 2.4 läßt sich unmittelbar folgern, daß es für eine solche Ableitung auch auf der abstrakten Ebene eine Ableitung gibt. Dies ist die Aussage des nächsten Satzes:

Satz 2.5 *Zu jeder $\alpha\Pi$-repräsentierbaren Ableitung $s \vdash^* r$ gibt es eine Ableitung*

$$\alpha(s, \Pi) \vdash^* \alpha(r, \Pi_r)$$

auf der abstrakten Ebene, wobei Π_r sich gemäß dem im Beweis von Satz 2.4 skizzierten Verfahren sukzessive aus Π berechnen läßt.

Beweis 2.5 *Folgt direkt aus Definition 2.12 und Satz 2.4.* ■

Daraus kann nun gefolgert werden, daß wenn für ein Gleichheitsproblem $s \equiv t$ ein Beweis

$$s \vdash^*_\Phi t$$

existiert, so daß, bezüglich einer Menge (Π_s, Π_t) von Vorkommen von gemeinsamen Untertermen, die Ableitung $s \vdash^*_\Phi t$ eine $\alpha\Pi_s$-repräsentierbar ist, auch ein Beweis für das abstrakte Gleichheitsproblem $\alpha(s, \Pi_s) \equiv \alpha(t, \Pi_t)$ gefunden werden kann.

2.2 Abstraktion: "Termstrukturen"

In einer Termstrukur eines gegebenen Terms soll die Struktur dieses Termes bezüglich bestimmter vorgegebener Unterterme repräsentiert werden. Der Unterschied zu der vorherigen Abstraktion besteht in der Abstraktionsfunktion. Die neue Abstraktionsfunktion wird mit β bezeichnet. Diese erhält, wie auch die vorherige Abstraktionsfunktion α, einen Term $\mathsf{s} \in \mathsf{T}(\Sigma, \mathcal{V})$ und eine Menge $\Pi \in \Pi(\mathsf{s})$ von Vorkommen. Sei $\{\mathsf{t}_1, \ldots, \mathsf{t}_\mathsf{n}\}$ die Menge der durch die Termpositionen Π denotierten Unterterme von s, und seien $\mathsf{x}_1, \ldots, \mathsf{x}_\mathsf{n} \in \mathcal{V}$ neue Variablen. Dadurch läßt sich für jeden Term s und jede Menge von Termpositionen $\Pi \subseteq \Pi(\mathsf{s})$ eine Termsubstitution $\rho_{\mathsf{s},\Pi}$ definieren:

$$\rho_{\mathsf{s},\Pi} := \{\mathsf{t}_1 \to \mathsf{x}_1, \ldots \mathsf{t}_\mathsf{n} \to \mathsf{x}_\mathsf{n}\}$$

Der berechnete abstrakte Term $\beta(\mathsf{s}, \Pi)$ ist der Term $\alpha(\mathsf{s}, \Pi)$ auf den die Termsubstitution $\rho_{\mathsf{s},\Pi}$ angewendet wurde. Es gilt also:

$$\beta(\mathsf{s}, \Pi) := \rho_{\mathsf{s},\Pi}(\alpha(\mathsf{s}, \Pi))$$

Für die Abstraktion einer Gleichung $\mathsf{s} \equiv \mathsf{t}$ aus $\mathsf{At}(\Sigma, \mathcal{V})$, wird eine Menge $(\Pi_\mathsf{s}, \Pi_\mathsf{t})$ von Vorkommen von gemeinsamen Untertermen der beiden Terme s und t verwendet. Die Abstraktion der Gleichung $\mathsf{s} \equiv \mathsf{t}$ ist dann eine Gleichung aus $\mathsf{At}(\Sigma \cup \{@\}, \mathcal{V})$, die sich wie folgt berechnet:

$$\beta(\mathsf{s} \equiv \mathsf{t}, (\Pi_\mathsf{s}, \Pi_\mathsf{t})) := \rho_{\mathsf{s},\Pi_\mathsf{s}}(\alpha(\mathsf{s}, \Pi_\mathsf{s})) \equiv \rho_{\mathsf{t},\Pi_\mathsf{t}}(\alpha(\mathsf{t}, \Pi_\mathsf{t}))$$

Gleiches gilt für Ungleichungen $\neg \mathsf{s} \equiv \mathsf{t}$.

Analog zur Definition von α, läßt sich die neue Abstraktionsfunktion β auf bedingten Gleichungen definieren. Gegeben sei eine bedingte Gleichung $\phi_1 \wedge \ldots \wedge \phi_\mathsf{n} \Rightarrow \mathsf{s} \equiv \mathsf{t}$ und eine Menge $(\Pi_\mathsf{s}, \Pi_\mathsf{t})$ von Vorkommen von gemeinsamen Termen von s und t. Dann gilt:

$$\beta(\phi_1 \wedge \ldots \wedge \phi_\mathsf{n} \Rightarrow \mathsf{s} \equiv \mathsf{t}, (\Pi_\mathsf{s}, \Pi_\mathsf{t})) := \beta(\mathsf{s} \equiv \mathsf{t}, (\Pi_\mathsf{s}, \Pi_\mathsf{t}))$$

Gleiches gilt für bedingte Ungleichungen. Damit ließe sich die Abstraktionsfunktion β formal definieren. Es gibt allerdings noch eine kleine Änderung, die mit der folgenden Beobachtung zusammenhängt. Zu Beweisen sei das konkrete Gleichheitsproblem

$$\mathsf{f}(\mathsf{a}, \mathsf{b}) \equiv \mathsf{f}(\mathsf{b}, \mathsf{a})$$

Das nach der bisherigen Definition der Funktion β bezüglich der Vorkommen der gemeinsamen Unterterme a und b berechnete Gleichheitsproblem von Strukturtermen ist

$$\mathsf{f}(\mathsf{x},\mathsf{y}) \equiv \mathsf{f}(\mathsf{y},\mathsf{x}),$$

wobei x und y neue Variablen sind. Sei weiterhin ein Axiom $\forall \mathsf{z}\ \mathsf{f}(\mathsf{z},\mathsf{z}) \equiv \mathsf{z}$ gegeben. Die Strukturgleichung dieses Axioms bezüglich der Vorkommen von z ist wieder dieselbe Gleichung. Die Anwendung dieser Strukturgleichung von links nach rechts auf einen der beiden Terme, würde dazu führen, daß die Variablen x und y unifiziert würden, d. h. daß die beiden unterschiedlichen Unterterme zu einem einzigen zusammenfallen würden. Um diesen Effekt zu vermeiden, werden die gemeinsamen Unterterme eines zu lösenden Gleichheitsproblems nicht durch Variablen, sondern durch neue Konstanten $\mathsf{AC_i}$ ersetzt. Dadurch ist gewährleistet, daß sie durch Unifikation nicht zu einem einzigen Unterterm zusammengefaßt werden können. Daraus ergibt sich, daß die für einen Term s eines zu lösenden Gleichheitsproblems und eine Menge $\Pi_\mathsf{s} \subseteq \Pi(\mathsf{s})$ verwendete Termsubstitution die gemeinsamen Unterterme nicht durch Variablen, sondern durch neue Konstanten ersetzt. Ansonsten bleibt die Definition der Abstraktionsfunktion β unverändert.

Die Repräsentation von Strukturtermen ist somit die gleiche wie die eines abstrakten Terms aus dem vorherigen Abschnitt. Zur Repräsentation von Strukturtermen einer Signatur Σ wird diese Signatur um neue Konstanten $@, \mathsf{AC}_1, \ldots, \mathsf{AC}_\mathsf{n}$ erweitert. Ein Strukturterm eines Termes aus $\mathsf{T}(\Sigma, \mathcal{V})$ ist also ein Term aus

$$\mathsf{T}(\Sigma \cup \{@, \mathsf{AC}_1, \ldots, \mathsf{AC}_\mathsf{n}\}, \mathcal{V}).$$

Mit diesen einführenden Erläuterungen läßt sich die Funktion β formal definieren.

Definition 2.13 (Abstraktionsfunktion β) *Gegeben seien eine Signatur Σ und eine Menge $\mathcal{V}$ von Variablen. Seien $\mathsf{s}, \mathsf{t} \in \mathsf{T}(\Sigma, \mathcal{V})$, $\Pi_\mathsf{s} \in \Pi(\mathsf{s})$, $\Pi_\mathsf{t} \in \Pi(\mathsf{t})$ und ϕ eine Formel zur Signatur Σ und Variablenmenge $\mathcal{V}$. Dann definiert sich die Funktion β wie folgt:*

- *Falls $(\phi \Rightarrow)\mathsf{s} \equiv \mathsf{t}$ ein zu beweisendes (bedingtes) Gleichheitsproblem ist, dann sei eine Termsubstitution ρ gegeben, die die durch Π_s und Π_t betrachteten gemeinsamen Unterterme von s und t durch neue Konstanten $\mathsf{AC_i}$ ersetzt. Dann wird die Funktion β wie folgt definiert:*

$$\beta((\phi \Rightarrow)\mathsf{s} \equiv \mathsf{t}, (\Pi_\mathsf{s}, \Pi_\mathsf{t})) := \rho(\alpha(\mathsf{s}, \Pi_\mathsf{s})) \equiv \rho(\alpha(\mathsf{s}, \Pi_\mathsf{t}))$$

- *Falls $(\phi \Rightarrow)\mathsf{s} \equiv \mathsf{t}$ ein Axiom ist, dann sei eine Termsubstitution ρ gegeben, die die durch Π_s und Π_t betrachteten gemeinsamen Unterterme von s und t durch neue Variablen ersetzt. Dann wird die Funktion β wie folgt definiert:*

$$\beta((\phi \Rightarrow)\mathsf{s} \equiv \mathsf{t}, (\Pi_\mathsf{s}, \Pi_\mathsf{t})) := \rho(\alpha(\mathsf{s}, \Pi_\mathsf{s})) \equiv \rho(\alpha(\mathsf{s}, \Pi_\mathsf{t}))$$

Dadurch, daß Strukturterme auch abstrakte Terme sind, kann der in Abschnitt 2.1.2 definierte Kalkül auch für Struturterme verwendet werden.

2.2.1 Theoretische Eigenschaften

Wie leicht einzusehen ist, gilt das Ergebnis von Satz 2.2 auch entsprechend für die Abstraktionsfunktion β. Zu klären ist, welche Klasse von Gleichheitsbeweisen des "Grund"-Raumes auch einen Beweis im abstrakten Raum haben. Wie auch bei der Abstraktionsebene der abstrakten Terme, kann festgestellt werden, daß jedes konkrete Gleichheitsproblem zu einem trivialen abstrakten Gleichheitsproblem abstrahiert werden kann. Somit kann für jedes beweisbare konkrete Gleichheitsproblem theoretisch betrachtet auch ein Beweis für eine Abstraktion dieses Gleichheitsproblems gefunden werden. Wie auch schon bei der Diskussion der theoretischen Eigenschaften der Abstraktionsebene der abstrakten Terme festgestellt wurde, bringt der abstrakter Beweis eines *trivialen* Gleichheitsproblems keinerlei Hilfe im Hinblick auf die Erstellung eines konkreten Beweises. Deshalb wird auch hier wieder eine spezielle Klasse von Gleichheitsproblemen betrachtet. Dabei wird wieder das Paradigma zu Grunde gelegt, daß die gemeinsamen Unterterme eines zu lösenden Gleichheitsproblems erhalten bleiben müssen. Diese Einschränkung ist auch deshalb notwendig, weil sonst keinerlei Kriterium vorhanden ist, bezüglich welcher Unterterme eine Gleichung abstrahiert werden soll. Dies schränkt die Klasse der Gleichheitsprobleme, zu denen ein abstrakter Beweis existiert, auf jene ein, in denen die beiden Terme des Gleichheitsproblems mindestens einen gemeinsamen Unterterm haben. Im folgenden wird nun eine Beschreibung der gesuchten Klasse von Gleichheitsproblemen angegeben. Dafür wird zunächst angegeben, wann eine Ableitung $\mathsf{s} \vdash^* \mathsf{s}'$ des "Grund"-Raumes durch einen einzigen Ableitungsschritt im abstrakten Raum repräsentiert werden kann. Dies geschieht, wie bei der Abstraktion durch α, bezüglich einer Menge von Vorkommen $\Pi_\mathsf{s} \in \Pi(\mathsf{s})$. Eine solche Ableitung wird dann als durch einen $\beta\Pi_\mathsf{s}$ *-Ableitungschritt repräsentierbar* bezeichnet.

Definition 2.14 (**$\beta\Pi_\mathsf{s}$-Ableitungschritt repräsentierbare Ableitungen**)
Gegeben seien zwei Terme $\mathsf{s}, \mathsf{s}' \in \mathsf{T}(\Sigma, \mathcal{V})$ *und eine Menge von (bedingten) Gleichungen* Φ. *Weiterhin sei* $\mathsf{s} \vdash^*_\Phi \mathsf{s}'$ *eine Ableitung von* s *nach* s' *unter Verwendung von (bedingten) Gleichungen aus* Φ, *und* $\Pi_\mathsf{s} \subseteq \Pi(\mathsf{s})$. *Dieser Ableitungsschritt ist genau dann durch einen* $\beta\Pi_\mathsf{s}$-Ableitungschritt repräsentierbar, *wenn gilt: Es gibt eine bedingte Gleichung* $\mathsf{u} \equiv \mathsf{v}$ *und eine Termposition* $\pi \in \Pi(\mathsf{s}) \cap \Pi(\mathsf{s}')$, *die Präfix eines der* $\pi' \in \Pi_\mathsf{s}$ *ist, so daß gilt:*

$$\Phi \models \exists \mathsf{y}_1 \dots \mathsf{y}_\mathsf{n} \in \mathsf{Var}(\mathsf{u}) \left[\mathsf{s}_{|\pi \leftarrow \mathsf{u}} \equiv \mathsf{s} \wedge \exists \mathsf{x}_1 \dots \mathsf{x}_\mathsf{m} \in \mathsf{Var}(\mathsf{v}) \setminus \mathsf{Var}(\mathsf{u})\ \mathsf{s}'_{|\pi \leftarrow \mathsf{v}} \equiv \mathsf{s}'\right]$$

und entweder gibt es ein $\pi'' \in \Pi_\mathsf{s}$, $\pi'' \neq \pi'$, *so daß* π *kein Präfix von* π'' *ist, oder es gibt ein* $\pi_\mathsf{u} \in \Pi(\mathsf{u})$, *so daß* $\pi\pi_\mathsf{u}$ *Präfix von* π' *ist und* $\mathsf{u}_{|\pi_\mathsf{u}}$ *kommt in* v *vor.*

Im Klartext bedeutet diese Definition, daß eine Ableitung $\mathsf{s} \vdash^*_\Phi \mathsf{s}'$ durch einen einzigen abstrakten Ableitungsschritt repräsentiert werden kann, wenn es eine

Gleichung gibt, die unter der *Theorie* Φ auf s angewendet werden kann und dabei die durch Π_s denotierten Unterterme erhält. Das Ergebnis der abstrakten Gleichungsanwendung muß eine Abstraktion von s' sein.

Die Vorkommen in s' der durch Π_s betrachteten Vorkommen in s lassen sich direkt aus s und den Positionen der Vorkommen in u und v bestimmen.

Mit Hilfe dieser Definition kann definiert werden, wann eine Ableitung durch eine $\beta\Pi$-Ableitung repräsentiert werden kann.

Definition 2.15 (Durch $\beta\Pi_\mathsf{s}$-Ableitungen repräsentierbare Ableitungen) *Gegeben seien zwei Terme* $\mathsf{s}, \mathsf{s}' \in \mathsf{T}(\Sigma, \mathcal{V})$ *und eine Menge von (bedingten) Gleichungen* Φ. *Weiterhin sei* $\mathsf{s} \vdash_\Phi^* \mathsf{s}'$ *eine Ableitung von* s *nach* s' *unter Verwendung von Gleichungen aus* Φ, *und* $\Pi_\mathsf{s} \subseteq \Pi(\mathsf{s})$. *Dieser Ableitungsschritt ist genau dann durch eine* $\beta\Pi_\mathsf{s}$-Ableitung repräsentierbar, *wenn gilt:*

- *Entweder ist* $\mathsf{s} \vdash_\Phi^* \mathsf{s}'$ *durch einen* $\beta\Pi_\mathsf{s}$*-Ableitungsschritt repräsentierbar,*
- *oder es gibt einen Term* s'', *so daß* $\mathsf{s} \vdash_\Phi^* \mathsf{s}''$ *durch eine* $\beta\Pi_\mathsf{s}$*-Ableitung repräsentierbar ist und* $\mathsf{s}'' \vdash_\Phi^* \mathsf{s}'$ *durch eine* $\beta\Pi_{\mathsf{s}''}$*-Ableitung repräsentierbar ist.*

Mit Hilfe dieser Definitionen läßt sich nun eine Aussage darüber treffen, für welche Art von Gleichheitsproblemen mit Hilfe der durch β definierten Abstraktion und dem zu Grunde liegenden Paradigma im abstrakten Raum ein Beweis zu finden ist.

Satz 2.6 (Klasse der repräsentierbaren Beweise) *Gegeben sei eine Signatur* Σ *und eine Menge von Variablen* $\mathcal{V}$. *Weiterhin seien* $\mathsf{s}, \mathsf{t} \in \mathsf{T}(\Sigma, \mathcal{V})$ *und* Φ *eine Menge von (bedingten) Gleichungen. Falls* $\phi \models \mathsf{s} \equiv \mathsf{t}$ *gilt, gibt es einen Beweis im durch* β *definierten abstrakten Raum, wenn es eine nichtleere Menge* $(\Pi_\mathsf{s}, \Pi_\mathsf{t})$ *von Vorkommen von gemeinsamen Untertermen von* s *und* t *gibt, so daß es eine Ableitung* $\mathsf{s} \vdash_\Phi^* \mathsf{t}$ *gibt, die eine* $\beta\Pi_\mathsf{s}$-repräsentierbare Ableitung *ist.*

Beweis 2.6 *Folgt direkt aus Definition 2.15.* ■

Kapitel 3

Deduktion auf Abstraktionen

In Kapitel 2 wurde die in dieser Arbeit gewählte Form der Abstraktion von Termen und Formeln der Prädikatenlogik erster Stufe formal beschrieben. Das Ziel dieser Abstraktion ist es, in einem gegebenem Gleichheitsproblem solche Unterterme zu ignorieren, die für die Beweisführung als weniger wichtig erachtet werden. Dadurch soll die Beweisfindung erleichtert werden, da sich einerseits auf die Vorkommen von ausgewählten Untertermen konzentriert werden kann und andererseits nicht um störende Einzelheiten gekümmert werden muß. Darauf aufbauend wurden zwei Abstraktionsverfahren angegeben: Die der abstrakten Terme und die der Strukturterme. Auf diesen Abstraktionsebenen werden in diesem Kapitel nun Heuristiken angegeben, um die jeweiligen Abstraktionen zweier Terme anzugleichen.

Den Kern der im folgenden vorgestellten Heuristiken bilden jene, die sich an einer gegebenen Abstraktionsinformation (siehe Abschnitt 3.2.3) orientieren. Zur Berechnung der Abstraktionsinformation wird einerseits eine Heuristik benötigt, die, falls nicht vorgegeben, eine Menge von "interessanten" gemeinsamen Untertermen berechnet, wobei die Aussage, was "interessante" Unterterme sind, im ersten Abschnitt dieses Kapitels präzisiert wird. Zusätzlich wird dort noch eine Heuristik angegeben, mit der die verschiedenen Vorkommen von "interessanten" gemeinsamen Untertermen einander zugeordnet werden können. Da die Zuordnungen von Vorkommen gemeinsamer Unterterme nicht immer gegeben sind, wird hier ein mögliches Verfahren angegeben, wie diese Zuordnungen berechnet werden können. Anschließend werden im zweiten Abschnitt des Kapitels die Heuristiken angegeben, die sich in zwei Arten einteilen lassen: Solche, die direkt auf dem abstrakten Gleichheitsproblem arbeiten, und solche, die dem Problem Rechnung tragen, daß die gewählte Abstraktion zu "stark" von dem ursprünglichen Gleichheitsproblem abstrahiert. Für letztere gibt es eine Heuristik die sich damit beschäftigt, wie das Abstraktionsniveau "sinnvoll reduziert" werden kann.

Die oben geschilderten Heuristiken können sowohl auf der Abstraktionsebene der abstrakten Terme, als auch auf der der Strukturterme verwendet werden. Für die Strukturterme wird im dritten Abschnitt dieses Kapitel noch ein weiteres Verfahren angegeben.

3.1 Motivation

3.1.1 Zuordnung von Vorkommen

Um die in der Einleitung beschriebene Sichtweise auf die Funktionssymbollisten zu den Vorkommen von gemeinsamen Untertermen überhaupt durchführen zu können, müssen die Vorkommen in den beiden Termen eines Gleichheitsproblems paarweise zugeordnen werden, falls eine solche Zuordnung nicht durch einen anderen Teil des Beweisers schon vorgegeben ist. Um die Zuordnung durchführen zu können, muß ein Maß definiert werden, welches angibt, ob die Zuordnung zweier Vorkommen "besser" ist als eine andere. Der Erfolg der Heuristiken, die später vorgestellt werden, hängt wesentlich davon ab, wie gut diese Zuordnungen gemacht wurden. Eine solche Zuordnung ist formal betrachtet eine binäre Relation auf Funktionssymbollisten. Im zweiten Abschnitt dieses Kapitels über die Heuristiken wird dann ein konkretes Maß für die Güte einer Zuordnung angegeben.

3.1.2 Funktionssymbollisten

In der Einleitung wurde bereits informal der Begriff der **Funktionssymbolliste** erläutert. Dieser Begriff wird nun formal definiert. Zur Erinnerung: **Funktionssymbollisten** sind eine Möglichkeit, Termpositionen mit zusätzlichen Informationen anzureichern[1]. Dabei wird außer der Argumentposition zusätzlich noch das Funktionssymbol des nächsthöheren Termes aufgelistet. Beispiel: Sei der Term $\mathsf{f}(\mathsf{g}(\mathsf{x}),\mathsf{y})$ gegeben. Die Termposition zu dem Vorkommen von x ist

$$[1,1]$$

Die zugehörige **Funktionssymbolliste** dieses Vorkommens von x ist:

$$[(\mathsf{f},1),(\mathsf{g},1)]$$

Definition 3.1 (Funktionssymbolliste) *Sei ein abstrakter Term* t *gegeben. Dann ist die Liste*

$$[(\mathsf{f}_1,\mathsf{i}_1),\ldots,(\mathsf{f}_n,\mathsf{i}_n)]$$

genau dann eine Funktionssymbolliste *eines Termes* t, *wenn gilt:*

[1] Eine weitere Möglichkeit Termpositionen zu erweitern ist in [GM93] beschrieben.

- $[i_1, \ldots, i_n] \in \Pi(t)$,
- t *ist der Form* $f_1(t_1, \ldots, t_l)$ *und*
- *Für alle* $2 \leq j \leq n$ *gilt: Der Term* $t_{|[i_1 \ldots i_j]}$ *ist der Form* $f_{j-1}(\ldots)$.

3.1.3 Skelett zweier Funktionssymbollisten

In der Einleitung wurde beschrieben, daß die Vorkommen eines Untertermes in zwei anzugleichenden Termen einander zugeordnet werden, um daraus Rückschlüsse auf die zu tätigenden Veränderungen ziehen zu können, mit dem Ziel, die beiden Terme anzugleichen. Dafür werden Informationen darüber benötigt, was in irgendeiner Art und Weise in zwei gegebenen Funktionssymbollisten schon "gleich" ist. Dafür wird der Begriff des **Skelettes** zweier Funktionssymbollisten eingeführt.

Zunächst wird der Begriff der **Unterliste** eingeführt:

Definition 3.2 (Unterliste) *Gegeben sei eine Funktionssymbolliste*

$$fsl = [(f_1, i_1), \ldots, (f_n, i_n)].$$

Weiterhin sei eine Menge von Indizes $k_j, j = 1 \ldots m$, *mit* $m \leq n$ *gegeben, für die gilt:*

- $\forall\ 1 \leq j \leq m \quad 1 \leq k_j \leq n$
- $\forall\ 1 \leq j \leq m\text{-}1 \quad k_j < k_{j+1}$

Dann ist die Funktionssymbolliste $[(f_{k_1}, i_{k_1}), \ldots, (f_{k_m}, i_{k_m})]$ *eine Unterliste der Funktionssymbolliste* fsl.

Definition 3.3 (Skelett) *Eine Funktionssymbolliste* fsl *ist genau dann* Skelett *zweier Funktionssymbollisten* $fsl1$ *und* $fsl2$, *wenn* fsl *Unterliste von* $fsl1$ *und* $fsl2$ *ist.*

Beispiel:

Die Funktionssymbolliste $fsl1 = [(f, 1)(g, 1)]$ ist **Skelett** der Funktionssymbollisten $fsl2 = [(h, 1)(f, 1)(g, 1)]$ und $fsl3 = [(f, 1)(f, 2)(g, 1)]$. Die Paare aus $fsl2$ und $fsl3$, die nicht in $fsl1$ vorkommen, werden als **Kontexte** bezeichnet. Diese Terminologie ist in Anlehnung an die Begriffe von **Skelett** und **Kontext** in der Farbtermlogik bzw. Rippling (Siehe [Hut91, BvHSI90, BvHHS90, BSvH+93]) entstanden und wird im folgenden Abschnitt formal für Funktionssymbollisten definiert.

3.1.4 Kontexte

Ein Skelett zweier Funktionssymbollisten stellt so etwas wie die Ähnlichkeit dieser beiden Funktionssymbollisten dar. Die Paare von Funktionssymbolen und Argumentpositionen, welche nicht in einem solchen Skelett enthalten sind, stellen die Unterschiede zwischen den beiden Funktionssymbollisten dar. Diese Anteile der jeweiligen Funktionssymbollisten werden als **Kontexte** bezeichnet.

Definition 3.4 (Kontexte) *Gegeben sei eine Funktionssymbolliste* fsl *und sei* skeleton *ein Skelett dieser Funktionssymbolliste. Dann lassen sich die* Kontexte *von* fsl *bezüglich* skeleton *wie folgt definieren:*

- *Falls* fsl *leer ist, gibt es keine* Kontexte.
- *Falls* skeleton *leer ist, dann sind alle Paare aus* fsl Kontexte.
- *Falls* skeleton = [(f,i), skeleton'], *dann ist* fsl *der Form* [fsl1, (f,i), fsl2]. *Die* Kontexte *von* fsl *sind dann alle Paare aus* fsl1 *und alle Kontexte von* fsl2 *bezüglich* skeleton'.

Beispiel:

Die Kontexte der Funktionssymbolliste [(f,1),(g,1),(h,2),(f,1)] bezüglich des Skelettes [(g,1),(f,1)] sind das Paar (f,1) an der ersten Position der Liste und das Paar (h,2).

Anmerkung:

Die Kontexte einer Funktionssymbolliste bezüglich eines Skelettes sind im allgemeinen nicht eindeutig. Beispiel: Die Funktionssymbolliste $[(\mathsf{f},1),(\mathsf{f},1)]$ hat bezüglich ihres Skelettes $[(\mathsf{f},1)]$ zwei verschiedene Kontexte, je nachdem ob das erste oder das zweite Vorkommen von $(\mathsf{f},1)$ als Teil des Skelettes angesehen wird.

3.2 Heuristiken auf abstrakten Termen

3.2.1 Variable gemeinsame Unterterme

Ausgehend von den beiden Termen s und t eines Gleichheitsproblems gilt es, gemeinsame Unterterme von s und t zu bestimmen, die "interessant" sind. Das Problem ist zunächst einmal, daß definiert werden muß, was "interessante" gemeinsame Unterterme sein sollen. Dabei kann zunächst einmal von der Menge aller gemeinsamen Unterterme von s und t ausgegangen werden.

Definition 3.5 (Gemeinsame Unterterme) *Gegeben seien zwei Terme* s *und* t *aus* $\mathsf{T}(\Sigma, \mathcal{V})$. *Dann ist die Menge der* gemeinsamen Unterterme *von* s *und* t *wie folgt definiert:*

$$\mathsf{Common}(\mathsf{s},\mathsf{t}) := \{\mathsf{u} \in \mathsf{T}(\Sigma, \mathcal{V}) \mid \exists \pi_{\mathsf{s}} \in \Pi(\mathsf{s})\ \mathsf{s}_{|\pi_{\mathsf{s}}} = \mathsf{u} \wedge \exists \pi_{\mathsf{t}} \in \Pi(\mathsf{t})\ \mathsf{t}_{|\pi_{\mathsf{t}}} = \mathsf{u}\}$$

Beispiel:

Die gemeinsamen Unterterme von $\mathsf{s(x)} + \mathsf{0}$ und $\mathsf{s(0)} \times \mathsf{s(x)}$ sind

$$\{\mathsf{0}, \mathsf{x}, \mathsf{s(x)}\}$$

An diesem Beispiel wird schon sichtbar, daß nicht alle gemeinsamen Unterterme brauchbar sind, sondern nur die "größten". In dem Beispiel interessieren zunächst einmal eigentlich nur die gemeinsamen Unterterme $\mathsf{0}$ und $\mathsf{s(x)}$, nicht aber der Unterterm x. Dies wäre nicht so, wenn der Unterterm x nochmal vorkommen würde, und zwar nicht in einem Vorkommen von $\mathsf{s(x)}$. Deshalb wird der Begriff der maximalen gemeinsamen Unterterme definiert:

Definition 3.6 (Maximale gemeinsame Unterterme) *Gegeben seien zwei Terme* s *und* t *aus* $\mathsf{T}(\Sigma, \mathcal{V})$. *Dann ist die Menge der* maximalen gemeinsamen Unterterme *von* s *und* t *definiert als die Menge der Terme* u *aus* $\mathsf{Common}(\mathsf{s},\mathsf{t})$, *für die gilt: Es gibt in* s *und* t *jeweils ein Vorkommen von* u, *welches in keinem Vorkommen eines anderen Termes aus* $\mathsf{Common}(\mathsf{s},\mathsf{t})$ *auftritt.*

Beispiel:

Gegeben seien die beiden Terme $(\mathsf{x} + \mathsf{0}) + \mathsf{s(x)} + \mathsf{s(x)}$ und $\mathsf{x} + \mathsf{times}(\mathsf{s}(\mathsf{s}(\mathsf{0})), \mathsf{s(x)})$. Die Menge der maximalen gemeinsamen Unterterme dieser beiden Terme ist

$$\{\mathsf{0}, \mathsf{x}, \mathsf{s(x)}\}$$

An dem letzten Beispiel läßt sich ein weiteres Manko bei der Bestimmung der interessanten gemeinsamen Unterterme erkennen: In der Menge der maximalen gemeinsamen Unterterme taucht der Unterterm $\mathsf{0}$ auf. Dieser Unterterm $\mathsf{0}$ spielt aber keine Rolle für die Angleichung der beiden Terme. Auch würde man nicht behaupten, daß er so etwas wie eine Ähnlichkeit der beiden Terme ausmacht. Interessanter sind in diesem Fall die maximalen gemeinsamen Unterterme, in denen Variablen vorkommen, wie die Unterterme x und $\mathsf{s(x)}$. Dementsprechend wird die Menge der **variablen gemeinsamen Unterterme** definiert.

Definition 3.7 (Variable gemeinsame Unterterme) *Gegeben seien zwei Terme* s *und* t *aus* $\mathsf{T}(\Sigma, \mathcal{V})$. *Dann ist die Menge der* variablen gemeinsamen Unterterme *von* s *und* t *definiert als die Teilmenge der maximalen gemeinsamen Unterterme von* s *und* t, *in denen Variablen vorkommen.*

Mit dieser Definition von variablen gemeinsamen Untertermen erhält man die folgenden Ergebnisse:

- Die interessanten gemeinsamen Unterterme von $(\mathsf{x} + 0) + \mathsf{s}(\mathsf{x}) + \mathsf{s}(\mathsf{x})$ und $\mathsf{x} + \mathsf{times}(\mathsf{s}(\mathsf{s}(0)), \mathsf{s}(\mathsf{x}))$ sind: $\{\mathsf{x}, \mathsf{s}(\mathsf{x})\}$.
- Der von $\mathsf{s}(\mathsf{x}) + 0$ und $\mathsf{s}(0) \times \mathsf{s}(\mathsf{x})$ ist $\mathsf{s}(\mathsf{x})$.

3.2.1.1 Unabhängigkeit von Untertermen

Wie in der Einleitung beschrieben, wird das Hauptaugenmerk auf den Funktionssymbollisten zu den variablen gemeinsamen Untertermen zweier anzugleichender Terme liegen. Dabei kann folgendes Problem entstehen: Gegeben seien zwei interessante Unterterme u und v zweier Terme s und t, wovon einer in dem anderem vorkommt. Dies bedeutet zum Beispiel, daß es eine Termposition p für v gibt, so daß:

$$\mathsf{v}_{|\mathsf{p}} = \mathsf{u}$$

Die Funktionssymbollisten zu den Vorkommen von v in s oder t sind immer auch Präfixe von Funktionssymbollisten zu Vorkommen von u in s oder t. Da aber eine Gleichungsanwendung auf ein Vorkommen von u, welches in einem Vorkommen von v auftritt, dieses Vorkommen von v zerstören würde, sollen diese Vorkommen von u nicht betrachtet werden. Um dieses Problem zu umgehen, wird gefordert, daß die wichtigen Unterterme sich nicht gegenseitig enthalten. Diese Mengen von Untertermen werden dann als **unabhängig** bezeichnet. Anschließend wird ein Trick angegeben, wie dennoch mit Mengen von Untertermen umgegangen werden kann, die nicht unabhängig sind.

Definition 3.8 (Unabhängige Terme) *Eine Menge von Untertermen* M *ist* unabhängig*, genau dann wenn es keine zwei Terme* $\mathsf{u}, \mathsf{v} \in \mathsf{M}$ *gibt, so daß* u *Unterterm von* v. *Ansonsten sind diese Terme* abhängig.

Es kann passieren, daß die variablen gemeinsamen Unterterme nicht unabhängig im Sinne dieser Definition sind. Um dies zu vermeiden, wird für jeden variablen gemeinsamen Unterterm eine neue Konstante eingeführt, die für diesen Unterterm eingesetzt wird. Dies entspricht dem Prinzip der Generalisierung. Diese Menge von neuen Konstanten bildet die neue Menge von variablen[2] gemeinsamen Untertermen und erfüllt trivialerweise die Unabhängigkeitsbedingung. Diese Signaturerweiterung stellt kein logisches Problem dar, da die Signatur für die Zwecke der Abstraktion schon um die Konstante @ erweitert wurde und somit die Begriffe

[2] Die neuen Konstanten enthalten keine Variablen; da sie aber nur ein Ersatz für die ursprünglichen variablen gemeinsamen Unterterme sind, wird dieser Name beibehalten.

Korrektheit und Vollständigkeit auf der abstrakten Termebene sowieso schon losgelöst sind von der Korrektheit und Vollständigkeit auf der konkreten Termebene.

Beispiel:

Gegeben sei die zu beweisende Gleichung

$$\mathsf{x} + \mathsf{s}(\mathsf{x}) + \mathsf{s}(\mathsf{x}) \equiv \mathsf{x} + \mathsf{times}(\mathsf{s}(\mathsf{s}(0)), \mathsf{s}(\mathsf{x}))$$

Die durch das Verfahren aus Abschnitt 3.2.1 bestimmten *abhängigen* Unterterme sind $\{\mathsf{x}, \mathsf{s}(\mathsf{x})\}$. Jetzt werden jeweils für $\mathsf{s}(\mathsf{x})$ und x neue Konstanten $\mathsf{AC1}$ und $\mathsf{AC2}$ generiert, und entsprechend für die ursprünglichen Terme eingesetzt. Dabei muß allerdings auf die Reihenfolge geachtet werden: Zunächst müssen die "größten" Terme ersetzt werden, anschließend die kleineren. Daraus ergibt sich die folgende abstrakte Gleichung:

$$\mathsf{AC2} + \mathsf{AC1} + \mathsf{AC1} \equiv \mathsf{AC2} + \mathsf{times}(@, \mathsf{AC1})$$

Die Menge der variablen gemeinsamen Unterterme ist dann $\{\mathsf{AC1}, \mathsf{AC2}\}$, die die Unabhängigkeitseigenschaft erfüllt.

Im weiteren Verlauf der Arbeit werden diese neu eingeführten Konstantensymbole nur dann explizit verwendet, wenn dies notwendig ist. Ansonsten wird aus Gründen der Übersichtlichkeit auf sie verzichtet.

3.2.2 Zuordnung der Vorkommen

3.2.2.1 Hierarchie auf Funktionssymbolen

Die Eingabesprache des INKA-Systems (siehe [CHS94]) ist speziell auf die Definition von algorithmischen Funktionen ausgerichtet, wodurch sich in kanonischer Art und Weise eine natürliche partielle Ordnung auf den Funktionssymbolen ergibt. Beispiel: Gegeben sei eine Spezifikation Nat der natürlichen Zahlen mit Hilfe der Funktionssymbole $0 : \mathsf{Nat}$, $\mathsf{s}, \mathsf{p} : \mathsf{Nat} \to \mathsf{Nat}$. Die algorithmische Definition des Funktionssymbols $+$ zur Spezifikation der Addition auf natürlichen Zahlen ist im INKA-System wie folgt:

$$\begin{array}{l} \textbf{FUNCTION } +(\mathsf{x}, \mathsf{y} : \mathsf{Nat}) : \mathsf{Nat} = \\ \textbf{IF } \mathsf{x} \equiv 0 \textbf{ THEN } \mathsf{y} \\ \textbf{IF } \mathsf{x} \equiv \mathsf{s}(\mathsf{p}(\mathsf{x})) \textbf{ THEN } \mathsf{s}(+(\mathsf{p}(\mathsf{x}), \mathsf{y})) \end{array}$$

Aus der algorithmischen Funktionsdefinition läßt sich folgende partielle Ordnung $>$ auf den Funktionssymbolen etablieren:

$$0, \mathsf{p}, \mathsf{s} < +$$

Wenn anschließend die Multiplikation auf natürlichen Zahlen algorithmisch definiert wird, wird die partielle Ordnung mit erweitert.

$$\begin{array}{l}\textbf{FUNCTION } \times(\mathsf{x},\mathsf{y}:\mathsf{Nat}):\mathsf{Nat} = \\ \textbf{IF } \mathsf{x} \equiv 0 \textbf{ THEN } 0 \\ \textbf{IF } \mathsf{x} \equiv \mathsf{s}(\mathsf{p}(\mathsf{x})) \textbf{ THEN } +(\times(\mathsf{p}(\mathsf{x}),\mathsf{y}),\mathsf{y})\end{array}$$

Die erweiterte partielle Ordnung ist dann

$$0, \mathsf{p}, \mathsf{s} < + < \times$$

Dabei werden die bezüglich der Ordnung $<$ maximalen Funktionssymbole als die wichtigsten angesehen. Insbesondere können damit heuristische Annahmen gemacht werden, wie zum Beispiel, daß es "einfacher" ist, mit Hilfe einer Gleichung ein maximales Funktionssymbol in einem Term zu eliminieren, als ein nicht maximales. Beispiel: In einer Axiomatisierung der natürlichen Zahlen mit der Addition $+$ gibt es mehr Gleichungen, die das Funktionssymbol $+$ eliminieren, als solche, die das Funktionssymbol des Sukzessors eliminieren.

3.2.2.2 Das Zuordnungsverfahren

Gegeben seien zwei Terme s und t und ein Paar $(\Pi_\mathsf{s}, \Pi_\mathsf{t})$ von Vorkommen von gemeinsamen Untertermen von s und t. Die Funktionssymbollisten zu den Termpositionen Π_s und Π_t von s und t seien durch $\mathsf{FSL}(\mathsf{s}, \Pi_\mathsf{s})$ und $\mathsf{FSL}(\mathsf{t}, \Pi_\mathsf{t})$ gegeben. Das Problem ist nun, welche Funktionssymbolliste aus $\mathsf{FSL}(\mathsf{t}, \Pi_\mathsf{t})$ wird einer gegebenen Funktionssymbolliste aus $\mathsf{FSL}(\mathsf{s}, \Pi_\mathsf{s})$ zugeordnet. Die Heuristik dafür ist, daß jeweils die Funktionssymbollisten einander zugeordnet werden, die den gleichen Unterterm denotieren und die meisten Gemeinsamkeiten haben. Dabei baut der Begriff der Gemeinsamkeit auf dem in Abschnitt 3.1.3 eingeführten Begriff des Skeletts auf. Da es im allgemeinen zu einer gegebenen Funktionssymbolliste aus $\mathsf{FSL}(\mathsf{s}, \Pi_\mathsf{s})$ mehrere Funktionssymbollisten aus $\mathsf{FSL}(\mathsf{t}, \Pi_\mathsf{t})$ geben kann, mit denen sie gleich große Skelette hat, wird die Auswahl weiter eingeschränkt. Hierzu wird die Ordnung auf den Funktionssymbolen verwendet. Aus allen Funktionssymbollisten aus $\mathsf{FSL}(\mathsf{t}, \Pi_\mathsf{t})$, die gleich große Skelette haben mit einer gegebenen Funktionssymbolliste aus $\mathsf{FSL}(\mathsf{s}, \Pi_\mathsf{s})$, werden diejenigen ausgewählt, in deren Skelette die meisten maximalen Funktionssymbole vorkommen und die Anzahl der Kontexte (im Sinne von Abschnitt 3.1.4) am geringsten ist. Bleiben immer noch mehrere Funktionssymbollisten übrig, wird eine beliebige davon zugeordnet.

Beispiel:

Gegeben seien die Terme $\mathsf{x} \times (\mathsf{x} + \mathsf{y})$ und $(\mathsf{x} \times \mathsf{x}) + (\mathsf{x} \times \mathsf{y})$. Betrachtet werden alle Vorkommen der Variablen x und y. Die Zuordnung der Vorkommen ist wie folgt (dabei werden die einzelnen Vorkommen über die entsprechende Funktionssymbolliste identifiziert):

	$\mathsf{x} \times (\mathsf{x} + \mathsf{y})$	$(\mathsf{x} \times \mathsf{x}) + (\mathsf{x} \times \mathsf{y})$
x	$[(\times, 1)]$	$[(+, 1), (\times, 1)]$ oder $[(+, 2), (\times, 1)]$
	$[(\times, 2), (+, 1)]$	$[(+, 1), (\times, 2)]$
y	$[(\times, 2), (+, 2)]$	$[(+, 2), (\times, 2)]$

Die erhaltene Zuordnung der Vorkommen des Untertermes x entspricht in diesem Beispiel der Zuordnung, die intuitiv gemacht worden wäre.

3.2.2.3 Bemerkungen

Das letzte Verfahren zur Berechnung der Zuordnung von Vorkommen von gemeinsamen Untertermen wird immer wieder gebraucht, um die Abstraktionsinformation zu einem gegebenem abstrakten Gleichheitsproblem zu berechnen. Das erste Verfahren zur Bestimmung von interessanten gemeinsamen Untertermen wird nur dann verwendet, wenn nicht bestimmte gemeinsame Unterterme vorgegeben sind. Diese können sowohl durch einen Benutzer oder durch einen anderen Teil des Beweisers vorgegeben werden, wenn sie die abstrakte Planung aufrufen. Die abstrakte Planung wird im allgemeinen immer eine bestimmte Menge von Vorkommen von gemeinsamen Untertermen erhalten, bezüglich derer sie einen Beweisplan berechnen soll. Deshalb spielt die stark heuristisch geprägte und etwas willkürlich aussehende Bestimmung von gemeinsamen Untertermen aus Abschnitt 3.2.1 in der Praxis kaum eine Rolle. Auch wird die Zuordnung der gemeinsamen Unterterme meist von den anderen Heuristiken vorgegeben, so daß die hier vorgeschlagene Art der Zuordnung wegfällt oder nur eingeschränkt benötigt wird.

3.2.3 Abstraktionsinformation

Für die Heuristiken werden Informationen darüber benötigt, welche Vorkommen von Untertermen in zwei abstrakten Termen s und t einander zugeordnet sind, und welches Skelett der Funktionssymbollisten zweier zugeordneter Vorkommen betrachtet wird. Diese gesammelte Information wird als **Abstraktionsinformation** bezeichnet. Diese ist implementierungstechnisch eine Liste von Tripeln über Funktionssymbollisten. Dabei ist die erste Funktionssymbolliste des Tripels diejenige, die zu dem Vorkommen eines Unterterms in s gehört; die zweite, diejenige die zu dem zugeordneten Vorkommen des gleichen Unterterms in t gehört, und die letzte das betrachtete Skelett dieser beiden Funktionssymbollisten.

Definition 3.9 (Abstraktionsinformation) *Gegeben seien zwei abstrakte Terme* s *und* t *bezüglich einer Menge* $(\Pi_\mathsf{s}, \Pi_\mathsf{t})$ *von Vorkommen von gemeinsamen Untertermen. Die* Abstraktionsinformation *zu diesen beiden Termen* s *und* t *ist eine Menge von Tripeln* (fsl1, fsl2, skeleton)*, die die folgenden Eigenschaften erfüllen:*

- fsl1 *ist Funktionssymbolliste von* s *bezüglich eines Vorkommens eines Termes* u *aus* Π_s,
- fsl2 *ist Funktionssymbolliste von* t *bezüglich eines Vorkommens des gleichen Termes* u *aus* Π_t *und*
- skeleton *ist ein größtes Skelett von* fsl1 *und* fsl2.

Beispiel:

Gegeben seien die beiden abstrakten Terme aus $\mathsf{AT}(\Sigma, \mathcal{V})$

$$\mathsf{f}(\mathsf{g}(\mathsf{t}), \mathsf{h}(\mathsf{s}, @)) \quad \text{und} \quad \mathsf{h}(\mathsf{f}(\mathsf{s}, @), \mathsf{t})$$

Die **Abstraktionsinformation** dieser beiden Terme ist

	$\mathsf{f}(\mathsf{g}(\mathsf{t}), \mathsf{h}(\mathsf{s}, @))$	$\mathsf{h}(\mathsf{f}(\mathsf{s}, @), \mathsf{t})$	Skelett
t	$[(\mathsf{f}, 1), (\mathsf{g}, 1)]$	$[(\mathsf{h}, 2)]$	$[]$
s	$[(\mathsf{f}, 2), (\mathsf{h}, 1)]$	$[(\mathsf{h}, 1), (\mathsf{f}, 1)]$	$[(\mathsf{h}, 1)]$

Die Funktion, die zu zwei gegebenen abstrakten Termen s, t und der Menge M der betrachteten Vorkommen von gemeinsamen Untertermen von s und t die Abstraktionsinformation liefert, ist die Funktion AI.

3.2.4 Heuristiken

Dieser Abschnitt beschäftigt sich mit der Angleichung zweier abstrakter Terme. Die hierfür vorgesehenen Heuristiken lassen sich grob in zwei Arten unterteilen: Zum einen gibt es vorkommenorientierte Heuristiken, die versuchen, die jeweilige Anzahl der Vorkommen eines gemeinsamen Untertermes in den beiden abstrakten Termen anzugleichen. Zum anderen gibt es Heuristiken, die mit Hilfe der in der Abstraktionsinformation repräsentierten Informationen versuchen, die beiden abstrakten Terme anzugleichen.

Beiden Arten von Heuristiken ist gemeinsam, daß sie abstrakte, gerichtete Gleichungen verwenden. Diese sind nach noch zu erläuternden Kriterien in der Datenbasis abgelegt. Die Anwendung einer abstrakten Gleichung erfolgt nach festgelegten Kriterien, die im folgenden Abschnitt beschrieben werden.

3.2.4.1 Anwendung einer abstrakten Gleichung

Die Anwendung einer abstrakten, gerichteten Gleichung erfolgt in drei Schritten: Im ersten Schritt wird versucht, die Gleichung durch kleine Modifikationen anwendbar zu machen. Im zweiten wird sie mit Hilfe der Kalkül-Regel angewendet (siehe Abschnitt 2.1.2). Im letzten Schritt werden noch zwei Tests durchgeführt: Ersterer überprüft, ob das neue Gleichheitsproblem bezüglich einer noetherschen Ordnung kleiner ist als das ursprüngliche. Diese noethersche Ordnung[3] $<_{\mathsf{GO}}$ wird formal in Kapitel 5 eingeführt und garantiert somit die Terminierung des aus der Kombination aller Heuristiken resultierenden Gesamtverfahrens. Für jede durch eine Heuristik durchgeführte Transformation eines Gleichheitsproblems $\mathsf{s} \equiv \mathsf{t}$ in ein Gleichheitsproblem $\mathsf{s}' \equiv \mathsf{t}'$, gilt dann:

$$\mathsf{s}' \equiv \mathsf{t}' <_{\mathsf{GO}} \mathsf{s} \equiv \mathsf{t}$$

Deshalb gibt es an dieser Stelle diesen ersten Test, der die Wahrung der Terminierungsordnung überprüft. Der zweite Test überprüft eine weitere Einschränkung, die im Gegensatz zur ersten rein heuristische Gründe hat. Auf diese Einschränkung wird in Abschnitt 3.2.4.1 genauer eingegangen.

Anpassung der Funktionssymbollisten Bei der Anwendung einer gerichteten Gleichung, wird zunächst überprüft, ob die gerichtete Gleichung auch die gewünschte Veränderung bringt. Es werden die Funktionssymbollisten des Vorkommens eines Unterterms des aktuellen Gleichheitsproblems und die Funktionssymbolliste des Vorkommens eines Unterterms der anzuwendenden gerichteten Gleichung, bezüglich dessen diese klassifiziert wurde, miteinander verglichen. Passen beide zueinander, wird mit dem nächsten Schritt weitergemacht. Sollten diese aber nicht passen, wird versucht mittels einfacher Modifikationen diese beiden anzupassen.

Die hierzu verwendete Heuristik ist die folgende: Unterscheiden sich die beiden Funktionssymbollisten nur um Argumentpositionen, wird mit Hilfe von speziellen gerichteten Gleichungen versucht, diese beiden Funktionssymbollisten anzugleichen. Bei diesen speziellen gerichteten Gleichungen handelt es sich in grober Näherung um eine Art Gleichungen, die die Kommutativität repräsentieren. Ein Beispiel einer solchen gerichteten Gleichung ist

$$\mathsf{f}(\mathsf{g}(\mathsf{x},\mathsf{y}),\mathsf{z}) \to \mathsf{f}(\mathsf{g}(\mathsf{y},\mathsf{x}),\mathsf{z})$$

Diese speziellen gerichteten Gleichungen werden gezielt ausgesucht, da an diesen gerichteten Gleichungen Informationen darüber enthalten sind, welche Argumentpositionsveränderungen sie durchführen. In dem obigen Beispiel würde zum

[3] GO steht für "Gesamt-Ordnung".

Beispiel stehen, daß das durch die Funktionssymbolliste $[(\mathsf{f}, 1), (\mathsf{g}, 1)]$ denotierte Argument des Vorkommens von g in das zweite Argument von g "verschoben" wird. Für die Anwendung dieser "argumentverschiebenden" gerichteten Gleichungen müssen sowohl die Wahrung der Terminierungsordnung, als auch des abschließenden heuristischen Tests nicht überprüft werden. Gelingt die Anpassung der Funktionssymbollisten, wird versucht, die eigentliche gerichtete Gleichung anzuwenden. Ansonsten bricht das Verfahren an dieser Stelle mit einem Mißerfolg ab.

Eigentliche Anwendung einer abstrakten gerichteten Gleichung Die eigentliche Anwendung einer gerichteten Gleichung wird durch die in Abschnitt 2.1.2 beschriebene Kalkül-Regel durchgeführt. Dabei kann folgendes Problem auftreten: Angenommen die gerichtete Gleichung

$$\mathsf{f}(\mathsf{x}, \mathsf{g}(\mathsf{y})) \to \mathsf{f}(\mathsf{g}(\mathsf{x}), \mathsf{y})$$

soll auf den abstrakten Term $\mathsf{f}(\mathsf{a}, @)$ angewendet werden. Dies mißlingt, da @ und $\mathsf{g}(\mathsf{y})$ nicht unifizierbar sind. Obige Gleichung kann aber, wenn anstatt bezüglich der Variablen x und y nur bezüglich der Variablen x abstrahiert wird, zu der gerichteten Gleichung

$$\mathsf{f}(\mathsf{x}, @) \to \mathsf{f}(\mathsf{g}(\mathsf{x}), @)$$

abstrahiert werden, welche anwendbar ist. Deshalb gibt es eine Routine, die im Falle eines Mißerfolgs bei der Anwendung der gegebenen gerichteten Gleichung versucht diesen zu abstrahieren. Dabei wird die Information berücksichtigt, bezüglich welchem Vorkommen eines Unterterms die gerichtete Gleichung klassifiziert wurde. Dieser Unterterm darf dann nicht "wegabstrahiert" werden. Anschließend wird versucht diese abstraktere gerichtete Gleichung anzuwenden. Gelingt dies, werden noch die im folgendem Abschnitt beschriebenen Tests durchgeführt. Ansonsten bricht das Verfahren ab.

Die anschließenden Tests Die beiden eingebauten Tests haben zwei Funktionen: Der erste überprüft die Einhaltung der noch zu definierenden noetherschen Ordnung $<_{\mathsf{GO}}$. Der zweite ist rein heuristischer Natur, um überflüssige Schritte oder "Sackgassen" zu vermeiden.

Überprüfung der Terminierungsordnung Die Terminierungsordnung wird erst in Abschnitt 5.2 formal eingeführt. Deshalb wird an dieser Stelle nur auf ein implementierungstechnisches Detail eingegangen. Wie in den folgenden Abschnitten zu sehen sein wird, brauchen manche Heuristiken eine temporäre Ausschaltung der Terminierungsordnung. Aus diesem Grund gibt es 4 Stufen von Überprüfungen der Terminierungsordnung. Diese unterscheiden sich nur in der Schärfe, mit der die Terminierungsordnung überprüft wird. Schlägt die Überprüfung fehl, bricht das Anwendungsverfahren ab. Ansonsten ist es endlich geschafft und der durch die gerichtete Gleichung veränderte Term wird zurückgeliefert.

Heuristischer Test Diese Heuristik hat einzig und allein zum Ziel potentiell überflüssige Schritte oder solche, die in Sackgassen führen könnten, zu vermeiden. Die Motivation ist, wie der Name schon sagt rein heuristischer Natur. Um diese zu erklären, wird wieder von der durch die algorithmische Funktionsdefinition gegebene Ordnung auf Funktionssymbolen ausgegangen (Siehe Abschnitt 3.2.2.1). Gegeben sei das abstrakte Gleichheitsproblem

$$\mathsf{s} \equiv \mathsf{t}$$

Zu diesem Gleichheitsproblem kann so etwas wie eine Unter- und eine Obergrenze bezüglich der Funktionssymbolordnung angeben werden; das heißt, es können eine Menge von **minimalen** Funktionssymbolen und eine Menge von **maximalen** Funktionssymbolen jeweils respektive der Menge aller Funktionssymbole in s und t angeben werden. Die Idee ist nun, daß der zu findende Beweis zunächst oberhalb der unteren Schranke gesucht werden soll. Dies bedeutet, daß die durch eine gerichtete Gleichung neu eingeführten Funktionssymbole nicht kleiner als die alten minimalen Funktionssymbole sein dürfen. Der Grund für diese Heuristik ist, den Beweis für ein gegebenes Gleichheitsproblem auf einer "hohen" Ebene zu suchen; "hoch" versteht sich dabei relativ zur partiellen Ordnung auf den Funktionssymbolen. Scheitert dieser **Unterschranken-Test**, bricht die Anwendungsroutine ab.

3.2.4.2 Anzahl der Vorkommen eines Termes

Die in dieser Arbeit betrachtete Abstraktion orientiert sich an gemeinsamen Untertermen zweier Terme s und t eines Gleichheitsproblems. Die Anzahl der Vorkommen eines solchen Untertermes u in den beiden Termen s und t sind im allgemeinen ungleich. Ein Beispiel dafür ist der Unterterm x in den beiden Termen

$$\mathsf{x} \times (\mathsf{x} + \mathsf{y}) \quad \text{und} \quad (\mathsf{x} \times \mathsf{x}) + (\mathsf{x} \times \mathsf{y})$$

Die Grundidee dieser Heuristik ähnelt etwas der "Collection" Heuristik aus dem PRESS System (siehe [BW81, SBB⁺82]). Allerdings war dort die Idee, die Anzahl der Vorkommen einer Variablen zu reduzieren, während in der hier vorgestellten Heuristik versucht wird, die Anzahl gemeinsamer Unterterme anzugleichen.

Die Differenz in der Anzahl der Vorkommen eines gemeinsamen Unterterms ist somit ein mögliches Maß für die Ungleichheit zweier Terme. Deshalb gibt es diese Heuristik, die gezielt versucht, diese Differenzen zu minimieren. Dafür werden die abstrakten, gerichteten Gleichungen $\mathsf{q} \rightarrow \mathsf{t}$ geordnet nach solchen, die die Anzahl der Vorkommen von gemeinsamen Untertermen von q und r erhöhen oder reduzieren. Die gerichtete, abstrakte Gleichung

$$\mathsf{x} \times (\mathsf{y} + \mathsf{z}) \rightarrow (\mathsf{x} \times \mathsf{y}) + (\mathsf{x} \times \mathsf{z})$$

erhöht die Anzahl der Vorkommen der Variablen x. Die umgedrehte Gleichung reduziert die Anzahl der Vorkommen der Variablen x.

Kommt ein Unterterm u häufiger in dem Term s als in dem Term t vor, können zwei verschiedene Aktionen versucht werden: Entweder es wird eine vorkommenreduzierende Gleichung $\mathsf{q} \rightarrow \mathsf{r}$ auf den Term s angewendet oder eine vorkommenerhöhende Gleichung auf den Term t. Dabei gibt es zwei Einschränkungen: Der Unterterm v der Gleichung $\mathsf{q} \rightarrow \mathsf{r}$, deren Vorkommen reduziert oder erhöht werden, muß mit dem Unterterm u von s und t "gematched" werden können, d. h. es muß eine Substitution σ geben, so daß gilt:

$$\mathsf{u} = \sigma(\mathsf{v})$$

Dies schränkt die Auswahl der Gleichungen zusätzlich ein. Eine weitere hilfreiche Einschränkung ist folgende: Die Vorkommen von v müssen mit den Vorkommen von u bei der Anwendung übereinstimmen. Dadurch ergibt sich eine Restriktion durch die Positionen in s oder t, an denen die Gleichung angewendet wird.

Gemäß diesen Einschränkungen wird für das Gleichheitsproblem

$$\mathsf{x} \times (\mathsf{x} + \mathsf{y}) \equiv (\mathsf{x} \times \mathsf{x}) + (\mathsf{x} \times \mathsf{y})$$

die Gleichung $\mathsf{u} \times (\mathsf{v} + \mathsf{w}) \rightarrow (\mathsf{u} \times \mathsf{v}) + (\mathsf{u} \times \mathsf{w})$ selektiert. Deren Anwendung auf den linken Term löst in diesem Fall schon das Gleichheitsproblem:

$$(\mathsf{x} \times \mathsf{x}) + (\mathsf{x} \times \mathsf{y}) \equiv (\mathsf{x} \times \mathsf{x}) + (\mathsf{x} \times \mathsf{y})$$

Diese Heuristik ist allerdings sehr grob, da sie sich nicht an den in den Termen vorkommenden Funktionssymbolen orientiert, über die aber aufgrund der natürlichen Ordnung einige Information vorliegen. Insbesondere ist sie stark abhängig von den verwendeten gemeinsamen Untertermen: Wurden hier nicht die für diese Heuristik "richtigen" ausgewählt, bringt der Einsatz dieser Heuristik keinen Vorteil. Dies ist zum Beispiel in folgendem Gleichheitsproblem der Fall:

$$\mathsf{x} + \mathsf{s}(0) \equiv \mathsf{s}(\mathsf{x})$$

3.2.4.3 Skelettorientierte Heuristiken

Die gemeinsamen Unterterme spielen bei der betrachteten Abstraktion die zentrale Rolle, da sie die Ähnlichkeit zweier Terme s und t eines Gleichheitsproblems ausdrücken. Es gilt also die beiden Terme s und t unter Erhaltung der gemeinsamen Unterterme anzugleichen. Eine weitere Heuristik kann also sein, die Funktionssymbollisten, welche zu den jeweiligen Untertermen führen, anzugleichen. Dieser Heuristik liegt folgende Beobachtung zu Grunde: Wenn die Gleichheit zweier Terme s und t bewiesen werden kann, und die Annahme gemacht wird, daß die ursprünglich gewählten gemeinsamen Unterterme auch in dem Term r, in den s und t mit Hilfe von Gleichungen überführt werden können, enthalten sind, dann müssen die Funktionssymbollisten zu den Untertermen angeglichen werden. Dies bedeutet, daß gezielt solche Gleichungen ausgewählt werden, die zwei gegebene Funktionssymbollisten ähnlicher machen. Das Ähnlichkeitskriterium ist hierbei der Begriff des Skelettes zweier Funktionssymbollisten (siehe 3.1.3). An dieser Stelle kommt die in Abschnitt 3.2.3 beschriebene Abstraktionsinformation ins Spiel: In der Abstraktionsinformation zu einem gegebenen Gleichheitsproblem sind alle für diese Zwecke benötigten Informationen enthalten: Die verschiedenen Vorkommen der gemeinsamen Unterterme, die einander zugeordnet sind, sowie deren gemeinsames Skelett.

Es gibt verschiedene Heuristiken, die versuchen die Skelette zu vergrössern, Kontexte (im Sinne von Abschnitt 3.1.4) zu eliminieren oder zu verschieben. Letzteres ähnelt stark den Heuristiken, die im Rippling-Verfahren (siehe [Hut91, BvHSI90, BvHHS90, BSvH+93]) verwendet werden. Die implementierten Heuristiken werden im folgenden vorgestellt.

Kontextverschiebende Heuristiken Zu jedem abstrakten Gleichheitsproblem $\mathsf{s} \equiv \mathsf{t}$ gibt es eine Abstraktionsinformation. In dieser Abstraktionsinformation sind für jedes Paar von zugeordneten Vorkommen eines gemeinsamen Unterterms die entsprechenden Funktionssymbollisten, sowie das betrachtete gemeinsame Skelett enthalten. Diese Einzelinformationen werden als Tripel

$$(\mathsf{fsl_L}, \mathsf{fsl_R}, \mathsf{skeleton})$$

repräsentiert. An dieser Stelle sei noch angemerkt, daß Vorkommen, die nicht zugeordnet sind, auch nicht in der Abstraktionsinformation repräsentiert werden.

Zu jedem Tripel $(\mathsf{fsl_L}, \mathsf{fsl_R}, \mathsf{skeleton})$ können die vorkommenden Kontexte im Sinne von Abschnitt 3.1.4 in den Funktionssymbollisten $\mathsf{fsl_L}$ und $\mathsf{fsl_R}$ bezüglich des Skelettes $\mathsf{skeleton}$ berechnet werden. Bei den Kontexten handelt es sich genaugenommen um Positionen in den Funktionssymbollisten $\mathsf{fsl_L}$ und $\mathsf{fsl_R}$. Zum Beispiel: Gegeben sei wieder das obige (leicht veränderte) Gleichheitsproblem

$$(\mathsf{x} \times \mathsf{x}) + (\mathsf{x} \times \mathsf{y}) \equiv \mathsf{x} \times (\mathsf{y} + \mathsf{x})$$

Betrachtet werden die folgenden, durch die Funktionssymbollisten eindeutig bestimmten, einander zugeordneten Vorkommen des Untertermes x :

$$[(+,1),(\times,2)] \quad \text{und} \quad [(\times,2),(+,2)]$$

Bezüglich des Skelettes $[(\times,2)]$ ist der Kontext in $[(+,1),(\times,2)]$ das Paar $(+,1)$ an der ersten Position der Funktionssymbolliste; der Kontext in der Funktionssymbolliste $[(\times,2),(+,2)]$ ist das Paar $(+,2)$ an der zweiten Position der Funktionssymbolliste.

Die kontextverschiebenden Heuristiken versuchen, ausgehend von den Informationen über die Kontexte, jene entweder nach links in den Listen zu verschieben oder zu eliminieren. Dabei dürfen die vorhandenen Skelette nicht zerstört werden. Wird ein Kontext in einer Funktionssymbolliste nach links verschoben, entspricht dies einer Verschiebung des Kontextes nach außen in dem entsprechendem abstrakten Term. Dies wird später noch an einem Beispiel veranschaulicht.

Die gerichteten, abstrakten Gleichungen im Beweiser werden nach den benötigten Kriterien sortiert: Zum einen gibt es die gerichteten abstrakten Gleichungen, die Kontexte nach außen schieben. Zum anderen gibt es diejenigen, die Kontexte eliminieren. Die für die Lösung des obigen Problems benötigte Gleichung

$$\mathsf{x} \times (\mathsf{y} + \mathsf{z}) \equiv (\mathsf{x} \times \mathsf{y}) + (\mathsf{x} \times \mathsf{z})$$

wird aus folgenden Gründen als kontexthochschiebende gerichtete Gleichung

$$(\mathsf{x} \times \mathsf{y}) + (\mathsf{x} \times \mathsf{z}) \rightarrow \mathsf{x} \times (\mathsf{y} + \mathsf{z})$$

eingetragen: Die Abstraktionsinformation dieser Gleichung bezüglich der gemeinsamen Unterterme x, y und z hat folgende Gestalt:

	$(\mathsf{x} \times \mathsf{y}) + (\mathsf{x} \times \mathsf{z})$	$\mathsf{x} \times (\mathsf{y} + \mathsf{z})$	Skelett
x	$[(+,1),(\times,1)]$	$[(\times,1)]$	$[(\times,1)]$
y	$[(+,1),(\times,2)]$	$[(\times,2),(+,1)]$	$[(\times,2)]$
z	$[(+,2),(\times,2)]$	$[(\times,2),(+,2)]$	$[(\times,2)]$

In der Abstraktionsinformation zu dem Unterterm y ist zu sehen, daß der Kontext bezüglich des Skelettes $[(\times,2)]$ nach links in der Liste verschoben wird. Somit kann diese Gleichung als kontexthochschiebende Gleichung bezüglich des Vorkommen von y eingetragen werden. Gleichzeitig wird diese Gleichung aber auch als kontexteliminierende Gleichung bezüglich des Vorkommen von x eingetragen. Daran läßt sich schon das generelle Problem erkennen, daß eine Gleichung, wenn sie angewendet wird, unangenehme Nebeneffekte haben kann.

Die hinter der Heuristik der Verschiebung von Kontexten nach außen, bzw. deren Eliminierung steckende Idee ist die folgende: Indem die Kontexte, die um die ursprünglichen Unterterme herum stehen, verschwinden, sei es durch Verschiebung nach außen oder Eliminierung, "vergrößern" sich die gemeinsamen Unterterme. Die ursprünglichen gemeinsamen Unterterme bleiben zwar erhalten, nur wird um die Vorkommen dieser Unterterme herum in den abstrakten Termen eine Termstruktur geschaffen, die sukzessive immer ähnlicher wird. Ziel ist es, daß immer größere variable gemeinsame Unterterme entstehen. Insofern unterstützt das Verfahren eine Art *inner-out reasoning*, was im allgemeinen eine schwer zu handhabende Technik ist.

Die Ähnlichkeit zum Rippling-Verfahren ist unverkennbar. Dennoch gibt es einen zentralen Unterschied: Während beim Rippling-Verfahren ein Skelett wieder ein Term sein muß, ist dies in diesem Ansatz im allgemeinen nicht der Fall. Würden die einzelnen in der Abstraktionsinformation zu zwei gegebenen Termen enthaltenen Skelette in den beiden Termen markiert werden, entstände dabei im allgemeinen nur ein "partieller" Term. Dies hat Vorteile, da es leichter ist ein Skelett auf den einander zugeordneten Funktionssymbollisten zu zwei Termen zu finden als ein Skelett im Sinne von Rippling. Dies wird in dem folgendem Beispiel sichtbar: Seien die beiden Terme

$$\mathsf{length}(\mathsf{append}(\mathsf{x},\mathsf{y})) \quad \text{und} \quad \mathsf{length}(\mathsf{append}(\mathsf{y},\mathsf{x}))$$

gegeben. Im Sinne von Rippling gibt es für diese beiden Terme nur zwei, sich gegenseitig ausschließende, Skelette:

$$\mathsf{length}(\mathsf{x}) \quad \text{und} \quad \mathsf{length}(\mathsf{y})$$

Im Sinne der hier verwendeten Definition eines Skelettes, sind die beiden Rippling-Skelette gleichzeitig in der Abstraktionsinformation repräsentiert, was auch der intuitiven Analyse der Ähnlichkeit der beiden Terme näher kommt.

Es gibt aber auch mehr oder weniger schwerwiegende Nachteile: Die Charakterisierung der gerichteten Gleichungen ist, wie oben schon angedeutet, nicht so eindeutig wie im Rippling-Verfahren. Dort hat eine Gleichung, bezÜglich eines gegebenen Skelettes, entweder die Eigenschaft kontexthochschiebend, kontexteliminierend, usw... zu sein, nicht aber mehrere Eigenschaften auf einmal. Dies ist allerdings der Fall in diesem Verfahren, wie es sich an der oben betrachteten gerichteten Gleichung

$$(\mathsf{x} \times \mathsf{y}) + (\mathsf{x} \times \mathsf{z}) \to \mathsf{x} \times (\mathsf{y} + \mathsf{z})$$

gezeigt hat: Diese ist bezüglich eines gemeinsamen Untertermes kontexteliminierend und vorkommenreduzierend, bezüglich der beiden anderen Unterterme kontexthochschiebend.

Dies ist zunächst nur ein unschöner Effekt in Hinblick auf die Speicherverwaltung des Beweissystems, dem aber durch geschickte Repräsentationen von gerichteten Gleichungen entgegen gewirkt werden kann. Ein weitaus gravierenderer Nachteil ist der, daß die Nebeneffekte bei der Anwendung einer gerichteten Gleichung nur sehr schwer abschätzbar sind: Wird die Charakterisierung so stark eingeschränkt, daß eine entsprechend charakterisierte gerichtete Gleichung keine oder praktisch keine Nebeneffekte mehr hat, wird nur ein Bruchteil der benötigten Gleichungen erkannt. Diese Problematik ist allerdings ein generelles Problem für alle in dieser Arbeit vorgestellten Arten von Charakterisierungen. Dies ist bei der Implementierung so gelöst worden, daß nach jeder Gleichungsanwendung überprüft wird, ob das neue Gleichheitsproblem bezüglich der noetherschen Ordnung $<_{\mathsf{GO}}$ kleiner ist als das ursprüngliche. Daraus ergibt sich ein Design-Merkmal für die Ordnung $<_{\mathsf{GO}}$: Sie muß in irgendeiner Art eine Vergrößerung des Skelettes berücksichtigen, bzw. eine Verringerung der Kontexte. Auf die genauen Einzelheiten wird bei der Definition der Ordnung in Kapitel 5 noch näher eingegangen. Mit dieser Lösung lassen sich dann schon relativ gute Erfolge erzielen.

Skelettverändernde Heuristiken Für die im obigen Abschnitt beschriebene Heuristik gibt es bestimmte Problemsituationen, in denen sie versagt. Dies sind Problemsituationen, die manchen Problemsituationen des Rippling-Verfahrens ähnlich sind. Ein Beispiel ist, wenn ein bestimmter Kontext nicht mehr verschoben oder eliminiert werden kann, ohne das Skelett zu verändern. Hierfür wurden einige Ansätze im Rahmen von Rippling entwickelt, sogenannte *unblocking-strategies* (siehe [BI95]). Die im folgenden beschriebene Heuristik ist eine solche *unblocking-strategy* für die kontextverschiebende Heuristik aus dem vorherigen Abschnitt. Das folgende Beispiel soll eine typische Situation vorstellen, in der das Skelett *kontrolliert* verändert werden muß. Gegeben sei ein abstraktes Gleichheitsproblem

$$\mathsf{f}(\mathsf{g}(\mathsf{a},\mathsf{b})) \equiv \mathsf{f}(\mathsf{g}(\mathsf{a},\mathsf{h}(\mathsf{b})))$$

Weiterhin seien eine Gleichung $\mathsf{f}(\mathsf{g}(\mathsf{x},\mathsf{y})) \equiv \mathsf{g}(\mathsf{f}(\mathsf{y}),\mathsf{f}(\mathsf{x}))$ und eine den Kontext $(\mathsf{h},1)$ eliminierende Gleichung $\mathsf{f}(\mathsf{h}(\mathsf{x})) \equiv \mathsf{f}(\mathsf{x})$ gegeben. Die Abstraktionsinformation des Gleichheitsproblems beinhaltet unter anderem für das Vorkommen von b folgenden Eintrag:

	$\mathsf{f(g(a,b))}$	$\mathsf{f(g(a,h(b)))}$	Skelett
b	$[(\mathsf{f},1),(\mathsf{g},2)]$	$[(\mathsf{f},1),(\mathsf{g},2),(\mathsf{h},1)]$	$[(\mathsf{f},1),(\mathsf{g},2)]$

Die genannte Gleichung, die den Kontext $(\mathsf{h},1)$ eliminieren könnte, ist nicht anwendbar. Schlimmer noch ist, daß, um sie anwendbar zu machen, das in der Abstraktionsinformation enthaltene Skelett zerstört werden müßte.

Für solche Situationen wurde folgende Heuristik entwickelt: Das Problem besteht darin auf einen Term s eines Gleichheitsproblems $\mathsf{s} \equiv \mathsf{t}$ an einer Termposition π eine gerichtete Gleichung $\mathsf{u} \rightarrow \mathsf{v}$ anzuwenden. Diese Gleichung würde einen Kontext in s eliminieren. Allerdings würde sie dabei auch das Skelett von s und t zerstören. Um dieses Problem zu lösen werden die Terme u und $\mathsf{s}_{|\pi}$ miteinander verglichen. Genauer wird die Funktionssymbolliste $\mathsf{fsl}_{\mathsf{s}_{|\pi}}$ zu dem Unterterm von $\mathsf{s}_{|\pi}$, in der der zu eliminierende Kontext enthalten ist, verglichen mit der Funktionssymbolliste $\mathsf{fsl}_{\mathsf{u}}$ zu dem Vorkommen von u, bezüglich dem die Gleichung als kontexteliminierend kategorisiert wurde. Wenn die Funktionssymbolliste $\mathsf{fsl}_{\mathsf{u}}$ Skelett von $\mathsf{fsl}_{\mathsf{s}_{|\pi}}$ ist, dann wird versucht die Kontexte in $\mathsf{fsl}_{\mathsf{s}_{|\pi}}$ nach außen zu schieben. Hierfür dürfen nur skeletterhaltende gerichtete Gleichungen verwendet werden. In dem Beispiel oben würde die gerichtete Gleichung $\mathsf{f(g(x,y))} \rightarrow \mathsf{g(f(y),f(x))}$ auf $\mathsf{f(g(a,h(b)))}$ angewendet werden. Daraus resultiert der Term $\mathsf{g(f(h(b)),f(a))}$, auf den die gerichtete Gleichung $\mathsf{f(h(x))} \rightarrow \mathsf{f(x)}$ anwendbar ist.

Durch den Versuch, die Gleichung $\mathsf{u} \rightarrow \mathsf{v}$ anwendbar zu machen, wurde aber das ursprüngliche Skelett von s und t zerstört. Die Stellen, an denen es zerstört wurde, sind durch einen Vergleich des alten und des neuen Skelettes zu lokalisieren; An dieser Stelle lohnt sich die Verwendung von Funktionssymbollisten. Durch den Vergleich der Funktionssymbollisten werden die Termpositionen der ursprünglichen Skelettteile in t erhalten. In dem Beispiel oben wird die Information erhalten, daß das Paar $(\mathsf{g},2)$ der Funktionssymbolliste $[(\mathsf{f},1),(\mathsf{g},2)]$ des linken Termes ursprünglich zum Skelett gehörte. Nun wird versucht die Gleichungen, die auf den Term $\mathsf{s}_{|\pi}$ angewendet wurden, an den entsprechenden Stellen auf den Term t anzuwenden. In dem Beispiel würde die gerichtete Gleichung $\mathsf{f(g(x,y))} \rightarrow \mathsf{g(f(y),f(x))}$ auf den Term $\mathsf{f(g(a,b))}$ an der Termposition $[]$ angewendet werden. Daraus resultiert der Term $\mathsf{g(f(b),f(a))}$.

Gelingt die Anwendung auf t sämtlicher auf s angewendeter Gleichungen, werden die beiden neuen Terme s' und t' miteinander verglichen. Wenn das Skelett dieser beiden Terme genauso groß wie das ursprüngliche ist, wird die ursprünglich ausgewählte Gleichung $\mathsf{u} \rightarrow \mathsf{v}$ an der entsprechenden Termposition auf s' angewendet. In dem Beispiel wurde nach der Anpassung der Skelette das Gleichheitsproblem $\mathsf{g(f(b),f(a))} \equiv \mathsf{g(f(h(b)),f(a))}$ erhalten. Auf den rechten Term wird dann die Gleichung $\mathsf{f(h(x))} \rightarrow \mathsf{f(x)}$ an der Termposition $[1]$ angewendet. Für das daraus resultierende Gleichheitsproblem $\mathsf{g(f(b),f(a))} \equiv \mathsf{g(f(b),f(a))}$ muß gelten, daß es bezüglich der noch zu definierenden Ordnung $<_{GO}$ kleiner ist als das ursprüngliche.

In dem Beispiel ist das neue Gleichheitsproblem schon in gelöster Form, was in der Regel aber nicht der Fall ist, wie an den in Kapitel 6 aufgeführten Beispielen gezeigt werden wird.

Diese Heuristik hält sich an die in Kapitel 5 angegebene Ordnung. Zu dieser Heuristik gibt es noch eine Variante, die dann zum Einsatz kommt, wenn das neue Gleichheitsproblem bezüglich der Ordnung $<_{GO}$ nicht kleiner ist als das ursprüngliche. Dann wird zum einen nur überprüft, ob das Skelett des neuen Gleichheitsproblems größer ist als das des ursprünglichen. Zum anderen werden die Kontexte des alten und des neuen Gleichheitsproblems miteinander wie folgt verglichen: Für jedes Funktionssymbol, daß irgendwo als Kontext auftritt, wird die Anzahl der gerichteten Gleichungen in der Datenbasis gezählt, die diesen Kontext verschieben oder eliminieren. Gibt es für das neue Gleichheitsproblem insgesamt mehr solcher Gleichungen als für das ursprüngliche, wird mit dem neuen Gleichheitsproblem weiter gearbeitet. Die Motivation für diese Heuristik ist die, daß wenn der ursprüngliche Kontext nicht weiter bearbeitet werden kann, versucht werden muß eine Situation zu schaffen, mit der die kontextbearbeitenden Heuristiken besser zurechtkommen. Als Maß dafür wird die Anzahl der kontextbearbeitenden Gleichungen verwendet. Da die Einhaltung der Ordnung $<_{GO}$ nicht gefordert wird, birgt diese Variante die Gefahr eines nicht terminierenden Gesamtverfahrens. Deshalb muß diese Heuristik mit Vorsicht eingesetzt werden. Wie dies geschieht wird in Kapitel 5 über die Terminierung besprochen. Ein Beispiel für diese Heuristik ist in Kapitel 6 zu finden.

3.2.4.4 Skeletterweiternde Heuristiken

In diesem Abschnitt wird eine weitere spezielle Heuristik vorgestellt. Die Idee hinter dieser Heuristik ist, daß gezielt versucht wird, die Kontexte in den beiden Termen eines Gleichheitsproblems anzupassen. Dabei ist wieder die Repräsentation der Skelette und Kontexte durch Funktionssymbollisten hilfreich, da dadurch die Termpositionen bekannt sind, an denen ein neuer Kontext erzeugt werden muß, damit ein größeres Skelett entsteht.

Zur Motivation dieser Heuristik wird folgendes Beispiel betrachtet: Sei eine Signatur Σ gegeben, in der es zwei Funktionssymbole f und g gibt. Das Funktionssymbol f sei kommutativ, d. h. es gibt eine Gleichung

$$\forall \mathsf{x}, \mathsf{y}\ \mathsf{f}(\mathsf{x}, \mathsf{y}) \equiv \mathsf{f}(\mathsf{y}, \mathsf{x})$$

Weiterhin seien s, t_1 und t_2 Terme[4] zur Signatur Σ. Das zu lösende Gleicheits-

[4] Die Terme s und t können abstrakte Terme sein, wenn Σ eine abstrakte Signatur im Sinne

problem sei

$$\mathsf{g}(\mathsf{f}(\mathsf{g}(\mathsf{s}),\mathsf{t}_1)) \equiv \mathsf{g}(\mathsf{f}(\mathsf{t}_2,\mathsf{g}(\mathsf{s})))$$

Der Ausschnitt für den gemeinsamen Unterterm s der Abstraktionsinformation dieses Gleichheitsproblems ist

	$\mathsf{f}(\mathsf{s},\mathsf{t})$	$\mathsf{f}(\mathsf{t},\mathsf{s})$	Skelett
s	$[(\mathsf{g},1),(\mathsf{f},1),(\mathsf{g},1)]$	$[(\mathsf{g},1),(\mathsf{f},2),(\mathsf{g},1)]$	$[(\mathsf{g},1),(\mathsf{g},1)]$

Angenommen es gibt keine Gleichung, mit deren Hilfe die Kontexte $(\mathsf{f},1)$ und $(\mathsf{f},2)$ eliminiert werden könnten. Dann versagen alle bisher vorgestellten Heuristiken auf diesem Gleichheitsproblem. Die Idee, solche Problemfälle zu lösen, ist die, zu versuchen Kontexte, die in dem linken Term auftreten, gezielt an geeigneter Position im rechten Term zu erzeugen, um somit das Skelett zu vergrößern, oder umgekehrt. Die Position, an der ein solcher Kontext erzeugt werden muß, läßt sich relativ zum Skelett bestimmen. In dem Beispiel oben wird der Kontext $(\mathsf{f},1)$ in der Funktionssymbolliste zu dem Vorkommen von s im linken Term betrachtet. Dieser Kontext tritt zwischen den beiden Skelettteilen $(\mathsf{g},1)$ und $(\mathsf{g},1)$ auf, welche im rechten Term den Skelettteilen an den Termpositionen $[1]$ und $[1,2,1]$ entsprechen. Um den Kontext $(\mathsf{f},1)$ so zu erzeugen, daß das Skelett größer wird, muß dieser Kontext zwischen diesen beiden Termpositionen des rechten Terms erzeugt werden. Dazu sucht die Heuristik in der Datenbasis nach einer Gleichung, die einen solchen Kontext erzeugt. Auf diese Art wird die Gleichung $\forall \mathsf{x},\mathsf{y}\ \mathsf{f}(\mathsf{x},\mathsf{y}) \equiv \mathsf{f}(\mathsf{y},\mathsf{x})$ gefunden. Die Abstraktionsinformation dieser Gleichung enthält bezüglich des gemeinsamen Untertermes y folgenden Eintrag:

	$\mathsf{f}(\mathsf{x},\mathsf{y})$	$\mathsf{f}(\mathsf{y},\mathsf{x})$	Skelett
y	$[(\mathsf{f},2)]$	$[(\mathsf{f},1)]$	$[]$

Daraus ist ersichtlich, daß sie einen Kontext $(\mathsf{f},2)$ eliminiert, um einen Kontext $(\mathsf{f},1)$ zu erzeugen. Somit wird diese Gleichung von links nach rechts an der Termposition $[1]$ auf den rechten Term angewendet. Das neue Gleicheitsproblem ist

$$\mathsf{g}(\mathsf{f}(\mathsf{t}_1,\mathsf{g}(\mathsf{s}))) \equiv \mathsf{g}(\mathsf{f}(\mathsf{t}_2,\mathsf{g}(\mathsf{s})))$$

Das Skelett dieses Gleichheitsproblems ist größer als das Skelett des ursprünglichen Gleichheitsproblems. Anschließend kann mit der Angleichung der Funktionssymbollisten zu den gemeinsamen Untertermen von $\mathsf{g}(\mathsf{f}(\mathsf{t}_1,\mathsf{g}(\mathsf{s})))$ und $\mathsf{g}(\mathsf{f}(\mathsf{t}_2,\mathsf{g}(\mathsf{s})))$, die in t_1 und t_2 vorkommen, fortgefahren werden.

von Abschnitt 2.1.1 ist.

3.2.4.5 Anmerkungen

Erreichen die vorgestellten Heuristiken ihr Ziel, nämlich die vollständige Angleichung der Funktionssymbollisten, dann bedeutet dies nur, daß die beiden abstrakten Terme gleich sind. Da durch die Abstraktion Informationen weggefallen sind, sind die ursprünglichen Terme im allgemeinen noch nicht gleich. Dennoch sind sie sich ähnlicher geworden und die übriggebliebenen Probleme können als Unterprobleme behandelt werden. Dies setzt allerdings voraus, daß der durch die Angleichung der Abstraktionen entstandene Plan auch vollständig ausgeführt werden kann. Die in diesem Zusammenhang entstehenden Probleme werden ausführlich im Abschnitt 4.1 behandelt.

3.2.4.6 Verringerung des Abstraktionsniveaus

In diesem Abschnitt wird das Problem angesprochen, daß durch die Abstraktion eventuell zuviel wegabstrahiert wurde. Dies ist zum Beispiel der Fall, wenn die Informationen, die in gemeinsamen Untertermen stehen, nicht mehr verfügbar sind auf Grund der Abstraktion. Dies wird durch das folgende Beispiel illustriert: Seien die folgenden Axiome gegeben:

$$\forall \mathsf{n} : \mathsf{Nat}\ \forall \mathsf{x}, \mathsf{y} : \mathsf{List}\ \mathsf{app}(\mathsf{cons}(\mathsf{n}, \mathsf{x}), \mathsf{y}) \equiv \mathsf{cons}(\mathsf{n}, \mathsf{app}(\mathsf{x}, \mathsf{y}))$$
$$\forall \mathsf{n} : \mathsf{Nat}\ \forall \mathsf{x} : \mathsf{List}\ \mathsf{car}(\mathsf{cons}(\mathsf{n}, \mathsf{x})) \equiv \mathsf{n}$$

Das zu beweisende Theorem sei:

$$\forall \mathsf{n} : \mathsf{Nat}\ \forall \mathsf{x}, \mathsf{y} : \mathsf{List}\ \mathsf{car}(\mathsf{cons}(\mathsf{n}, \mathsf{x})) \equiv \mathsf{car}(\mathsf{app}(\mathsf{cons}(\mathsf{n}, \mathsf{x}), \mathsf{y}))$$

Wird der gemeinsame Unterterm $\mathsf{cons}(\mathsf{n}, \mathsf{x})$ dieses Theorems betrachtet, erhält man die folgende Abstraktion des Theorems:

$$\mathsf{car}(\mathsf{AC}_1) \equiv \mathsf{car}(\mathsf{app}(\mathsf{AC}_1, @))$$

Diese Abstraktionen der beiden Terme des Theorems können mit Hilfe der Axiome nicht angeglichen werden. Das Problem besteht darin, daß der anfangs gewählte gemeinsame Unterterm des Theorems im Verlauf des Beweises nicht erhalten bleibt. Für den Beweis muß der gemeinsame Unterterm aufgelöst werden. Da in dieser Abstraktion des Theorems die Struktur des gemeinsamen Unterterms nicht bekannt ist, scheitert die Angleichung der Abstraktionen. Deshalb wird ein Mechanismus benötigt, der es erlaubt, zu "starke" Abstraktionen zurückzunehmen. Im Falle des obigen Theorems bedeutet dies, daß anstatt den gemeinsamen Unterterm

$$\mathsf{cons}(\mathsf{n}, \mathsf{x})$$

zu betrachten, die gemeinsamen Unterterme n und x betrachtet werden müssen. Dies führt zu folgender Abstraktion des Theorems

$$\mathsf{car}(\mathsf{cons}(\mathsf{AC}_2, \mathsf{AC}_3)) \equiv \mathsf{car}(\mathsf{app}(\mathsf{cons}(\mathsf{AC}_2, \mathsf{AC}_3), @))$$

Diese beiden Abstraktionen können mit Hilfe der Abstraktionen der Axiome angeglichen werden.

Im folgenden wird eine formale Definition der Tiefe eines Termes benötigt, welche nichts anderes ist als die Länge der längsten Termposition dieses Termes.

Definition 3.10 (Größe/Tiefe eines Termes) *Sei ein Term* t *zu einer Signatur* Σ *gegeben. Dann ist die Größe des Termes durch die folgende Funktion* $\mathsf{Depth} : \mathsf{T}(\Sigma, \mathcal{V}) \to \mathbb{N}$ *gegeben:*

- *Falls* t *Variable oder Konstante:*

$$\mathsf{Depth}(\mathsf{t}) := 0$$

- *Falls* $\mathsf{t} = \mathsf{f}(\mathsf{t}_1, \ldots, \mathsf{t}_n)$ *:*

$$\mathsf{Depth}(\mathsf{t}) := 1 + \max\{\mathsf{Depth}(\mathsf{t}_1), \ldots, \mathsf{Depth}(\mathsf{t}_n)\}$$

Im folgenden wird der allgemeine Mechanismus beschrieben, wie die Verringerung des Abstraktionsniveaus realisiert ist. Gegeben sei eine abstraktes Gleichheitsproblem $\mathsf{s} \equiv \mathsf{t}$. Die betrachteten Vorkommen von gemeinsamen Untertermen seien $\Pi_\mathsf{s} \subseteq \Pi(\mathsf{s})$ und $\Pi_\mathsf{t} \subseteq \Pi(\mathsf{t})$. Zunächst muß die Menge der durch Π_s und Π_t denotierten gemeinsamen Unterterme berechnet werden. Dazu wird eine Funktion $\mathsf{Subterms}$ eingeführt:

Definition 3.11 *Gegeben seien zwei Terme* s *und* t *aus* $\mathsf{T}(\Sigma, \mathcal{V})$ *sowie zwei Mengen* $\Pi_\mathsf{s} \subseteq \Pi(\mathsf{s})$ *und* $\Pi_\mathsf{t} \subseteq \Pi(\mathsf{t})$ *von Vorkommen von gemeinsamen Untertermen von* s *und* t*. Dann wird die Funktion* $\mathsf{Subterms}$ *wie folgt definiert:*

$$\mathsf{Subterms}(\mathsf{s}, \mathsf{t}, (\Pi_\mathsf{s}, \Pi_\mathsf{t})) := \{\mathsf{u}\mathsf{T}(\Sigma, \mathcal{V}) \mid \exists \pi \in \Pi_\mathsf{s}\ \mathsf{s}_{|\pi} = \mathsf{u}\}$$

Das Prinzip ist das folgende: Aus der Menge $\mathsf{M} := \mathsf{Subterms}(\mathsf{s}, \mathsf{t}, \mathsf{M})$ der gemeinsamen Unterterme wird ein größter Unterterm ausgewählt. Die Größe eines Termes wird dabei durch die Termtiefe gegeben (siehe oben). Ist dieser größte Unterterm eine Variable oder Konstante, ist das niedrigste Abstraktionsniveau erreicht und kann daher nicht verringert werden. Ansonsten, falls dieser Unterterm der Form $\mathsf{f}(\mathsf{t}_1, \ldots, \mathsf{t}_n)$ ist, wird er aus der Menge M entfernt und die Terme $\mathsf{t}_1, \ldots \mathsf{t}_n$ hinzugefügt.

In Abschnitt 3.2.1.1 über die Unabhängigkeit von Untertermen wurden für die gemeinsamen Unterterme neue Konstanten eingeführt. Diese wurden an Stelle der alten Unterterme betrachtet, da sie die Unabhängigkeitseigenschaft erfüllen. Diese neuen Konstanten haben aber alle die Termtiefe 0. Dies bedeutet, daß das Abstraktionsniveau nicht mehr verringert werden kann. Da aber die vorgenommenen Ersetzungen bekannt sind, können sie auch zurückgenommen werden; d. h. daß die ursprünglich eingeführten Konstanten $\mathsf{AC_i}$ wieder durch die respektiven Terme $\mathsf{t_i}$ ersetzt werden; hierbei bleiben die Mengen Π_s und Π_t unverändert. Anschließend wird die Menge der $\mathsf{t_i}$ betrachtet und die Verringerung des Abstraktionsniveaus vorgenommen. Anschließend werden diese wieder durch Konstanten $\mathsf{AC_i}'$ ersetzt. Dies läßt sich wie folgt skizzieren: Gegeben sei ein abstraktes Gleichheitsproblem $\mathsf{s} \equiv \mathsf{t}$, in dem die betrachteten gemeinsamen Unterterme $\mathsf{t_i}$ durch Konstanten $\mathsf{AC_i}$ ersetzt wurden.

$$\begin{array}{ccc}
\mathsf{s[AC_i]} \equiv \mathsf{t[AC_i]} & & \mathsf{s[AC_i']} \equiv \mathsf{t[AC_i']} \\
| & & \uparrow \\
\text{Ersetze alle} & & \text{Ersetze alle} \\
\mathsf{AC_i} \text{ durch } \mathsf{t_i} & & \mathsf{t_i'} \text{ durch } \mathsf{AC_i'} \\
\downarrow & & | \\
\mathsf{s[t_i]} \equiv \mathsf{t[t_i]} & \longrightarrow & \mathsf{s[t_i']} \equiv \mathsf{t[t_i']}
\end{array}$$

Die neuen Mengen Π_s' und Π_t' von Vorkommen von $\mathsf{s[AC_i']}$ und $\mathsf{t[AC_i']}$ lassen sich direkt aus den alten Mengen Π_s und Π_t bestimmen.

3.3 Heuristiken auf Termstrukturen

Die Abstraktionsebene der Termstrukturen wurde in Abschnitt 2.2 eingeführt. Diese wurden als abstrakte Terme definiert, nur daß das Abstraktionsverfahren ein anderes ist. Den Abstraktionsverfahren der Termstrukturen und der abstrakten Terme ist gemeinsam, daß sie beide den Fokus auf gemeinsame Unterterme einer Gleichung richten. Somit können alle Heuristiken, die bisher nur auf den abstrakten Termen definiert wurden, auch auf den Strukturtermen verwendet werden. Ein Unterschied zwischen den Strukturtermen und den abstrakten Termen ist, daß die berechneten Pläne für den Beweis eines konkreten Gleichheitsproblems in unterschiedlicher Art und Weise verfeinert werden müssen. Darauf wird in Kapitel 4 näher eingegangen. Ein weiterer Unterschied liegt in der Klasse der beweisbaren konkreten Gleichheitsprobleme, für die es auf der Abstraktionsebene der Strukturterme und der der abstrakten Terme Beweise gibt. In der Tat können manche Beweise auf der Abstraktionsebene der abstrakten Terme nicht geführt werden, wohingegen auf der Abstraktionsebene der Strukturterme schon, weshalb auch diese zweite Abstraktionsebene eingeführt wurde.

Zusätzlich zu den im vorherigen Abschnitt eingeführten Heuristiken, die auch auf Termstrukturen verwendbar sind, wird hier ein weiteres Verfahren angegeben, um die Strukturterme eines gegebenen Gleichheitsproblems anzugleichen. Dieses

Verfahren besteht aus der Definition eines Termersetzungssystems, in dem die durch Abstraktion der Axiome erhaltenen Strukturgleichungen mit Hilfe einer noch zu definierenden Ordnung gerichtet werden. Um einen Plan für einen Beweis eines konkreten Gleichheitsproblems zu erhalten, werden die Strukturterme der beiden Terme des Gleichheitsproblems mit Hilfe des Termersetzungssystems vollständig reduziert. Sind diese reduzierten Strukturterme gleich, kann die Liste der angewendeten gerichteten Strukturgleichungen als Plan für den Beweis des konkreten Gleichheitsproblems verwendet werden.

Im folgenden wird zunächst eine Ordnung auf Strukturtermen angegeben, mit Hilfe derer die Strukturgleichungen gerichtet werden. Daß diese Ordnung eine sogenannte "Terminierungsordnung" ist, wird in Kapitel 5 über die Terminierung bewiesen. Anschließend wird noch die Konfluenz eines solchen "Strukturterm"-Ersetzungssystems diskutiert.

Für die Definition der Ordnung $<_{\mathsf{TO}}$ werden die Begriffe der Multimengen und der Multimengenordnungen benötigt. Diese werden zunächst kurz eingeführt.

3.3.1 "Strukturterm"-Ersetzungssystem

3.3.1.1 Multimengen

Definition 3.12 (Multimengen) *Sei eine Menge* M *gegeben. Dann ist eine* Multimenge X *über* M *eine totale Funktion* $\mathsf{X} : \mathsf{M} \to \mathbb{N}$ *, die jedem Element aus* M *eine natürliche Zahl größer gleich* 0 *zuordnet.*

Eine **Multimenge** X , die wie folgt definiert $(\mathsf{a}, \mathsf{b}, \mathsf{c} \in \mathsf{M})$ ist

$$\mathsf{X}(\mathsf{a}) = 3, \mathsf{X}(\mathsf{b}) = 2, \mathsf{X}(\mathsf{c}) = 0$$

wird als $\{\mathsf{a}, \mathsf{a}, \mathsf{a}, \mathsf{b}, \mathsf{b}\}$ geschrieben. Eine alternative Darstellung der Multimenge X ist $\{\mathsf{a}, \mathsf{b}, \mathsf{a}, \mathsf{b}, \mathsf{a}\}$.

Folgende Notationen werden auf Multimengen definiert:

Definition 3.13 (Differenz und Vereinigung zweier Multimengen) *Sei eine Menge* M *und zwei Multimengen* X *und* Y *über* M *gegeben. Dann ist die* Differenz $\mathsf{X} \setminus_{\mathsf{M}} \mathsf{Y}$ *wie folgt definiert:*

$$\forall \mathsf{m} \in \mathsf{M} \quad (\mathsf{X} \setminus_{\mathsf{M}} \mathsf{Y})(\mathsf{m}) := \begin{cases} 0, & \text{falls } \mathsf{X}(\mathsf{m}) < \mathsf{Y}(\mathsf{M}) \\ \mathsf{X}(\mathsf{m}) - \mathsf{Y}(\mathsf{m}) & \text{sonst} \end{cases}$$

Die Vereinigung *der beiden Multimengen wird wie folgt definiert:*

$$\forall \mathsf{m} \in \mathsf{M} \quad (\mathsf{X} \cup_{\mathsf{M}} \mathsf{Y})(\mathsf{m}) := \mathsf{X}(\mathsf{m}) + \mathsf{Y}(\mathsf{m})$$

Notation: Anstelle von $X(m) > 0$ wird auch $m \in X$ geschrieben.

Satz 3.1 (Kommutativität und Assoziativität von $\cup_M$) *Die* Vereinigung $\cup_M$ *von Multimengen ist* assoziativ *und* kommutativ.

Beweis 3.1 Beide Eigenschaften folgen direkt aus der Assoziativität und der Kommutativität der Addition von natürlichen Zahlen. ∎

3.3.1.2 Multimengenordnung

Definition 3.14 (Multimengenordnung) *Gegeben sei eine partielle Ordnung $>$ über einer Menge M. Dann ist die* Multimengenordnung $\gg$ *auf der Menge der endlichen Multimengen über M wie folgt definiert: Für zwei Multimengen X und Y über M ist $X \gg Y$, falls $X \neq Y$ und für alle $y \in Y \setminus_M X$ existiert ein $x \in X \setminus_M Y$ mit $x > y$.*

Satz 3.2 *Sei $>$ eine noethersche partielle Ordnung über einer Menge M. Dann ist die entsprechende Multimengenordnung $\gg$ auf der Menge der endlichen Multimengen über M auch noethersch.*

Beweis 3.2 Angenommen es gäbe eine unendliche Folge von Multimengen $X_i, i = 0 \ldots \infty$, für die gilt:

$$X_0 \gg X_1 \gg \ldots$$

Aus der Definition der Multimengenordnung folgt dann, daß es eine unendliche Menge von Elementen $x_i \in M, i = 0 \ldots M$ geben muß (mit $x_i \in X_i$, für alle $i = 0 \ldots \infty$), für die gilt:

$$x_0 > x_1 > \ldots$$

Dies widerspricht aber der Tatsache, daß $>$ noethersch ist. ∎

3.3.2 Terminierungsordnung

Sei $<_F$ eine Ordnung auf den Funktionssymbolen und $\ll_F$ die entsprechende Multimengen Erweiterung (siehe Abschnitt 3.3.1.2) von $<_F$. Dann ist die Terminierungsordnung $<_{TO}$ wie folgt definiert: Sei eine Strukturgleichung $s \equiv t$ gegeben. Betrachtet wird die Richtung "$s \rightarrow t$". Verglichen werden die Multimengen der Funktionssymbole von s und t. Ist die von s bezüglich der Multimengenordnung $\ll_F$ größer als die von t , gilt $s >_{TO} t$.

Zur Definition der Ordnung $<_{TO}$ wird eine Funktion benötigt, die die Anzahl der Vorkommen eines Termes u in einem Term t berechnet. Diese sei im folgendem mit $\#_u(t)$ bezeichnet. Beispiel: $\#_x(plus(s(x), x)) = 2$.

Formal wird die Terminierungsordnung $<_{TO}$ folgendermaßen definiert:

Definition 3.15 (Terminierungsordnung) *Gegeben sei eine Strukturgleichung* $\mathsf{s} \equiv \mathsf{t}$ *zu einer Signatur* Σ *und einer Variablenmenge* $\mathcal{V}$ *und seien* $\mathsf{Fu(s)}$ *und* $\mathsf{Fu(t)}$ *die respektiven Multimengen von Funktionssymbolen von* s *und* t*. Dann definiert sich die* Terminierungsordnung $<_{\mathsf{TO}}$ *wie folgt:*

$$\mathsf{s} <_{\mathsf{TO}} \mathsf{t}$$

genau dann, wenn $\mathsf{Fu(s)} \ll_{\mathsf{F}} \mathsf{Fu(t)}$ *und für alle Variablen* $\mathsf{x} \in \mathcal{V}$ *gilt:* $\#_{\mathsf{x}}(\mathsf{s}) \leq \#_{\mathsf{x}}(\mathsf{t})$.

3.3.3 Das Ersetzungssystem

Definition 3.16 (Ersetzungssystem) *Gegeben seien eine Menge* Φ *von Strukturgleichungen aus* $\mathsf{AAt}(\Sigma, \mathcal{V})$ *. Dann ist das zu* Φ *gehörige* Ersetzungssystem, *definiert als eine Menge von gerichteten Ersetzungsregeln*

$$\mathsf{s} \rightarrow \mathsf{t}$$

für die gilt:

1. $\mathsf{s} \equiv \mathsf{t} \in \Phi$ *und*
2. *es gilt:* $\mathsf{s} <_{\mathsf{TO}} \mathsf{t}$

Die so entstehenden gerichteten Ersetzungssysteme werden im folgendem mit E_Φ bezeichnet, oder einfach E, wenn Φ aus dem Kontext klar oder nicht relevant ist.

Ein Ableitungsschritt in diesem Ersetzungssystem entspricht der Anwendung einer gerichteten Gleichung auf einen Term (siehe Definition 2.8).

Definition 3.17 (Ableitung im Ersetzungssystem) *Gegeben seien ein Ersetzungssystem* E *und zwei Strukturterm* s *und* s'*. Dann ist*

$$\mathsf{s} \vdash_{\mathsf{E}} \mathsf{s}'$$

ein Ableitungsschritt *in* E *, falls es eine gerichtete Gleichung in* E *gibt, die auf den Strukturterm* s *anwendbar ist und das Ergebnis dieser Anwendung der Strukturterm* s' *ist.*

Notation: $\vdash_{\mathsf{E}}^{*}$ *ist die reflexive, transitive Hülle von* $\vdash_{\mathsf{E}}$ *.*

Durch das Symbol $\leftrightarrow_{\mathsf{E}}{}^{*}$ wird der Äquivalenzabschluß der Ableitungsrelation $\vdash_{\mathsf{E}}$ beschrieben. Formal bedeutet dies:

Definition 3.18 (Äquivalenzabschluß) *Gegeben seien zwei Strukturterme* s *und* t*, sowie ein "Strukturterm"-Ersetzungssystem* E*. Dann gilt* $\mathsf{s} \leftrightarrow_{\mathsf{E}} \mathsf{t}$ *genau dann, wenn es einen Strukturterm* s' *gibt, mit:*

- *Entweder gilt* $\mathsf{s} = \mathsf{t}$
- *oder*
  1. *Entweder* $\mathsf{s} \vdash_{\mathsf{E}} \mathsf{s}'$ *oder* $\mathsf{s}' \vdash_{\mathsf{E}} \mathsf{s}$
  2. *und* $\mathsf{s}' \leftrightarrow_{\mathsf{E}} \mathsf{t}$

3.3.4 Eigenschaften

Der Begriff der **Konfluenz** ist ein zenraler Begriff in der Theorie der Termersetzungssysteme. Der Grund, warum diese Eigenschaft wünschenswert ist, ist ein pragmatischer Grund: Im allgemeinen sind auf einen Term mehrere gerichtete Gleichungen eines Termersetzungssystems E anwendbar. In einer solchen Situation, müßte eine Entscheidung getroffen werden, welche Ersetzungsregel angewendet werden soll. Da außerdem nicht sichergestellt ist, daß es eine eindeutige Normalform für jeden Term gibt, könnte aus der Ungleichheit der Normalformen zweier Terme s und t nicht gefolgert werden, daß die Gleichung $\mathsf{s} \equiv \mathsf{t}$ in der betrachteten Theorie nicht gültig ist. Ist das Termersetzungssystem E aber konfluent, gibt es für jeden Term eine eindeutige Normalform, und der Rückschluß wäre korrekt. Da das oben definierte "Strukturterm"-Ersetzungssystem noethersch ist, da zur Orientierung der Gleichungen eine Simplifikationsordnung verwendet wird (siehe Kapitel 5), gilt, daß wenn es in einem solchen "Strukturterm"-Ersetzungssystem E nur kritischen Paare (p, q) gibt, für die es einen Term r gibt, so daß $\mathsf{p} \rightarrow_{\mathsf{E}}^{*} \mathsf{r}$ und $\mathsf{q} \rightarrow_{\mathsf{E}}^{*} \mathsf{r}$ gilt, dieses konfluent ist. Diese Eigenschaften sind von der Theorie der Termersetzungssysteme her bekannt und zum Beispiel in [SA91] nachzulesen. Daraus ergeben sich zwei Möglichkeiten, auf den Strukturtermen mit einem konfluenten "Strukturterm"-Ersetzungssystem zu arbeiten:

Die erste Möglichkeit besteht darin, nicht alle orientierbaren Strukturgleichungen in ein zu bauendes "Strukturterm"-Ersetzungssystem aufzunehmen. Eine orientierbare Strukturgleichung wird nur dann in ein bestehendes "Strukturterm"-Ersetzungssystem E aufgenommen, wenn dadurch keine kritischen Paare entstehen, die nicht konvergieren. Da daß so entstehende Termersetzungssytem noethersch ist und keine divergierenden kritischen Paare enthält, ist es trivialerweise konfluent. Allerdings wird die Klasse der konkreten Gleichheitsprobleme, für deren Strukturgleichheitsprobleme ein Beweis gefunden werden kann, weiter eingeschränkt. Da aber sowieso schon nicht für alle konkreten Gleichheitsprobleme Strukturbeweise existieren, und die Ebene der Strukturterme nur eine Heuristik ist, um einen Plan für einen Beweis des konkreten Gleichheitsproblems zu finden, kann diese Einschränkung ruhig gemacht werden. Dies bedeutet, daß, wie auch bei den anderen Heuristiken aus Abschnitt 3.2.4, wenn für ein Strukturgleichheitsproblem eines konkreten Gleichheitsproblems kein Beweis gefunden werden kann, dies nicht bedeutet, daß es auch keinen Beweis gibt. Wenn aber einer gefunden wird, dann kann dieser als Plan für einen Beweis des konkreten Gleichheitsproblems verwendet werden.

Eine zweite Möglichkeit besteht darin, alle orientierbaren Gleichungen in dem "Strukturterm"-Ersetzungssystem E aufzunehmen. Anschließend kann mit Hilfe der von Knuth und Bendix (siehe [KB70]) eingeführten Methode versucht werden, dieses Termersetzungssytem zu einem konfluentem Termersetzungssystem zu vervollständigen. Gelingt die Vervollständigung, kann das neue Termersetzungssystem für die abstrakte Planung verwendet werden. Gelingt dies nicht, kann überhaupt keine Planung erfolgen. Dies ist der größte Nachteil dieser Möglichkeit gegenüber der ersten.

Kapitel 4

Verfeinerung des Plans

4.1 Abstraktionsebene der “abstrakten Terme”

Sinn und Zweck der vorgestellten Abstraktion von Termen und Gleichungen war es, einen Plan für ein zu lösendes Gleichheitsproblem zu finden. Dabei ist die Abstraktion so gewählt worden, daß während der Beweisfindung auf bestimmte interessante gemeinsame Untertermen fokussiert wird. Dieses Vorgehen entspricht einer intuitiven, am Menschen beobachtbaren Vorgehensweise auf bestimmten Gleichheitsproblemen. Dabei wird ein Plan erstellt, der eine Abschätzung darstellt, wie das Ziel, die Lösung des Gleichheitsproblems, erreicht werden könnte. Anschließend wird versucht, den Plan auszuführen. Dabei kann es zu einigen Problemen kommen. Die gleiche Art von Problemen taucht auch in der vorliegenden Arbeit auf, da die Abstraktion und die Heuristiken teilweise diesem Problemlösungsverhalten nachempfunden sind.

Während der Angleichung der abstrakten Terme werden die Informationen gespeichert, wann welche abstrakte, gerichtete Gleichung auf welchen der beiden abstrakten Terme an welcher Position angewendet wurden. An jeder Stelle des Beweises ist bekannt, welchem konkreten Term der aktuell bearbeitete abstrakte Term entspricht. Diese Informationen werden automatisch durch die zentrale Funktion (siehe Abschnitt 3.2.4.1) zur Anwendung einer abstrakten, gerichteten Gleichung auf einen abstrakten Term aktualisiert. Die so entstehenden Ableitungsketten für die Abstraktionen zweier anzugleichender Terme `s` und `t` stellen einen Plan für die Angleichung der beiden Terme `s` und `t` dar.

Nun muß noch sichergestellt werden, daß die Ableitungsschritte auf der abstrakten Ebene einen Plan für die Lösung eines konkreten Gleichheitsproblems darstellen. Ein Ableitungsschritt auf der abstrakten Ebene ist die Anwendung einer Klasse von Gleichungen auf eine Klasse von Termen wodurch eine neue Klasse von Termen erhalten wird. Die Notation für Äquivalenzklassen von Termen wurde in

Abschnitt 2.1.3.1 eingeführt, d. h. wenn ein abstrakter Ableitungsschritt

$$[\mathsf{s}] \xrightarrow{\pi,[\mathsf{u}]\to[\mathsf{v}]} [\mathsf{t}]$$

gegeben ist, muß folgendes gelten, damit er ein Plan für einen Ableitungsschritt auf der konkreten Ebene ist: Für jeden Term w, dessen Abstraktion $[\mathsf{s}]$ ist und jede Gleichung $\mathsf{q} \to \mathsf{r}$, deren Abstraktion $[\mathsf{u}] \to [\mathsf{v}]$ ist, muß so etwas gelten wie: Wenn mit Hilfe einiger Modifikationen auf w oder q die Gleichung $\mathsf{q} \to \mathsf{r}$ auf den Term w angewendet werden kann, muß die Abstraktion des Ergebnisses dieser Anwendung der abstrakte Term $[\mathsf{t}]$ sein. Würde dies nicht gelten, wären alle weiteren abstrakten Ableitungsschritte hinfällig und die restliche abstrakte Ableitung kein Plan mehr. Dadurch werden die Modifikationen, die auf w und q vorgenommen werden dürfen, eingeschränkt. Im folgendem wird definiert, welche Eigenschaften die Modifikationen erfüllen müssen, damit eine abstrakte Ableitung einen Plan für eine Ableitung auf konkreten Termen darstellen kann. Im folgenden werden zwei Theoreme angegeben, die aussagen, daß wenn nur solche Modifikationen verwendet werden, die diese Eigenschaften erfüllen, die abstrakte Ableitung ein Plan für die konkrete Ableitung ist.

4.1.1 Eigenschaften der Modifikationen

Der Ausgangspunkt ist folgender: Gegeben sei ein Term s der, bezüglich einer Menge von Vorkommen Π_s, in der Äquivalenzklasse $[\mathsf{s}]_{\Pi_\mathsf{s}}$ ist. Das Problem ist, welche Gleichungen $\mathsf{u} \to \mathsf{v}$ dürfen an welchen Termpositionen π auf s angewendet werden, so daß das Ergebnis $\sigma(\mathsf{s})_{|\pi\leftarrow\sigma(\mathsf{v})}$ der Anwendung wieder aus der Äquivalenzklasse $[\mathsf{s}]_{\Pi_\mathsf{s}}$ oder zumindest eine Instanz dieser Äquivalenzklasse ist. Dabei wird die Anwendung zunächst darauf eingeschränkt, daß der Term u den entsprechenden Unterterm $\mathsf{s}_{|\pi}$ matched. Der allgemeine Fall der Unifikation wird später betrachtet.

Es muß ausgeschlossen werden, daß eine Gleichung auf eines der durch Π_s denotierten Vorkommen eines Untertermes von s angewendet werden kann, da ansonsten dieser Unterterm zerstört werden könnte, und somit der ganze Plan hinfällig würde. Deshalb wird für einen Term s und eine Menge $\Pi_\mathsf{s} \subseteq \Pi(\mathsf{s})$ von Vorkommen die Menge der **zulässigen Termpositionen** definiert.

Definition 4.1 (Zulässige Termposition) *Sei ein Term $\mathsf{s} \in \mathsf{T}(\Sigma, \mathcal{V})$ und eine Menge von Vorkommen $\Pi_\mathsf{s} \subseteq \Pi(\mathsf{s})$ gegeben. Dann ist eine Termposition π von s eine* zulässige *Termposition von s bezüglich Π_s , genau dann wenn es kein $\pi_\mathsf{s} \in \Pi_\mathsf{s}$ gibt, so daß π_s Präfix von π.*

Beispiel:
Bezüglich des Termes $\mathsf{plus}(\mathsf{s}(\mathsf{x}), \mathsf{y})$ und des Vorkommens von $\mathsf{s}(\mathsf{x})$ ist die Termposition $[2]$ zulässig, die Termposition $[1, 1]$ hingegen nicht.

Satz 4.1 *Sei ein Term* $\mathsf{s} \in \mathsf{T}(\Sigma, \mathcal{V})$, *eine Menge von Vorkommen* $\Pi_\mathsf{s} \subseteq \Pi(\mathsf{s})$ *und eine gerichtete Gleichung* $\mathsf{u} \to \mathsf{v}$ *gegeben. Weiterhin sei* π *eine zulässige Termposition bezüglich* s *und* Π_s *und* σ *eine Substitution, so daß gilt:* $\mathsf{s}_{|\pi} = \sigma(\mathsf{u})$. *Sei eine Menge von Termpositionen* $\Pi_{\sigma(\mathsf{u})} \subseteq \Pi(\sigma(\mathsf{u}))$ *wie folgt definiert:*

$$\Pi_{\sigma(\mathsf{u})} := \{\pi_{\sigma(\mathsf{u})} \in \Pi(\sigma(\mathsf{u})) \mid \exists \pi' \; \pi'\pi \in \Pi(\mathsf{s})\}$$

Falls $\Pi_{\sigma(\mathsf{u})} \subseteq \Pi(\sigma(\mathsf{v}))$ *und* $[\sigma(\mathsf{u})]_{\Pi_{\sigma(\mathsf{u})}} = [\sigma(\mathsf{v})]_{\Pi_{\sigma(\mathsf{u})}}$ *gilt, gilt auch:*

$$[\mathsf{s}_{|\pi, \leftarrow \sigma(\mathsf{v})}]_{\Pi_\mathsf{s}} = [\mathsf{s}]_{\Pi_\mathsf{s}}$$

Beweis 4.1 Der Beweis wird induktiv über die Länge der Termposition π geführt. Die betrachteten Termpositionen müssen alle zulässig sein. An der entsprechenden Stelle des Induktionsschrittes wird dieser Forderung Rechnung getragen werden.

$\pi = []$: Es gilt: $\mathsf{s} = \sigma(\mathsf{u})$. Da per Annahme $[\sigma(\mathsf{u})]_{\Pi_{\sigma(\mathsf{u})}} = [\sigma(\mathsf{v})]_{\mathsf{Pi}_{\sigma(\mathsf{u})}}$ gilt, gilt auch $[\mathsf{s}]_{\Pi_\mathsf{s}} = [\sigma(\mathsf{v})]_{\Pi_\mathsf{s}}$.

$\pi \to [\mathsf{i}, \pi]$: D. h. s ist der Form $\mathsf{f}(\mathsf{s}_1, \ldots, \mathsf{s}_\mathsf{n})$. Da $[\mathsf{i}, \pi]$ eine zulässige Termposition bezüglich s und Π_s ist, ist einerseits π eine zulässige Termposition bezüglich s_i und $\Pi_{\mathsf{s}_\mathsf{i}}$ ($\Pi_{\mathsf{s}_\mathsf{i}}$ definiert wie $\Pi_{\sigma(\mathsf{u})}$) und andererseits i eine zulässige Termposition bezüglich s und Π_s; d. h. per Induktionsannahme gilt:

$$[\mathsf{s}_{\mathsf{i}_{|\pi, \leftarrow \sigma(\mathsf{v})}}]_{\Pi_{\mathsf{s}_\mathsf{i}}} = [\mathsf{s}_\mathsf{i}]_{\Pi_{\mathsf{s}_\mathsf{i}}}.$$

Wiederrum per Induktionsannahme gilt dann auch:

$$[\mathsf{s}_{|\mathsf{i}, \leftarrow \mathsf{s}_{\mathsf{i}_{|\pi, \leftarrow \sigma(\mathsf{v})}}}]_{\Pi_\mathsf{s}} = [\mathsf{s}]_{\Pi_\mathsf{s}}$$

■

4.1.2 Anmerkung

In diesem Abschnitt sollen die Konsequenzen der Ergebnisse aus dem vorherigen Abschnitt auf die Planausführung diskutiert werden. Unumstritten ist, daß versucht werden muß, eine, durch den Plan vorgeschlagene Gleichung $\mathsf{u} \to \mathsf{v}$ auf einen Term s anzuwenden. Problematischer sind die Modifikationen, denen man den Term s unterziehen muß, um die Gleichung $\mathsf{u} \to \mathsf{v}$ anwenden zu können. Hierfür gibt es zwei Szenarien: Zum einen können die notwendigen Modifikation “blind” durchgeführt werden, so daß die Gleichung anwendbar wird. Nachdem die Modifikationen getätigt wurden, muß überprüft werden, ob der erhaltene Term s' in derselben Äquivalenzklasse wie der ursprüngliche Term s liegt. Gilt dies, kann die Gleichung $\mathsf{u} \to \mathsf{v}$ angewendet werden und die Planausführung fortgesetzt werden. Gilt dies nicht, müssen alle Modifikationen verworfen werden, da keine Kontrolle darüber besteht, an welcher Stelle die Modifikationen die Äquivalenzklassen-Zugehörigkeit verletzt haben. Dies könnte dadurch vermieden werden, daß Buch über die Modifikationen geführt wird, so daß anschließend analysiert werden kann, an welcher Stelle die Äquivalenzklassen-Zugehörigkeit verletzt wurde. Ist die Stelle gefunden, könnte an dieser Stelle neu angesetzt werden, um eine andere, die Äquivalenzklassen-Zugehörigkeit nicht verletzende Modifikation zu finden. Es kann aber sein, daß die notwendigen Modifikationen zwar in ihrem ganzen die Äquivalenzklasse erhalten, sie aber ohne temporäre Verletzungen nicht durchführbar sind. Dann schlägt dieser Ansatz fehl. Das gleiche Problem taucht im zweiten Szenario auf.

Das zweite Szenario ist die Weiterentwicklung des letzten Ansatzes. Anstatt im nachhinein zu versuchen, die Modifikationen, die die Äquivalenzklasse verlassen zurückzunehmen, wäre es sinnvoller, gleich nur solche Modifikationen zuzulassen, die dies nicht tun. Dies bedeutet zum einen, daß Modifikationen nur an **zulässigen Termpositionen** durchgeführt werden dürfen. Zum anderen müssen die instantierten Gleichungen die in Theorem 4.1 beschriebenen Eigenschaften erfüllen. Der oben beschriebene Nachteil, daß mögliche Modifikationen gar nicht erst gefunden werden, gilt auch hier. Dieser Ansatz hat aber den Vorteil, daß, wenn alle Modifikationen durchgeführt sind und die Gleichung $\mathsf{u} \to \mathsf{v}$ anwendbar ist, sicher ist, daß der Plan weiter gültig ist und dessen Ausführung fortgesetzt werden kann.

4.2 Abstraktionsebene der "Strukturterme"

Auf der Abstraktionsebene der Strukturterme wurden Strukturgleichungen auf Strukturterme angewendet. Dabei beschreiben die Strukturterme eine Äquivalenzklasse von konkreten Termen, die die gleiche Termstruktur aufweisen bezüglich bestimmter Vorkommen von Untertermen. Die Strukturgleichungen sind in diesem Sinne Übergänge zwischen Äquivalenzklassen. Diese Strukturgleichungen sind, analog zu den Strukturtermen, Äquivalenzklassen von Gleichungen, die bezüglich gemeinsamer Unterterme dieselbe Strukturveränderung beschreiben. Für die Vefeinerung eines Planschrittes

$$[\mathsf{s}] \vdash_{\pi,[\mathsf{u}]\to[\mathsf{v}]} [\mathsf{s}']$$

muß für einen Term $\mathsf{t} \in [\mathsf{s}]$ und eine Gleichung $\mathsf{q} \equiv \mathsf{r} \in [\mathsf{u} \equiv \mathsf{v}]$ folgendes gemacht werden: Durch den Vergleich von $[\mathsf{s}]_{|\pi}$ und$[\mathsf{u}]$ läßt sich eine Termsubstitution ρ generieren, so daß $[\mathsf{s}]_{|\pi} = \rho([\mathsf{u}])$ gilt. Durch die Vergleiche von $\mathsf{t}_{|\pi}$ mit $[\mathsf{s}]_{|\pi}$ und q mit $[\mathsf{u}]$ läßt sich daraus eine Termsubstitution σ für Unterterme von q durch Unterterme von t generieren. Diese Termsubstitution ist der Form

$$\sigma = [\mathsf{u}_1 \setminus \mathsf{q}_1 \ldots \mathsf{u}_n \setminus \mathsf{q}_n]$$

Dann muß folgender Unterbeweis geführt werden:

$$\forall \mathsf{x}_1 \ldots \mathsf{x}_l \in \mathsf{Var}(\mathsf{q})\ \exists \mathsf{y}_1 \ldots \mathsf{y}_m \in \mathsf{Var}(\mathsf{u})\ \left[\mathsf{u}_1 \equiv \mathsf{q}_1 \wedge \ldots \mathsf{u}_n \equiv \mathsf{q}_n \wedge \mathsf{q}_{|\pi} \equiv \mathsf{u}\right]$$

Gelingt dieser Beweis, definiert sich der aus der Anwendung von $\mathsf{q} \equiv \mathsf{r}$ auf t resultierende Term t' wie folgt:

$$\mathsf{t}' = \mathsf{t}_{|\pi, \leftarrow \sigma(\mathsf{r})}$$

Für diesen Term t' gilt dann, daß es eine Menge Π von Vorkommen aus $\Pi(\mathsf{t}')$ gibt, so daß gilt: $\beta(\mathsf{t}', \Pi) = [\mathsf{s}']$.

Kapitel 5

Terminierung

Die in Kapitel 3 eingeführten Heuristiken auf abstrakten Termen und Strukturtermen werden zusammengebaut zu einem Gesamtverfahren. Von diesem Gesamtverfahren muß die Terminierung gezeigt werden, was im ersten Abschnitt dieses Kapitels untersucht wird. Auf den Strukturtermen wurde noch ein weiteres Verfahren angegeben, nämlich die Konstruktion eines "Strukturterm"-Ersetzungssystems aus den Strukturgleichungen. Dafür wurde eine Ordnung $<_{\mathsf{TO}}$ definiert. Von dieser Ordnung muß gezeigt werden, daß sie noethersch, substitutionsstabil und subtermstabil ist. Daraus folgt dann die Terminierung eines mit dieser Ordnung erzeugten Ersetzungssystems. Diese Eigenschaften werden im zweiten Abschnitt bewiesen.

5.1 Heuristiken

Das Gesamtverfahren zur Angleichung zweier abstrakter Terme unterteilt sich in drei Arten von Heuristiken: Erstens gibt es die Heuristiken, die sich an die im folgenden definierte noethersche Ordnung $<_{\mathsf{GO}}$ halten. Eine Gleichungsanwendung einer solchen Heuristik liefert immer ein bezüglich der Ordnung $<_{\mathsf{GO}}$ kleineres Gleichheitsproblem zurück. Dadurch wird die Terminierung gesichert. Diese Heuristiken versuchen die Kontexte in der Abstraktionsinformation eines Gleichheitsproblems nach außen zu schieben, oder das Skelett dieser Abstraktionsinformation zu vergrößern. Zweitens gibt es die Spezialheuristiken, die sich zwar auch an den Skeletten und Kontexten der Abstraktionsinformation eines Gleichheitsproblems orientieren, sich aber nicht an die Ordnung $<_{\mathsf{GO}}$ halten. Diese werden dann verwendet, wenn das abstrakte Gleichheitsproblem nicht gelöst ist, erstere Heuristiken aber blockiert sind. Diese Heuristiken sind somit mit Vorsicht einzusetzen.

In diesem Abschnitt wird zunächst die Ordnung $<_{\mathsf{GO}}$ definiert und bewiesen, daß diese noethersch ist. Anschließend wird geklärt, wie Heuristiken der oben definierten ersten Art sich an diese Ordnung halten. Desweiteren wird Reduktion des Abstraktionsniveaus diskutiert, und unter welchen Umständen diese eine Heuristik der ersten Art ist. Zuletzt wird diskutiert, wie sich die Heuristiken der zweiten Art in das Gesamtverfahren integrieren lassen, ohne die Terminierung des Verfahrens zu gefährden.

5.2 Verfahren auf den abstrakten Termen

Alle beschriebenen Heuristiken aus Kapitel 3, mit Ausnahme der Heuristik zur Reduktion des Abstraktionsniveaus, der Variante der Skelettveränderung aus Abschnitt 3.2.4.3 und der Heuristik zur Skeletterweiterung aus Abschnitt 3.2.4.4, müssen im Zusammenspiel terminieren. Dafür wird eine partielle noethersche Ordnung eingeführt, und nach jeder Anwendung einer Gleichung durch eine dieser Heuristiken wird überprüft, ob das neue abstrakte Gleichheitsproblem im Sinne dieser partiellen Ordnung kleiner ist als das ursprüngliche.

Diese partielle Ordnung, die mit $<_{\mathsf{GO}}$ bezeichnet wird, setzt sich aus zwei unabhängigen partiellen Ordnungen zusammen. Zum einen wird eine partielle Ordnung $<_{\mathsf{AI}}$ auf den Abstraktionsinformationen definiert. Diese richtet sich nach den darin enthaltenen Skeletten (siehe Abschnitt 3.1.3) und den Kontexten (siehe Abschnitt 3.1.4) bezüglich dieser Skelette. Zum anderen wird eine partielle Ordnung $<_{\mathsf{Occ}}$ auf den abstrakten Gleichheitsproblemen definiert. Diese orientiert sich an der Anzahl der Vorkommen der betrachteten gemeinsamen Variablen. Von beiden partielle Ordnungen wird bewiesen, daß es partielle noethersche Ordnungen sind. Diese Eigenschaften werden dann dazu verwendet zu zeigen, daß $<_{\mathsf{GO}}$ eine partielle noethersche Ordnung ist.

5.2.1 Teilordnung auf Abstraktionsinformationen

Die partielle Ordnung $<_{\mathsf{AI}}$ berücksichtigt die Skelette und die Kontexte aus einer Abstraktionsinformation. Für deren Definition werden Funktionen benötigt, die die Größe der Kontexte einer Funktionssymbolliste bezüglich eines ihrer Skelette berechnen.

Die Größe des Kontextes einer Funktionssymboliste bezüglich eines ihrer Skelette ist zunächst einmal die Anzahl der in dieser Liste enthaltenen Kontexte. Zum Beispiel ist die Größe des Kontextes der Funktionssymbolliste $[(\mathsf{f},1),(\mathsf{g},1),(\mathsf{h},2)]$ bezüglich des Skelettes $[(\mathsf{g},1)]$ 2. Dies führt dazu, daß zum Beispiel die Funktionssymbolliste $[(\mathsf{f},1),(\mathsf{h},2),(\mathsf{g},1)]$ dieselbe Kontextgröße hat. Um dies zu vermeiden, wird für jeden Kontext die Position dieses Kontextes in der betrachteten Funktionssymbolliste betrachtet. Dabei ist $\mathsf{0}$ die erste Listenposition. Dies führt dazu, daß die erste Funktionssymbolliste die Kontextgröße 2 und die zweite die Kontextgröße 1 erhalten. Dadurch werden sie unterscheidbar.

Definition 5.1 (Kontextgröße einer Funktionssymbolliste) *Sei eine Funktionssymbolliste* fsl *und ein Skelett* $\mathsf{skeleton}$ *dieser Liste gegeben. Dann ist die* Größe des Kontextes *in* fsl *bezüglich* $\mathsf{skeleton}$ *die Summe der Positionen der Kontexte in* fsl *bezüglich* $\mathsf{skeleton}$.

Notation: $\omega(\mathsf{fsl}, \mathsf{skeleton})$.

Für ein Tripel $(\mathsf{fsl}_1, \mathsf{fsl}_2, \mathsf{skeleton})$ aus einer Abstraktionsinformation berechnet sich die Größe des Kontextes aus der Summe der Kontextgrößen von fsl_1 und fsl_2 bezüglich $\mathsf{skeleton}$. Alle Kontexte einer Abstraktionsinformation zusammen betrachtet, werden im folgendem als **Kontext der Abstraktionsinformation** bezeichnet. Die Größe des Kontextes einer Abstraktionsinformation ist definiert als die Summe der Kontextgrößen der einzelnen Tripel dieser Abstraktionsinformation.

Definition 5.2 (Kontextgröße einer Abstraktionsinformation) *Sei eine Abstraktionsinformation* AI *gegeben. Dann ist die* Größe des Kontextes dieser Abstraktionsinformation *definiert als*

$$\Omega(\mathsf{AI}) := \sum_{(\mathsf{fsl}_1, \mathsf{fsl}_2, \mathsf{skeleton})} (\omega(\mathsf{fsl}_1, \mathsf{skeleton}) + \omega(\mathsf{fsl}_2, \mathsf{skeleton}))$$

Die partielle Ordnung $<_{\mathsf{AI}}$ auf den Abstraktionsinformationen vergleicht die Kontextgrößen zweier Abstraktionsinformationen.

Definition 5.3 (Ordnung $<_{\mathsf{AI}}$) *Seien zwei Abstraktionsinformationen* AI *und* AI' *gegeben. Dann ist die Ordnung* $<_{\mathsf{AI}}$ *wie folgt definiert:*

$$\mathsf{AI} <_{\mathsf{AI}} \mathsf{AI}' :\Leftrightarrow \Omega(\mathsf{AI}) < \Omega(\mathsf{AI}')$$

Die reflexive Relation $\leq_{\mathsf{AI}}$ *zu* $<_{\mathsf{AI}}$ *läßt sich wie folgt definieren:*

$$\mathsf{AI} \leq_{\mathsf{AI}} \mathsf{AI}' :\Leftrightarrow \Omega(\mathsf{AI}) \leq \Omega(\mathsf{AI}')$$

Satz 5.1 *Die Relation* $<_{\mathsf{AI}}$ *ist eine partielle Ordnung.*

Beweis 5.1

Irreflexivität: *Sei eine Abstraktionsinformation* AI *gegeben. Da* $\Omega(\mathsf{AI}) < \Omega(\mathsf{AI})$ *nicht gilt, ist* $<_{\mathsf{AI}}$ *irreflexiv.*

Transitivität: *Seien drei Abstraktionsinformationen* AI, AI' *und* AI'' *gegeben, für die gilt:*

$$\mathsf{AI} <_{\mathsf{AI}} \mathsf{AI}' \quad \text{und} \quad \mathsf{AI}' <_{\mathsf{AI}} \mathsf{AI}''$$

Es gelten also: $\Omega(\mathsf{AI}) < \Omega(\mathsf{AI}')$ *und* $\Omega(\mathsf{AI}') < \Omega(\mathsf{AI}'')$. *Aus der Transitivität von* $<$ *folgt, daß dann auch* $\Omega(\mathsf{AI}) < \Omega(\mathsf{AI}'')$ *gilt, und somit gilt* $\mathsf{AI} <_{\mathsf{AI}} \mathsf{AI}''$. *Also ist* $<_{\mathsf{AI}}$ *transitiv.*

■

Anmerkung: Die partielle Ordnung $<_{\mathsf{AI}}$ ist sogar total.

Satz 5.2 *Die partielle Ordnung* $<_{\mathsf{AI}}$ *ist noethersch.*

Beweis 5.2 **Annahme:** *Es gibt eine unendliche Menge von Abstraktionsinformationen* AI_i, $\mathsf{i} = 1 \ldots \infty$, *so daß für alle* $\mathsf{i} \geq 1$ *gilt:*

$$\mathsf{AI}_{\mathsf{i}+1} <_{\mathsf{AI}} \mathsf{AI}_\mathsf{i}$$

Daraus folgt, daß es eine unendlich absteigende Kette $\Omega(\mathsf{AI}_\mathsf{i})$ *auf den natürlichen Zahlen bezüglich der Ordnung* $<$ *geben muß. Dies ist aber ein Widerspruch dazu, daß* $<$ *eine noethersche Ordnung ist. Also ist* $<_{\mathsf{AI}}$ *eine noethersche Ordnung.* ■

5.2.2 Teilordnung auf abstrakten Gleichungen

Die zweite Ordnung $<_{\mathsf{Occ}}$ soll zu zwei gegebenen Paaren von abstrakten Termen die Differenz der Anzahl der Vorkommen von gemeinsamen Untertermen vergleichen. Deshalb müssen die betrachteten Vorkommen von gemeinsamen Unterterme als zusätzliche Information mitgegeben werden. Deshalb wird die Ordnung $<_{\mathsf{Occ}}$ auf Tripeln der Form $(\mathsf{s}, \mathsf{t}, (\Pi_\mathsf{s}, \Pi_\mathsf{t}))$ definiert, wobei s und t die Abstraktionen zweier konkreter Terme bezüglich der Vorkommen $(\Pi_\mathsf{s}, \Pi_\mathsf{t})$ sind:

Die Summe der Differenzen der Anzahl der Vorkommen aller Vorkommen $(\Pi_\mathsf{s}, \Pi_\mathsf{t})$ gemeinsamer Unterterme in zwei Termen s und t ist wie folgt definiert:

Definition 5.4 *Gegeben seien zwei Terme* s *und* t *aus* $\mathsf{T}(\Sigma, \mathcal{V})$ *und eine Menge* $(\Pi_\mathsf{s}, \Pi_\mathsf{t})$ *von Vorkommen von Termen aus* $\mathsf{T}(\Sigma, \mathcal{V})$ *gegeben. Weiterhin sei die* M *Menge der durch* Π_s *und* Π_t *denotierten gemeinsamen Unterterme von* s *und* t *wie folgt definiert:*

$$\mathsf{M} := \mathsf{Subterms}(\mathsf{s}, \mathsf{t}, (\Pi_\mathsf{s}, \Pi_\mathsf{t}))$$

Dann ist die Funktion Δ *definiert als*

$$\Delta(\mathsf{s}, \mathsf{t}, \mathsf{M}) := \sum_{\mathsf{u} \in \mathsf{M}} \mid \#_\mathsf{u}(\mathsf{s}) - \#_\mathsf{u}(\mathsf{t}) \mid$$

Mit Hilfe dieser Funktion läßt sich die Ordnung $<_{\alpha c}$ wie folgt definieren:

Definition 5.5 (Ordnung $<_{\alpha c}$) *Gegeben seien die Terme* $s_i, t_i \in AT(\Sigma, \mathcal{V})$, $i = 1, 2$. *Weiterhin seien* (Π_{s_1}, Π_{t_1}) *und* (Π_{s_2}, Π_{t_2}) *Mengen von Vorkommen von Termen aus* $T(\Sigma, \mathcal{V})$. *Dann ist die Ordnung* $<_{\alpha c}$ *wie folgt definiert:*

$$\begin{aligned}(s_1, t_1, (\Pi_{s_1}, \Pi_{t_1})) &<_{\alpha c} (s_2, t_2, (\Pi_{s_2}, \Pi_{t_2})) \\ &:\Leftrightarrow \Delta(s_1, t_1, (\Pi_{s_1}, \Pi_{t_1})) < \Delta(s_2, t_2, (\Pi_{s_2}, \Pi_{t_2}))\end{aligned}$$

Entsprechend wird das reflexive Pendant zu $<_{\alpha c}$ *wie folgt definiert:*

$$\begin{aligned}(s_1, t_1, (\Pi_{s_1}, \Pi_{t_1})) &\leq_{\alpha c} (s_2, t_2, (\Pi_{s_2}, \Pi_{t_2})) \\ &:\Leftrightarrow \Delta(s_1, t_1, (\Pi_{s_1}, \Pi_{t_1})) \leq \Delta(s_2, t_2, (\Pi_{s_2}, \Pi_{t_2}))\end{aligned}$$

Anmerkung: Die partielle Ordnung $<_{\alpha c}$ ist eine totale Ordnung.

Satz 5.3 *Die Relation* $<_{\alpha c}$ *ist eine partielle Ordnung.*

Beweis 5.3

Irreflexivität: *Die Relation* $<_{\alpha c}$ *ist irreflexiv, da* $\Delta(s, t, (\Pi_s, \Pi_t)) < \Delta(s, t, (\Pi_s, \Pi_t))$ *nicht gilt.*

Transitivität: *Seien* s_i *und* t_i, $i = 1, 2, 3$, *abstrakte Terme bezüglich der respektiven Mengen von Vorkommen* (Π_{s_i}, Π_{t_i}), $i = 1, 2, 3$, *von Untertermen, so daß gilt:*

$$\begin{gathered}(s_1, t_1, (\Pi_{s_1}, \Pi_{t_1})) <_{\alpha c} (s_2, t_2, (\Pi_{s_2}, \Pi_{t_2})) \\ \textit{und} \\ (s_2, t_2, (\Pi_{s_2}, \Pi_{t_2})) <_{\alpha c} (s_3, t_3, (\Pi_{s_3}, \Pi_{t_3})),\end{gathered}$$

d. h. es gelten $\Delta(s_1, t_1, (\Pi_{s_1}, \Pi_{t_1})) < \Delta(s_2, t_2, (\Pi_{s_2}, \Pi_{t_2}))$ *und* $\Delta(s_2, t_2, (\Pi_{s_2}, \Pi_{t_2})) < \Delta(s_3, t_3, (\Pi_{s_3}, \Pi_{t_3}))$. *Daraus folgt, daß* $\Delta(s_1, t_1, (\Pi_{s_1}, \Pi_{t_1})) < \Delta(s_3, t_3, (\Pi_{s_3}, \Pi_{t_3}))$ *gilt und folglich auch:*

$$(s_1, t_1, (\Pi_{s_1}, \Pi_{t_1})) <_{\alpha c} (s_3, t_3, (\Pi_{s_3}, \Pi_{t_3})).$$

Daraus folgt, daß $<_{\alpha c}$ *transitiv ist.*

■

Satz 5.4 *Die partielle Ordnung* $<_{\alpha c}$ *ist noethersch.*

Beweis 5.4 *Gegeben sei eine unendliche Kette von Tripeln* (s_i, t_i, M_i), $i = 1 \ldots \infty$, M_i *Paare von Vorkommen von Termen in* s *und* t, *so daß für alle* $i \geq 1$ *gilt:*

$$(s_{i+1}, t_{i+1}, M_{i+1}) <_{\alpha c} (s_i, t_i, M_i)$$

Dann gibt es eine unendlich absteigende Kette $\Delta(s_i, t_i, M_i)$, $i = 1 \ldots \infty$, *auf den natürlichen Zahlen bezüglich der Ordnung* $<$ *auf den natürlichen Zahlen. Dies ist aber ein Widerspruch dazu, daß* $<$ *eine noethersche Ordnung auf den natürlichen Zahlen ist. Also ist* $<_{\alpha c}$ *noethersch.* ■

5.2.3 Die Gesamtordnung

Die Gesamtordnung $<_{GO}$ ist eine Vereinigung der beiden Teilordnungen $<_{AI}$ und $<_{Occ}$. Die Semantik dieser Ordnung ist wie folgt: Ein abstraktes Gleichheitsproblem $s \equiv t$, betrachtet unter der Menge M von Vorkommen von gemeinsamen Untertermen ist kleiner als ein abstraktes Gleichheitsproblem $s' \equiv t'$ unter M' betrachtet, wenn entweder die Größe des Kontextes in $s \equiv t$ kleiner ist als der Kontext in $s' \equiv t'$ oder die Differenz der Anzahl der Vorkommen in $s \equiv t$ ist kleiner als die in $s' \equiv t'$. Je nachdem ob die Kontextgröße oder die Differenz der Anzahl der Vorkommen abnimmt, muß gelten daß die jeweils andere Größe mindestens gleich bleibt oder auch abnimmt. Dadurch wird gewährleistet, daß die Relation $<_{GO}$ eine partielle Ordnung auf abstrakten Gleichheitsproblemen ist. Zusätzlich werden auch die respektiven betrachteten Vorkommen von Untertermen berücksichtigt, da die Ordnung $<_{Occ}$ dies benötigt. Deshalb ist die Ordnung $<_{GO}$ eine partielle Ordnung über Tripeln (s, t, M), wie die Ordnung $<_{Occ}$, mit M Paar von Vorkommen von gemeinsamen Untertermen von s und t. Da die Ordnung $<_{AI}$ die Abstraktionsinformation der beiden Terme s und t bezüglich M benötigt, wird diese mit Hilfe der in Abschnitt 3.2.3 eingeführten Funktion AI berechnet.

Damit läßt sich die Ordnung $<_{GO}$ wie folgt definieren:

Definition 5.6 *Seien die Terme* $s_i, t_i \in T(\Sigma, \mathcal{V})$, $i = 1, 2$, *gegeben. Weiterhin seien* M_1 *bzw.* M_2 *Mengen von Vorkommen gemeinsamen Untertermen von* s_1 *und* t_1 *bzw. von* s_2 *und* t_2. *Dann ist die Ordnung* $<_{GO}$ *wie folgt definiert:*

$$(s_1, t_1, M_1) <_{GO} (s_2, t_2, M_2) :\Leftrightarrow \begin{cases} \text{Entweder} & AI(s_1, t_1, M_1) <_{AI} AI(s_2, t_2, M_2) \text{ und} \\ & (s_1, t_1, M_1) \leq_{Occ} (s_2, t_2, M_2) \\ \text{oder} & AI(s_1, t_1, M_1) \leq_{AI} AI(s_2, t_2, M_2) \text{ und} \\ & (s_1, t_1, M_1) <_{Occ} (s_2, t_2, M_2) \end{cases}$$

Nun werden noch die beiden Theoreme angegeben, die das Ziel dieses Abschnittes waren: Nämlich daß die definierte Relation $<_{GO}$ eine noethersche, partielle Ordnung ist. Damit ist gewährleistet, daß das Zusammenspiel der Heuristiken aus Kapitel 3 terminiert.

Satz 5.5 *Die Relation* $<_{GO}$ *ist eine partielle Ordnung.*

Beweis 5.5

Irreflexivität: *Sei ein Tripel* (s, t, M) *gegeben. Da sowohl* $<_{AI}$ *als auch* $<_{Occ}$ *irreflexiv sind, gilt weder* $AI(s, t, M) <_{AI} AI(s, t, M)$, *noch* $(s, t, M) <_{Occ} (s, t, M)$. *Also ist* $<_{GO}$ *irreflexiv.*

Transitivität: *Seien drei Tripel* (s_i, t_i, M_i), $i = 1, 2, 3$, *gegeben, so daß gilt:*

$$(s_1, t_1, M_1) <_{GO} (s_2, t_2, M_2) \quad \text{und} \quad (s_2, t_2, M_2) <_{GO} (s_3, t_3, M_3)$$

Falls $\mathsf{AI}(s_1, t_1, M_1) <_{\mathsf{AI}} \mathsf{AI}(s_2, t_2, M_2)$: *Da auch* $\mathsf{AI}(s_2, t_2, M_2) \leq_{\mathsf{AI}} \mathsf{AI}(s_3, t_3, M_3)$ *gilt, gilt:*$\mathsf{AI}(s_1, t_1, M_1) <_{\mathsf{AI}} \mathsf{AI}(s_3, t_3, M_3)$*, da* $<_{\mathsf{AI}}$ *eine totale Ordnung ist. Weiterhin gelten mindestens* $(s_1, t_1, M_1) \leq_{\mathsf{Occ}} (s_2, t_2, M_2)$ *und* $(s_2, t_2, M_2) <_{\mathsf{Occ}} (s_3, t_3, M_3)$*. Somit gilt auch* $(s_1, t_1, M_1) \leq_{\mathsf{Occ}} (s_3, t_3, M_3)$*, da* $<_{\mathsf{Occ}}$ *auch eine totale Ordnung ist.*

Falls $\mathsf{AI}(s_1, t_1, M_1) \leq_{\mathsf{AI}} \mathsf{AI}(s_2, t_2, M_2)$*; D. h. es gilt:* $(s_1, t_1, M_1) <_{\mathsf{Occ}} (s_2, t_2, M_2)$*. Weiterhin gelten mindestens* $\mathsf{AI}(s_2, t_2, M_2) \leq_{\mathsf{AI}} \mathsf{AI}(s_3, t_3, M_3)$ *und* $(s_2, t_2, M_2) \leq_{\mathsf{Occ}} (s_3, t_3, M_3)$*. Da* $<_{\mathsf{AI}}$ *und* $<_{\mathsf{Occ}}$ *beide totale Ordnungen sind, folgt dann:*

$$\mathsf{AI}(s_1, t_1, M_1) \leq_{\mathsf{AI}} \mathsf{AI}(s_3, t_3, M_3) \quad \text{und} \quad (s_1, t_1, M_1) <_{\mathsf{Occ}} (s_3, t_3, M_3)$$

Somit gilt: $(s_1, t_1, M_1) <_{\mathsf{GO}} (s_3, t_3, M_3)$.

Somit ist $<_{\mathsf{GO}}$ *transitiv.*

■

Satz 5.6 *Die partielle Ordnung* $<_{\mathsf{GO}}$ *ist noethersch.*

Beweis 5.6 *Sei eine unendliche Kette* (s_i, t_i, M_i)*,* $i = 1 \ldots \infty$*, gegeben, so daß für alle* $i \geq 1$ *gilt:*

$$(s_{i+1}, t_{i+1}, M_{i+1}) <_{\mathsf{GO}} (s_i, t_i, M_i)$$

Dann gibt es eine unendliche Liste $(s_{i_j}, t_{i_j}, M_{i_j})$*,* $j = 1 \ldots \infty$*, für die entweder*

$$\mathsf{AI}(s_{i_{j+1}}, t_{i_{j+1}}, M_{i_{j+1}}) <_{\mathsf{AI}} \mathsf{AI}(s_{i_j}, t_{i_j}, M_{i_j})$$

für alle $j \geq 1$ *gilt oder*

$$(s_{i_{j+1}}, t_{i_{j+1}}, M_{i_{j+1}}) <_{\mathsf{Occ}} (s_{i_j}, t_{i_j}, M_{i_j})$$

für alle $j \geq 1$ *gilt. Der erste Fall widerspricht der Tatsache, daß* $<_{\mathsf{AI}}$ *noethersch ist und der zweite Fall der Tatsache, daß* $<_{\mathsf{Occ}}$ *noethersch ist. Also kann es keine solche unendlich absteigende Kette* (s_i, t_i, M_i)*,* $i = 1 \ldots \infty$*, geben. Also ist* $<_{\mathsf{GO}}$ *noethersch.*

■

5.2.4 Die Strategien und die Terminierungsordnung

Alle in Kapitel 3 vorgestellten Heuristiken, mit Ausnahme der Heuristik zur Reduktion des Abstraktionsniveaus, der Variante zur Skelettveränderung und der speziellen skeletterweiternden Heuristik, halten sich an die in Abschnitt 5.2 definierte Ordnung $<_{GO}$, d. h. daß wenn eine Heuristik eine gerichtete Gleichung $\mathsf{u} \rightarrow \mathsf{v}$ zum Beispiel auf den linken Term eines abstrakten Gleichheitsproblems $\mathsf{s} \equiv \mathsf{t}$ an einer Termposition π angewendet hat, gilt:

$$(\mathsf{s}_{|\pi \leftarrow \sigma(\mathsf{v})}, \mathsf{t}, \mathsf{M}') <_{GO} (\mathsf{s}, \mathsf{t}, \mathsf{M})$$

Dabei ist σ die für die Anwendung von $\mathsf{u} \rightarrow \mathsf{v}$ benötigte Substitution und M die Menge der Vorkommen von betrachteten Untertermen. Solange also weder das Abstraktionsniveau reduziert wird, noch die Variante der skelettverändernden Heuristik oder die spezielle skeletterweiternde Heuristik verwendet werden, wird durch diese Heuristiken eine Ableitung

$$\begin{array}{ccccc} \mathsf{s}_1 \equiv \mathsf{t}_1 & \vdash & \ldots & \vdash & \mathsf{s}_n \equiv \mathsf{t}_n \\ \mathsf{M}_1 & & & & \mathsf{M}_n \end{array}$$

des ursprünglichen Gleichheitsproblems $\mathsf{s}_1 \equiv \mathsf{t}_1$ erhalten. Dabei sind die M_i die respektiven Mengen von Vorkommen von gemeinsamen Untertermen. Für die einzelnen Ableitungsschritte gilt:

$$(\mathsf{s}_{i+1}, \mathsf{t}_{i+1}, \mathsf{M}_{i+1}) <_{GO} (\mathsf{s}_i, \mathsf{t}_i, \mathsf{M}_i)$$

In den folgenden beiden Abschnitten wird gezeigt, daß die Reduktion des Abstraktionsniveaus verträglich ist mit der Ordnung $<_{GO}$. Somit ist die Terminierung des Zusammenspiels der Heuristiken und der Reduktion des Abstraktionsniveaus gewährleistet. Anschließend wird gezeigt, wie die Reduktion der Repräsentation durch Multimengen von Funktionssymbolen sich in die anderen Heuristiken integrieren läßt, so daß die Terminierung des daraus resultierenden Gesamtverfahrens gewährleistet ist.

5.2.5 Integration der Reduktion des Abstraktionsniveaus

Die Idee der Reduktion des Abstraktionsniveaus war anstelle eines gemeinsamen Unterterms $\mathsf{f}(\mathsf{t}_1, \ldots, \mathsf{t}_n)$ eines Gleichheitsproblems $\mathsf{s} \equiv \mathsf{t}$ die Unterterme t_1, $\ldots \mathsf{t}_n$ in diesem Gleichheitsproblem zu betrachten. Dadurch ändert sich die Menge $\mathsf{Subterms}(\mathsf{s}, \mathsf{t}, \mathsf{M})$, M Menge von Vorkommen von gemeinsamen Untertermen von s und t, der betrachteten gemeinsamen Unterterme wie folgt:

$$\mathsf{Subterms}(\mathsf{s}, \mathsf{t}, \mathsf{M}) \longrightarrow (\mathsf{Subterms}(\mathsf{s}, \mathsf{t}, \mathsf{M}) \setminus \{\mathsf{f}(\mathsf{t}_1, \ldots, \mathsf{t}_n)\}) \cup \{\mathsf{t}_1, \ldots, \mathsf{t}_n\}$$

Die Frage ist, ob dies mit der Ordnung $<_{GO}$ aus Abschnitt 5.2 verträglich ist. Im folgenden wird gezeigt werden, daß dies in der hier betrachteten Situation, in dem nur Matcher zur Anwendung einer Gleichung erlaubt sind, der Fall ist.

Um zu zeigen, daß die Reduktion des Abstraktionsniveaus die Terminierung des Gesamtverfahrens nicht gefährdet, wird eine partielle Ordnung $\ll_{\mathsf{Depth}}$ auf den betrachteten gemeinsamen Untertemen eingeführt. Mit Hilfe dieser Ordnung und der bisherigen Ordnung $<_{\mathsf{GO}}$ kann eine neue Gesamtordnung $<_{\mathsf{GO}_1}$ definiert werden. Für die Definition der Ordnung $\ll_{\mathsf{Depth}}$, wird eine Ordnung $<_{\mathsf{Depth}}$ auf Termen zu einer Signatur Σ und einer Variablenmenge $\mathcal{V}$ definiert. Diese Ordnung ordnet die Terme nach deren Größe (siehe Abschnitt 3.2.1 auf Seite 32).

Definition 5.7 *Seien zwei beliebige Term* t_1 *und* t_2 *zu einer Signatur* Σ *und einer Variablenmenge* $\mathcal{V}$ *gegeben. Dann gilt* $\mathsf{t}_1 <_{\mathsf{Depth}} \mathsf{t}_2$ *genau dann, wenn gilt*

$$\mathsf{Depth}(\mathsf{t}_1) < \mathsf{Depth}(\mathsf{t}_2)$$

Später wird die Eigenschaft bewiesen, daß die Ordnung $\ll_{\mathsf{Depth}}$ eine noethersche Ordnung ist. Dafür wird die Eigenschaft benötigt, daß die Ordnung $<_{\mathsf{Depth}}$ eine nothersche Ordnung ist. Die folgenden beiden Sätze sichern diese Eigenschaft der Ordnung $<_{\mathsf{Depth}}$.

Satz 5.7 *Die Relation* $<_{\mathsf{Depth}}$ *ist eine partielle Ordnung.*

Beweis 5.7 *Die Relation* $<_{\mathsf{Depth}}$ *ist irreflexiv, da für alle* $\mathsf{t} \in \mathsf{T}(\Sigma, \mathcal{V})$ *die Ungleichung* $\mathsf{Depth}(\mathsf{t}) < \mathsf{Depth}(\mathsf{t})$ *nicht gilt.*

Seien drei Terme t_1, t_2 *und* t_3 *aus* $\mathsf{T}(\Sigma, \mathcal{V})$ *gegeben für die gilt:* $\mathsf{t}_1 <_{\mathsf{Depth}} \mathsf{t}_2$ *und* $\mathsf{t}_2 <_{\mathsf{Depth}} \mathsf{t}_3$; *d. h. es gilt* $\mathsf{Depth}(\mathsf{t}_1) < \mathsf{Depth}(\mathsf{t}_2)$ *und* $\mathsf{Depth}(\mathsf{t}_2) < \mathsf{Depth}(\mathsf{t}_3)$. *Somit gilt auch* $\mathsf{Depth}(\mathsf{t}_1) < \mathsf{Depth}(\mathsf{t}_3)$, *und somit auch* $\mathsf{t}_1 <_{\mathsf{Depth}} \mathsf{t}_3$. *Also ist* $<_{\mathsf{Depth}}$ *auch transitiv, und deshalb eine partielle Ordnung.* ■

Anmerkung: Wie leicht zu sehen ist, ist die Ordnung $\ll_{\mathsf{Depth}}$ sogar total.

Satz 5.8 *Die partielle Ordnung* $<_{\mathsf{Depth}}$ *ist nothersch.*

Beweis 5.8 *Angenommen, es gäbe eine unedliche Menge von Termen* $\mathsf{t}_1, \mathsf{i} = 1 \ldots \infty$, *so daß für alle* $\mathsf{i} > 0$ *gilt:*

$$\mathsf{t}_{\mathsf{i}+1} <_{\mathsf{Depth}} \mathsf{t}_{\mathsf{i}}$$

Dies bedeutet, daß es eine unendlich absteigende Folge von natürlichen Zahlen gäbe. Dies widerspricht aber der Tatsache, daß die Ordnung $<$ *auf den natürlichen Zahlen nothersch ist. Folglich kann es keine solche unendlich vielen Terme* t_{i} *geben und somit ist* $<_{\mathsf{Depth}}$ *noethersch.* ■

Die Ordnung $\ll_{\mathsf{Depth}}$ vergleicht zwei Mengen von Termen zu einer Signatur $\mathsf{T}(\Sigma, \mathcal{V})$. Dabei wird die Ordnung als Multimengenordnung über den Multimengen von Termen aus $\mathsf{T}(\Sigma, \mathcal{V})$, aufbauend auf der Ordnung $<_{\mathsf{Depth}}$, definiert.

Definition 5.8 *Die Ordnung $\ll_{\mathsf{Depth}}$ ist die Multimengenerweiterung der Ordnung $<_{\mathsf{Depth}}$.*

Satz 5.9 *Die Relation $\ll_{\mathsf{Depth}}$ ist eine partielle Ordnung.*

Beweis 5.9 *Folgt direkt aus Satz 5.7 und daraus, daß $<_{\mathsf{Depth}}$ eine Ordnung ist.* ■

Satz 5.10 *Die partielle Ordnung $\ll_{\mathsf{Depth}}$ ist noethersch.*

Beweis 5.10 *Folgt direkt aus den Definition 5.8 und Satz 3.2.* ■

Mit Hilfe der Ordnung $\ll_{\mathsf{Depth}}$ kann die Ordnung $<_{\mathsf{GO}}$ zu einer Ordnung $<_{\mathsf{GO}_1}$ erweitert werden.

Definition 5.9 *Seien die Terme $\mathsf{s}_\mathsf{i}, \mathsf{t}_\mathsf{i} \in \mathsf{T}(\Sigma, \mathcal{V})$, $\mathsf{i} = 1, 2$, gegeben. Weiterhin seien M_1 und M_2 Mengen von Vorkommen von gemeinsamen Untertermen von s_1 und t_1 bzw. s_2 und t_2. Dann ist die Ordnung $<_{\mathsf{GO}_1}$ wie folgt definiert:*

$$
\begin{aligned}
&(\mathsf{s}_1, \mathsf{t}_1, \mathsf{M}_1) <_{\mathsf{GO}_1} (\mathsf{s}_2, \mathsf{t}_2, \mathsf{M}_2) \\
&\quad :\Leftrightarrow \begin{cases} \mathsf{Subterms}(\mathsf{s}_1, \mathsf{t}_1, \mathsf{M}_1) \ll_{\mathsf{Depth}} \mathsf{Subterms}(\mathsf{s}_2, \mathsf{t}_2, \mathsf{M}_2) \text{ oder} \\ \mathsf{Subterms}(\mathsf{s}_2, \mathsf{t}_2, \mathsf{M}_2) \not\ll_{\mathsf{Depth}} \mathsf{Subterms}(\mathsf{s}_1, \mathsf{t}_1, \mathsf{M}_1) \text{ und} \\ (\mathsf{s}_1, \mathsf{t}_1, \mathsf{M}_1) <_{\mathsf{GO}} (\mathsf{s}_2, \mathsf{t}_2, \mathsf{M}_2) \end{cases}
\end{aligned}
$$

Für die Ordnung $<_{\mathsf{GO}_1}$ gilt, daß die Reduktion des Abstraktionsniveaus verträglich mit ihr ist.

Daß die anderen Heuristiken verträglich sind mit der neuen Ordnung $<_{\mathsf{GO}_1}$ ist klar, da nur die Variablen in den angewendeten Gleichungen instantiiert werden. Deshalb bleiben die betrachteten Unterterme unverändert. Zudem ist die Anwendung einer Gleichung nur dann zulässig, wenn das resultierende Gleichheitsproblem bezüglich der Ordnung $<_{\mathsf{GO}}$ kleiner ist als das ursprüngliche. Daraus folgt, daß solange nur solche Ableitungsschritte gemacht werden, die daß Abstraktionsniveau nicht reduzieren, für jeden solchen Ableitungsschritt $(\mathsf{s}, \mathsf{t}, \mathsf{M}) \vdash (\mathsf{s}', \mathsf{t}', \mathsf{M}')$ gilt

$$(\mathsf{s}', \mathsf{t}', \mathsf{M}') <_{\mathsf{GO}_1} (\mathsf{s}, \mathsf{t}, \mathsf{M}),$$

da die Menge der betrachteten Unterterme (nicht der Vorkommen) gleich bleibt, das Gleichheitsproblem bezüglich M aber bezüglich der Ordnung $<_{\mathsf{GO}}$ kleiner wird.

Für die Reduzierung des Abstraktionsniveaus, was durch den Ableitungsschritt $(\mathsf{s},\mathsf{t},\mathsf{M}) \vdash (\mathsf{s}',\mathsf{t}',\mathsf{M}')$ ausgedrückt wird, gilt:

$$\mathsf{Subterms}(\mathsf{s}',\mathsf{t}',\mathsf{M}') \ll_{\mathsf{Depth}} \mathsf{Subterms}(\mathsf{s},\mathsf{t},\mathsf{M})$$

wodurch sichergestellt ist, daß die Reduktion des Abstraktionsniveaus verträglich ist mit der neuen Ordnung $<_{\mathsf{GO}_1}$. Dies wird in dem folgendem Theorem festgehalten.

Satz 5.11 *Sei eine nichtleere Menge von Termen* M *aus* $\mathsf{T}(\Sigma,\mathcal{V})$ *gegeben und sei* M' *eine Menge von Termen, die aus* M *durch Reduktion des Abstraktionsniveaus hervorgegangen ist. Dann gilt:*

$$\mathsf{M}' \ll_{\mathsf{Depth}} \mathsf{M}$$

Beweis 5.11 *In* M *gibt es einen Term* t*, der der Form* $\mathsf{f}(\mathsf{t}_1,\ldots,\mathsf{t}_\mathsf{n})$, $\mathsf{n} \geq 1$ *ist. Dann gilt:*

$$\mathsf{M}' = \mathsf{M} \setminus \{\mathsf{f}(\mathsf{t}_1,\ldots,\mathsf{t}_\mathsf{n})\} \cup \{\mathsf{t}_1,\ldots,\mathsf{t}_\mathsf{n}\}$$

Da $\mathsf{f}(\mathsf{t}_1,\ldots,\mathsf{t}_\mathsf{n})$ *aus* M *ist, folgt, daß* $\mathsf{M}' \neq \mathsf{M}$ *ist. Weiterhin gelten*

- $\mathsf{M} \setminus_{\mathsf{M}} \mathsf{M}' = \{\mathsf{f}(\mathsf{t}_1,\ldots,\mathsf{t}_\mathsf{n})\}$ *und*
- $\mathsf{M}' \setminus_{\mathsf{M}} \mathsf{M} = \{\mathsf{t}_1 \ldots \mathsf{t}_\mathsf{n}\}$

Für alle $1 \leq \mathsf{i} \leq \mathsf{n}$ *gilt:* $\mathsf{t}_\mathsf{i} <_{\mathsf{Depth}} \mathsf{f}(\mathsf{t}_1,\ldots,\mathsf{t}_\mathsf{n})$. *D. h. es gilt:*

$$\forall \mathsf{t} \in \mathsf{M}' \setminus_{\mathsf{M}} \mathsf{M}\ \exists \mathsf{s} \in \mathsf{M} \setminus_{\mathsf{M}} \mathsf{M}'\ \mathsf{t} <_{\mathsf{Depth}} \mathsf{s}$$

Also gilt: $\mathsf{M}' \ll_{\mathsf{Depth}} \mathsf{M}$, *woraus die Beahuptung folgt.* ■

An dieser Stelle werden jetzt die Beweise angegeben, die gewährleisten, daß $<_{\mathsf{GO}_1}$ eine nothersche partielle Ordnung ist.

Satz 5.12 *Die Relation* $<_{\mathsf{GO}_1}$ *ist eine partielle Ordnung.*

Beweis 5.12

Irreflexivität: *Sei ein Tripel* $(\mathsf{s},\mathsf{t},\mathsf{M})$, $\mathsf{M} \subseteq \Pi(\mathsf{s}) \times \Pi(\mathsf{t})$, *gegeben. Da sowohl* $\ll_{\mathsf{Depth}}$, *als auch* $<_{\mathsf{GO}}$ *irreflexiv sind, gilt weder* $\mathsf{Subterms}(\mathsf{s},\mathsf{t},\mathsf{M}) \ll_{\mathsf{Depth}} \mathsf{Subterms}(\mathsf{s},\mathsf{t},\mathsf{M})$, *noch* $(\mathsf{s},\mathsf{t},\mathsf{M}) <_{\mathsf{GO}} (\mathsf{s},\mathsf{t},\mathsf{M})$. *Somit gilt auch nicht* $(\mathsf{s},\mathsf{t},\mathsf{M}) <_{\mathsf{GO}_1} (\mathsf{s},\mathsf{t},\mathsf{M})$. *Also ist* $<_{\mathsf{GO}_1}$ *irreflexiv.*

Transitivität: *Seien drei Tripel* $(\mathsf{s},\mathsf{t},\mathsf{M}), \mathsf{M} \subseteq \Pi(\mathsf{s})\times\Pi(\mathsf{t}), (\mathsf{s}',\mathsf{t}',\mathsf{M}'), \mathsf{M}' \subseteq \Pi(\mathsf{s}')\times \Pi(\mathsf{t}')$ *und* $(\mathsf{s}'',\mathsf{t}'',\mathsf{M}''), \mathsf{M}'' \subseteq \Pi(\mathsf{s}'') \times \Pi(\mathsf{t}'')$ *gegeben für die gilt:*

$$(\mathsf{s},\mathsf{t},\mathsf{M}) <_{\mathbb{GO}_1} (\mathsf{s}',\mathsf{t}',\mathsf{M}') \quad \text{und} \quad (\mathsf{s}',\mathsf{t}',\mathsf{M}') <_{\mathbb{GO}_1} (\mathsf{s}'',\mathsf{t}'',\mathsf{M}'')$$

Falls $\mathsf{Subterms}(\mathsf{s},\mathsf{t},\mathsf{M}) \ll_{\mathsf{Depth}} \mathsf{Subterms}(\mathsf{s}',\mathsf{t}',\mathsf{M}')$ **:**
Dann gilt $\mathsf{Subterms}(\mathsf{s},\mathsf{t},\mathsf{M}) \ll_{\mathsf{Depth}} \mathsf{Subterms}(\mathsf{s}'',\mathsf{t}'',\mathsf{M}'')$, *da* $\ll_{\mathsf{Depth}}$ *eine totale Ordnung ist. Also gilt auch* $(\mathsf{s},\mathsf{t},\mathsf{M}) <_{\mathbb{GO}_1} (\mathsf{s}'',\mathsf{t}'',\mathsf{M}'')$.

Falls $\mathsf{Subterms}(\mathsf{s},\mathsf{t},\mathsf{M}) \not\ll_{\mathsf{Depth}} \mathsf{Subterms}(\mathsf{s}',\mathsf{t}',\mathsf{M}')$ **und** $\mathsf{Subterms}(\mathsf{s}',\mathsf{t}',\mathsf{M}') \not\ll_{\mathsf{Depth}} \mathsf{Subterms}(\mathsf{s},\mathsf{t},\mathsf{M})$:
Dann gilt, $(\mathsf{s},\mathsf{t},\mathsf{M}) <_{\mathbb{GO}} (\mathsf{s}',\mathsf{t}',\mathsf{M}')$.

Falls $\mathsf{Subterms}(\mathsf{s}',\mathsf{t}',\mathsf{M}') \ll_{\mathsf{Depth}} \mathsf{Subterms}(\mathsf{s}'',\mathsf{t}'',\mathsf{M}'')$ **:**
Da $\ll_{\mathsf{Depth}}$ *totale Ordnung, gilt* $\mathsf{Subterms}(\mathsf{s},\mathsf{t},\mathsf{M}) \ll_{\mathsf{Depth}} \mathsf{Subterms}(\mathsf{s}'',\mathsf{t}'',\mathsf{M}'')$, *und somit gilt:* $(\mathsf{s},\mathsf{t},\mathsf{M}) <_{\mathbb{GO}_1} (\mathsf{s}'',\mathsf{t}'',\mathsf{M}'')$.

Falls $\mathsf{Subterms}(\mathsf{s}',\mathsf{t}',\mathsf{M}') \not\ll_{\mathsf{Depth}} \mathsf{Subterms}(\mathsf{s}'',\mathsf{t}'',\mathsf{M}'')$ **und** $\mathsf{Subterms}(\mathsf{s}'',\mathsf{t}'',\mathsf{M}'') \not\ll_{\mathsf{Depth}} \mathsf{Subterms}(\mathsf{s}',\mathsf{t}',\mathsf{M}')$: *Dann gilt einerseits:*
$\mathsf{Subterms}(\mathsf{s},\mathsf{t},\mathsf{M}) \not\ll_{\mathsf{Depth}} \mathsf{Subterms}(\mathsf{s}'',\mathsf{t}'',\mathsf{M}'')$ *und*
$\mathsf{Subterms}(\mathsf{s}'',\mathsf{t}'',\mathsf{M}'') \not\ll_{\mathsf{Depth}} \mathsf{Subterms}(\mathsf{s},\mathsf{t},\mathsf{M})$
Andererseits gilt dann auch $(\mathsf{s}',\mathsf{t}',\mathsf{M}') <_{\mathbb{GO}} (\mathsf{s}'',\mathsf{t}'',\mathsf{M}'')$, *; da* $<_{\mathbb{GO}}$ *partielle Ordnung ist, gilt* $(\mathsf{s},\mathsf{t},\mathsf{M}) <_{\mathbb{GO}} (\mathsf{s}'',\mathsf{t}'',\mathsf{M}'')$. *Somit gilt:* $(\mathsf{s},\mathsf{t},\mathsf{M}) <_{\mathbb{GO}_1} (\mathsf{s}'',\mathsf{t}'',\mathsf{M}'')$.

■

Satz 5.13 *Die partielle Ordnung* $<_{\mathbb{GO}_1}$ *ist noethersch.*

Beweis 5.13 **Annahme:** *Es gibt eine unendliche Folge* $(\mathsf{s}_\mathsf{i},\mathsf{t}_\mathsf{i},\mathsf{M}_\mathsf{i})$, $\mathsf{i} = 1,\ldots\infty$, $\mathsf{M}_\mathsf{i} \subseteq \Pi(\mathsf{s}_\mathsf{i}) \times \Pi(\mathsf{t}_\mathsf{i})$, *so daß für alle* $\mathsf{i} \geq 1$ *gilt:*

$$(\mathsf{s}_{\mathsf{i}+1},\mathsf{t}_{\mathsf{i}+1},\mathsf{M}_{\mathsf{i}+1}) <_{\mathbb{GO}_1} (\mathsf{s}_\mathsf{i},\mathsf{t}_\mathsf{i},\mathsf{M}_\mathsf{i})$$

Daraus folgt, daß es entweder eine unendliche Teilfolge $(\mathsf{s}_{\mathsf{i}_\mathsf{j}},\mathsf{t}_{\mathsf{i}_\mathsf{j}},\mathsf{M}_{\mathsf{i}_\mathsf{j}})$, $\mathsf{j} = 1\ldots\infty$, *so daß* $(\mathsf{s}_{\mathsf{i}_{\mathsf{j}+1}},\mathsf{t}_{\mathsf{i}_{\mathsf{j}+1}},\mathsf{M}_{\mathsf{i}_{\mathsf{j}+1}}) <_{\mathbb{GO}} (\mathsf{s}_{\mathsf{i}_\mathsf{j}},\mathsf{t}_{\mathsf{i}_\mathsf{j}},\mathsf{M}_{\mathsf{i}_\mathsf{j}})$ *für alle* $\mathsf{j} \geq 1$ *gilt. Dies widerspricht aber der Tatsache, daß* $<_{\mathbb{GO}}$ *noethersch ist. Oder es gibt eine unendliche Teilfolge von* $\mathsf{M}_{\mathsf{i}_\mathsf{k}}$, $\mathsf{k} = 1\ldots\infty$, *so daß*

$$\mathsf{Subterms}(\mathsf{s}_{\mathsf{i}_{\mathsf{k}+1}},\mathsf{t}_{\mathsf{i}_{\mathsf{k}+1}},\mathsf{M}_{\mathsf{i}_{\mathsf{k}+1}}) \ll_{\mathsf{Depth}} \mathsf{Subterms}(\mathsf{s}_{\mathsf{i}_\mathsf{k}},\mathsf{t}_{\mathsf{i}_\mathsf{k}},\mathsf{M}_{\mathsf{i}_\mathsf{k}})$$

für alle $\mathsf{k} \geq 1$ *gilt. Dies widerspricht aber der Tatsache, daß* $\ll_{\mathsf{Depth}}$ *noethersch ist.*

Folglich ist $<_{\mathbb{GO}_1}$ *eine noethersche Ordnung.* ■

Damit ist gezeigt, daß das Verfahren, bestehend aus den nicht kritischen Heuristiken aus Kapitel 3 und der Reduktion des Abstraktionsniveaus, terminiert.

5.3 Gesamtverfahren

Das Gesamtverfahren besteht einerseits aus den Heuristiken aus Kapitel 3, die sich an die Ordnung $<_{\mathsf{GO}}$ halten. Andererseits gibt es Heuristiken, welche die Terminierung des Verfahrens gefährden. Dies sind die Heuristik zur Reduktion des Abstraktionsniveaus (siehe Abschnitt 3.2.4.6), die Variante der Heuristik zur Skelettveränderung (siehe Abschnitt 3.2.4.3 auf Seite 48) und die spezielle skeletterweiternde Heuristik. Im vorangeganen Abschnitt wurde gezeigt, daß die Heuristik zur Reduktion des Abstraktionsniveaus die Terminierung des Verfahrens nicht gefährdet. In dem folgenden Abschnitten soll gezeigt werden, wie die Variante der skelettverändernden Heuristik gehandhabt werden kann, so daß die Terminierung des resultierenden Gesamtverfahren gewährleistet bleibt.

5.3.1 Spezielle skelettverändernde Heuristik

Diese Heuristik wird dann eingesetzt, wenn es nicht möglich ist, einen Kontext so zu eliminieren, daß das neue Gleichheitsproblem bezüglich der Ordnung $<_{\mathsf{GO}}$ kleiner ist als das Alte. Folgende Restriktionen können eingeführt werden, damit das Verfahren dennoch terminiert: Angenommen der zu eliminierende Kontext sei (f, i), dann darf diese Heuristik diesen Kontext nur dann eliminieren, wenn die dafür benötigte Gleichung auch das Funktionssymbol f eliminiert. Die eliminierten Funktionssymbole werden in einer Liste vermerkt und in keiner anderen Gleichungsanwendung einer beliebigen Heuristik darf eines dieser Funktionssymbole wieder erzeugt werden. Dadurch werden Zyklen vermieden und die Terminierung des Gesamtverfahrens bleibt gewährleistet.

5.3.2 Spezielle skeletterweiternde Heuristik

Diese Heuristik kann im Prinzip genau so in das Gesamtverfahren eingebunden werden, wie die spezielle skelettverändernde Heuristik. Dies bedeutet konkret, daß wenn diese Heuristik eine Skeletterweiterung durchführt, dabei auch ein Kontext eliminiert werden muß. Das Funktionssymbol dieses Kontextes wird in der selben Liste wie im vorherigen Abschnitt vermerkt, und darf anschließend nicht mehr erzeugt werden. Dadurch bleibt die Terminierung des Verfahrens gewahrt, wie schon im obigen Abschnitt beschrieben wurde.

5.4 Strukturtermersetzungssystem

Definition 5.10 (Die Terminierungsordnung des Ersetzungssystems)
Gegeben sei eine Strukturgleichung $\mathsf{s} \equiv \mathsf{t}$ *zu einer Signatur* Σ *und einer Variablenmenge* $\mathcal{V}$*. Dann seien* $\mathsf{Fu}(\mathsf{s})$ *und* $\mathsf{Fu}(\mathsf{t})$ *die respektiven Multimengen von*

Funktionssymbolen von s *und* t. *Dann definiert sich die* Terminierungsordnung $<_{\mathsf{TO}}$ *wie folgt:*

$$\mathsf{s} <_{\mathsf{TO}} \mathsf{t}$$

genau dann, wenn $\mathsf{Fu}(\mathsf{s}) \ll_{\mathsf{F}} \mathsf{Fu}(\mathsf{t})$ *und für alle Variablen* $\mathsf{x} \in \mathcal{V}$ *gilt:* $\#_{\mathsf{x}}(\mathsf{s}) \leq \#_{\mathsf{x}}(\mathsf{t})$.

Folgender Satz sichert, daß die definierte Relation $<_{\mathsf{TO}}$ eine Ordnung ist.

Satz 5.14 ($<_{\mathsf{TO}}$ ist eine partielle Ordnung) *Die definierte Relation* $<_{\mathsf{TO}}$ *ist eine partielle Ordnung.*

Beweis 5.14 *Folgt direkt aus der Tatsache, daß* $\ll_{\mathsf{F}}$ *eine partielle Ordnung ist.* ■

5.4.1 Terminierung

In diesem Abschnitt wird sichergestellt, daß für jede Menge Φ von Strukturgleichungen das entsprechende "Strukturterm"-Ersetzungssystem E_Φ terminierend ist; d. h. es darf keine unendliche Anzahl von Strukturtermen $\mathsf{s}_\mathsf{i}, \mathsf{i} = 0 \ldots \infty$ geben, für die gilt:

$$\mathsf{s}_0 \vdash_{\mathsf{E}_\Phi} \mathsf{s}_1 \vdash_{\mathsf{E}_\Phi} \ldots$$

Definition 5.11 (Terminierung eines Ersetzungssystems) *Ein gerichtetes Ersetzungssystem* E *im Sinne von Definition 3.16 ist genau dann* terminierend, *wenn es keine unendliche Anzahl von Strukturtermen* $\mathsf{s}_\mathsf{i}, \mathsf{i} = 0 \ldots \infty$ *gibt, für die gilt:*

$$\mathsf{s}_0 \vdash_{\mathsf{E}_\Phi} \mathsf{s}_1 \vdash_{\mathsf{E}_\Phi} \ldots$$

Um dieses sicher zu stellen, wird zunächst bewiesen, daß die partielle Ordnung $<_{\mathsf{TO}}$ noethersch ist. Anschließend wird gezeigt, daß das Ergebnis der Anwendung einer gerichteten Gleichung aus E_Φ auf einen Strukturterm bezüglich der partiellen Ordnung $<_{\mathsf{TO}}$ kleiner ist als der ursprüngliche Strukturterm.

Aus diesen beiden Eigenschaften folgt dann die Terminierung des Ersetzungssystems.

Satz 5.15 *Die partielle Ordnung* $<_{\mathsf{TO}}$ *ist noethersch.*

Beweis 5.15 *Das Satz folgt direkt aus Satz 3.2.* ■

Im folgenden wird gezeigt, daß die Ordnung $<_{\mathsf{TO}}$ eine Simplifikationsordnung ist. Dies bedeutet, daß sie substitutionsstabil und subtermstabil sein muß. Bei der Substitutionsstabilität kann allerdings eine Einschränkung gemacht werden, da für eine gerichtete Gleichung $\mathsf{u} \to \mathsf{v}$ bei der Anwendung auf einen Term nur die Variablen instantiiert werden, die in u vorkommen. Somit muß die Substitutionsstabilität nur für solche Substitutionen gezeigt werden, die nur Variablen des größeren Terms instantiieren.

Satz 5.16 *Gegeben seien zwei Terme* s *und* t *zu einer Signatur* Σ *und einer Variablenmenge* $\mathcal{V}$. *Falls* $\mathsf{v}<_{\mathsf{TO}}\mathsf{u}$ *gilt, dann gilt für alle Substitutionen* σ *mit* $\mathsf{Dom}(\sigma) \subseteq \mathsf{Var}(\mathsf{u})$: $\sigma(\mathsf{v})<_{\mathsf{TO}}\sigma(\mathsf{u})$.

Beweis 5.16 *Es gelten*

$$\mathsf{Fu}(\sigma(\mathsf{u})) = \mathsf{Fu}(\mathsf{u}) \cup_{\mathsf{M}}^{\mathsf{x}\in\mathsf{Dom}(\sigma)} \underbrace{\mathsf{Fu}(\sigma(\mathsf{x})) \cup_{\mathsf{M}} \ldots \cup_{\mathsf{M}} \mathsf{Fu}(\sigma(\mathsf{x}))}_{\#_{\mathsf{x}}(\mathsf{u})\text{- mal}}$$

$$\mathsf{Fu}(\sigma(\mathsf{v})) = \mathsf{Fu}(\mathsf{v}) \cup_{\mathsf{M}}^{\mathsf{x}\in\mathsf{Dom}(\sigma)} \underbrace{\mathsf{Fu}(\sigma(\mathsf{x})) \cup_{\mathsf{M}} \ldots \cup_{\mathsf{M}} \mathsf{Fu}(\sigma(\mathsf{x}))}_{\#_{\mathsf{x}}(\mathsf{v})\text{- mal}}$$

Weiterhin gelten $\mathsf{Fu}(\sigma(\mathsf{v})) \setminus_{\mathsf{M}} \mathsf{Fu}(\sigma(\mathsf{u})) = \mathsf{Fu}(\mathsf{v}) \setminus_{\mathsf{M}} \mathsf{Fu}(\mathsf{u})$, *da für alle* $\mathsf{x} \in \mathsf{Dom}(\sigma)$ $\#_{\mathsf{x}}(\mathsf{u}) \geq \#_{\mathsf{x}}(\mathsf{v})$ *gilt, und*

$$\mathsf{Fu}(\sigma(\mathsf{u})) \setminus_{\mathsf{M}} \mathsf{Fu}(\sigma(\mathsf{v})) = \mathsf{Fu}(\mathsf{u}) \setminus_{\mathsf{M}} \mathsf{Fu}(\mathsf{v}) \cup_{\mathsf{M}}^{\mathsf{x}\in\mathsf{Dom}(\sigma)} \underbrace{\mathsf{Fu}(\sigma(\mathsf{x})) \cup_{\mathsf{M}} \ldots \cup_{\mathsf{M}} \mathsf{Fu}(\sigma(\mathsf{x}))}_{(\#_{\mathsf{x}}(\mathsf{u})-\#_{\mathsf{x}}(\mathsf{v}))\text{- mal}}$$

Aus $\mathsf{Fu}(\mathsf{v})\ll_{\mathsf{F}}\mathsf{Fu}(\mathsf{v})$ *und der Definition von* $\ll_{\mathsf{F}}$ *ist bekannt, daß für alle* $\mathsf{f} \in \mathsf{Fu}(\mathsf{v}) \setminus_{\mathsf{M}} \mathsf{Fu}(\mathsf{u})$ *ein* $\mathsf{g} \in \mathsf{Fu}(\mathsf{u}) \setminus_{\mathsf{M}} \mathsf{Fu}(\mathsf{v})$ *existiert, so daß* $\mathsf{f}<_{\mathsf{F}}\mathsf{g}$ *gilt. Da für alle* $\mathsf{x} \in \mathsf{Dom}(\sigma)$ $\#_{\mathsf{x}}(\mathsf{v}) \leq \#_{\mathsf{x}}(\mathsf{u})$ *gilt, ist dieses* g *auch in* $\mathsf{Fu}(\sigma(\mathsf{u})) \setminus_{\mathsf{M}} \mathsf{Fu}(\sigma(\mathsf{v}))$ *enthalten. Somit gilt* $\mathsf{Fu}(\sigma(\mathsf{v}))\ll_{\mathsf{F}}\mathsf{Fu}(\sigma(\mathsf{u}))$.

Da gilt: $\forall \mathsf{x} \in \mathcal{V}\ \#_{\mathsf{x}}(\mathsf{v}) \leq \#_{\mathsf{x}}(\mathsf{u})$, *gilt auch:*$\forall \mathsf{x} \in \mathcal{V}\ \#_{\mathsf{x}}(\sigma(\mathsf{v})) \leq \#_{(}\sigma(\mathsf{u}))$.

Also gilt: $\sigma(\mathsf{v})\ll_{\mathsf{F}}\sigma(\mathsf{u})$.

■

Satz 5.17 *Seien ein "Struktur"-Termersetzungssystem* E *und eine Strukturterm* s *gegeben. Sei* $\mathsf{u} \to \mathsf{v}$ *eine gerichtete Gleichung aus* E. *Falls* $\mathsf{u} \to \mathsf{v}$ *an einer Termposition* π *unter einer Substitution* σ *auf* s *anwendbar ist, dann gilt:*

$$\mathsf{s}>_{\mathsf{TO}}\mathsf{s}_{|\pi,\leftarrow\sigma(\mathsf{v})}$$

Beweis 5.17 *Der Beweis wird per Induktionüber die Länge* n *von* π *geführt.*

$\mathsf{n} = 0$ **:** *D. h.* $\mathsf{s} = \sigma(\mathsf{u})$ *. Also ist* $\mathsf{s}_{|[],\leftarrow\sigma(\mathsf{v})}$ *und per Definition des gerichteten Ersetzungssystems und der Substitutionsstabilität von* $<_{\mathsf{TO}}$ *gilt:* $\sigma(\mathsf{v}) <_{\mathsf{TO}} \sigma(\mathsf{u})$.

$\mathsf{n} \rightarrow \mathsf{n}+1$ **:** *Sei* $\pi = [\mathsf{i}, \pi']$ *. Dann ist* s *der Form* $\mathsf{f}(\mathsf{t}_1, \ldots, \mathsf{t}_\mathsf{n})$*, und per Induktionsannahme gilt:*

$$\mathsf{t}_{\mathsf{i}|\pi',\leftarrow\sigma(\mathsf{v})} <_{\mathsf{TO}} \mathsf{t}_\mathsf{i}$$

Sei $\mathsf{t}' = \mathsf{t}_{\mathsf{i}|\pi',\leftarrow\sigma(\mathsf{v})}$*. Wiederrum per Induktionsannahme gilt dann auch*

$$\mathsf{t}_{|[\mathsf{i}],\leftarrow\mathsf{t}'} <_{\mathsf{TO}} \mathsf{t}$$

und somit: $\mathsf{t}_{|\pi,\leftarrow\sigma(\mathsf{v})} <_{\mathsf{TO}} \mathsf{t}$.

■

Kapitel 6

Kommentierte Beispiele

In diesem Abschnitt wird die Funktionsweise des Gesamtverfahrens an vier ausgewählten Beispielen demonstriert. An diesen Beispielen werden die wichtigsten Heuristiken aus Kapitel 3 sowie deren Zusammenspiel demonstriert. Dabei werden die Beispiele in der Form präsentiert, wie sie von den implementierten Heuristiken gefunden wurden.
Die ersten drei Beispiele sind Gleichheitsprobleme über Listen von natürlichen Zahlen. Das vierte Beispiel ist ein etwas komplexeres Gleichheitsproblem, das im Rahmen der Verifikation eines Parser aufgetreten ist. Für alle Beispiele werden zunächst die verwendeten Signaturen und die Gleichungen angegeben, die die Semantik der Funktionssymbole aus den einzelnen Signaturen festlegen.

Konvention: Die Schreibweise einer Gleichung unterteilt sich in drei Teile:

$$(\mathsf{Typ}) \qquad \mathsf{s} \equiv \mathsf{t} \qquad (\mathsf{Nr.})$$

Am linken Rand wird der Typ der Gleichung angegeben. Es gibt insgesamt vier Typen von Gleichungen: Gleichungen zu einer Signatur Σ werden mit (Ax) bezeichnet. Abstraktionen einer solchen Gleichungen werden mit (Ax') bezeichnet. Die Theoreme zur Signatur Σ werden mit (Th) und deren Abstraktionen mit (Th') gekennzeichnet. In der Mitte steht die entsprechende (abstrakte) Gleichung und ganz rechts eine eindeutige Nummer für diese Gleichung, mit Hilfe derer auf diese Gleichung referenziert werden kann.

6.1 Listen

In diesem Abschnitt werden einige Beispiele über Listen von natürlichen Zahlen aufgeführt. Natürliche Zahlen seien durch eine sortierte Signatur

$$\Sigma_{\mathsf{Nat}} = (\emptyset, \{\mathsf{0} :\rightarrow \mathsf{Nat}\ , \mathsf{s} : \mathsf{Nat}\ \rightarrow \mathsf{Nat}\ \}, \{\mathsf{Nat}\ \})$$

und eine Menge von Axiomen $\mathsf{Ax}_{\mathsf{Nat}}$ spezifiziert. $\mathsf{Ax}_{\mathsf{Nat}}$ ist

$$(\mathsf{Ax}) \qquad \forall \mathsf{n} : \mathsf{Nat}\ \mathsf{s}(\mathsf{n}) \not\equiv 0 \tag{6.1}$$

Listen über natürlichen Zahlen werden dabei wie folgt spezifiziert: Gegeben ist eine sortierte Signatur

$$\begin{aligned}\Sigma_{\mathsf{List}} = (\{ & \quad \mathsf{Member} : \mathsf{Nat} \times \mathsf{List}\ \},\\ \{ & \quad \mathsf{Nil} :\to \mathsf{List}\ , \mathsf{cons} : \mathsf{Nat} \times \mathsf{List} \to \mathsf{List}\ ,\\ & \quad \mathsf{car} : \mathsf{List} \to \mathsf{Nat}\ , \mathsf{cdr} : \mathsf{List} \to \mathsf{List}\ ,\\ & \quad \mathsf{app} : \mathsf{List} \times \mathsf{List} \to \mathsf{List}\ \},\\ \{ & \quad \mathsf{List}\ \}) \cup \Sigma_{\mathsf{Nat}}\end{aligned}$$

Spezifiziert wird der Datentyp List durch die folgende Menge von Axiomen $\mathsf{Ax}_{\mathsf{List}}$:

$$(\mathsf{Ax}) \qquad \forall \mathsf{n} : \mathsf{Nat}\ \forall \mathsf{l} : \mathsf{List}\ \mathsf{cons}(\mathsf{n}, \mathsf{l}) \not\equiv \mathsf{Nil} \tag{6.2}$$

$$(\mathsf{Ax}) \qquad \mathsf{cdr}(\mathsf{Nil}) \equiv \mathsf{Nil} \tag{6.3}$$

$$(\mathsf{Ax}) \qquad \forall \mathsf{n} : \mathsf{Nat}\ \forall : \mathsf{List}\ \mathsf{cdr}(\mathsf{cons}(\mathsf{n}, \mathsf{l})) \equiv \mathsf{l} \tag{6.4}$$

$$(\mathsf{Ax}) \qquad \mathsf{car}(\mathsf{Nil}) \equiv 0 \tag{6.5}$$

$$(\mathsf{Ax}) \qquad \forall \mathsf{n} : \mathsf{Nat}\ \forall \mathsf{l} : \mathsf{List}\ \mathsf{car}(\mathsf{cons}(\mathsf{n}, \mathsf{l})) \equiv \mathsf{n} \tag{6.6}$$

$$(\mathsf{Ax}) \qquad \forall \mathsf{l} : \mathsf{Nat}\ \mathsf{app}(\mathsf{Nil}, \mathsf{l}) \equiv \mathsf{l} \tag{6.7}$$

$$(\mathsf{Ax}) \qquad \forall \mathsf{l}_1, \mathsf{l}_2 : \mathsf{List}\ \forall \mathsf{n} : \mathsf{Nat}\ \mathsf{app}(\mathsf{cons}(\mathsf{n}, \mathsf{l}_1), \mathsf{l}_2) \equiv \mathsf{cons}(\mathsf{n}, \mathsf{app}(\mathsf{l}_1, \mathsf{l}_2)) \tag{6.8}$$

$$(\mathsf{Ax}) \qquad \forall \mathsf{n} : \mathsf{Nat}\ \neg \mathsf{Member}(\mathsf{n}, \mathsf{Nil})$$

$$(\mathsf{Ax}) \qquad \forall \mathsf{n}, \mathsf{m} : \mathsf{Nat}\ \forall \mathsf{l} : \mathsf{List}\ (\mathsf{Member}(\mathsf{n}, \mathsf{l}) \vee \mathsf{n} \equiv \mathsf{m}) \Rightarrow \mathsf{Member}(\mathsf{n}, \mathsf{cons}(\mathsf{m}, \mathsf{l}))$$

Gemäß der Definition der abstrakten Gleichungen aus Abschnitt 2.1 lassen sich diese Gleichungen wie folgt abstrahieren: Jede Gleichung wird bezüglich der Menge der gemeinsamen Variablen der linken und rechten Seiten dieser Gleichung abstrahiert. Quantifizierte Variablen, die keine gemeinsamen Variablen sind, fallen in der abstrakten Gleichung weg. Damit erhält man die folgenden abstrakten Gleichungen zur Signatur $\Sigma_{\mathsf{List}} \cup \{@_{\mathsf{Nat}} :\to \mathsf{Nat}\ , @_{\mathsf{List}} :\to \mathsf{List}\ \}$:

$$(\mathsf{Ax}') \qquad @_{\mathsf{List}} \not\equiv @_{\mathsf{List}} \tag{6.9}$$

An dieser abstrakten Gleichung läßt sich gut das Phänomen beobachten, daß die Abstraktion einer erfüllbaren Formelmenge nicht notwendigerweise auch eine erfüllbare Formelmenge ist (siehe Satz 2.2 auf Seite 19).

$$(\mathsf{Ax}') \qquad @ \equiv @ \qquad (6.10)$$

$$(\mathsf{Ax}') \qquad \forall \mathsf{l} : \mathsf{List}\ \mathsf{cdr}(\mathsf{cons}(@_{\mathsf{List}}, \mathsf{l})) \equiv \mathsf{l} \qquad (6.11)$$

$$(\mathsf{Ax}') \qquad @_{\mathsf{Nat}} \equiv @_{\mathsf{Nat}} \qquad (6.12)$$

$$(\mathsf{Ax}') \qquad \forall \mathsf{n} : \mathsf{Nat}\ \mathsf{car}(\mathsf{cons}(\mathsf{n}, @_{\mathsf{SLIST}})) \equiv \mathsf{n} \qquad (6.13)$$

$$(\mathsf{Ax}') \qquad \forall \mathsf{l} : \mathsf{List}\ \mathsf{app}(@_{\mathsf{List}}, \mathsf{l}) \equiv \mathsf{l} \qquad (6.14)$$

$$(\mathsf{Ax}') \qquad \forall \mathsf{n} : \mathsf{Nat}\ \forall \mathsf{l}_1, \mathsf{l}_2 : \mathsf{List}\ \mathsf{app}(\mathsf{cons}(\mathsf{n}, \mathsf{l}_1), \mathsf{l}_2) \equiv \mathsf{cons}(\mathsf{n}, \mathsf{app}(\mathsf{l}_1, \mathsf{l}_2)) \qquad (6.15)$$

6.1.1 Ein einfaches Beispiel

Das in diesem Abschnitt besprochene Beispiel taucht als Unterproblem im Beweis des folgenden Theorems auf:

$$(\mathsf{Th}) \qquad \forall \mathsf{n} : \mathsf{Nat}\ \forall \mathsf{l} : \mathsf{List}\ \mathsf{Member}(\mathsf{n}, \mathsf{l}) \Rightarrow \exists \mathsf{l}_1, \mathsf{l}_2 : \mathsf{List}\ \mathsf{l} \equiv \mathsf{app}(\mathsf{l}_1, \mathsf{cons}(\mathsf{n}, \mathsf{l}_2,)) \qquad (6.16)$$

Der Beweis dieses Theorems wird per Induktion über l geführt. Die Basisfälle sind $\mathsf{l} \equiv \mathsf{Nil}$ und $\mathsf{l} \equiv \mathsf{cons}(\mathsf{n}, \mathsf{cdr}(\mathsf{l}))$. Der Induktionfall ist $\mathsf{l} \equiv \mathsf{cons}(\mathsf{car}(\mathsf{l}), \mathsf{cdr}(\mathsf{l})) \wedge \mathsf{n} \not\equiv \mathsf{car}(\mathsf{l})$. Betrachtet wird im folgendem der zweite Basisfall $\mathsf{l} \equiv \mathsf{cons}(\mathsf{n}, \mathsf{cdr}(\mathsf{l}))$. Hier erhält man das Gleichheitsproblem

$$(\mathsf{Th}) \qquad \forall \mathsf{n} : \mathsf{Nat}\ \forall \mathsf{l} : \mathsf{List}\ \exists \mathsf{l}_1, \mathsf{l}_2 : \mathsf{List}\ \mathsf{cons}(\mathsf{n}, \mathsf{cdr}(\mathsf{l})) \equiv \mathsf{app}(\mathsf{l}_1, \mathsf{cons}(\mathsf{n}, \mathsf{l}_2)) \qquad (6.17)$$

Die Menge der gemeinsamen Unterterme (siehe Definition 3.2.1 auf Seite 32) dieses Gleichheitsproblems ist $\{\mathsf{n}\}$. Die Abstraktion dieses Gleichheitsproblems bezüglich dieser gemeinsamen Unterterme ist die abstrakte Gleichung

$$(\mathsf{Th}') \qquad \forall \mathsf{n} : \mathsf{Nat}\ \mathsf{cons}(\mathsf{n}, @_{\mathsf{List}}) \equiv \mathsf{app}(@_{\mathsf{List}}, \mathsf{cons}(\mathsf{n}, @_{\mathsf{List}})) \qquad (6.18)$$

Die Abstraktionsinformation (siehe Abschnitt 3.2.1 auf Seite 32) für dieses abstrakte Gleichheitsproblem bezüglich des gemeinsamen Untertermes n ist:

	$\mathsf{cons}(\mathsf{n}, @_{\mathsf{List}})$	$\mathsf{app}(@_{\mathsf{List}}, \mathsf{cons}(\mathsf{n}, @_{\mathsf{List}}))$	Skelett
n	$[(\mathsf{cons}, 1)]$	$[(\mathsf{app}, 2)(\mathsf{cons}, 1)]$	$[(\mathsf{cons}, 1)]$

Abstraktionsinformation von (6.18)

In der dritten Spalte steht das betrachtete Skelett. Aus der Abstraktionsinformation läßt sich ablesen, daß der Kontext (siehe Definition 3.1.4 auf Seite 32) in der rechten Liste eliminiert werden muß, um das abstrakte Gleichheitsproblem zu lösen. Dazu wird in der Datenbasis nach einer abstrakten Gleichung gesucht, die in diesem Sinne *kontexteliminierend* ist. Dazu werden die Abstraktionsinformationen der in der Datenbasis vorhandenen abstrakten Gleichungen analysiert. Die Abstraktionsinformation der abstrakten Gleichung (6.14) ist

	$\mathsf{app}(@_{\mathsf{List}}, \mathsf{l})$	l	Skelett
l	$[(\mathsf{app}, 2)]$	$[]$	$[]$

Abstraktionsinformation von (6.14)

Aus der Abstraktionsinformation der Gleichung (6.14) läßt sich ablesen, daß diese abstrakte Gleichung, wenn von links nach rechts angewendet, den Kontext $[(\mathsf{app}, 2)]$ eliminiert. Somit wird diese abstrakte Gleichung ausgewählt und an der Top-Level Termposition des rechten abstrakten Termes des Gleichheitsproblems (6.18) angewendet. Daraus resultiert das triviale abstrakte Gleichheitsproblem

$$(\mathsf{Th}') \qquad \forall \mathsf{n} : \mathsf{Nat}\ \mathsf{cons}(\mathsf{n}, @_{\mathsf{List}}) \equiv \mathsf{cons}(\mathsf{n}, @_{\mathsf{List}}) \tag{6.19}$$

Das neue abstrakte Gleichheitsproblem (6.19) ist bezüglich der Ordnung $<_{\mathsf{GO}}$ kleiner als das ursprüngliche abstrakte Gleichheitsproblem (6.18), da die Kontexte abgenommen haben, die Differenz in der Anzahl der Vorkommen von n aber gleich $\mathsf{0}$ geblieben ist.

Der aus dem abstrakten Beweis resultierende Plan für das konkrete Gleichheitsproblem ist eine Gleichung, deren Abstraktion identisch ist mit der abstrakten Gleichung (6.14), an der Top-Level Position des rechten Termes des konkreten Gleichheitsproblems (6.16) von links nach rechts anzuwenden. In diesem Beispiel trifft dies nur auf die Gleichung (6.7) zu, was aber im allgemeinen nicht so sein muß.

Um die Gleichung $\mathsf{app}(\mathsf{Nil}, \mathsf{l})$ auf den rechten Term $\mathsf{app}(\mathsf{l}_1, \mathsf{cons}(\mathsf{n}, \mathsf{l}_2))$ anzuwenden, muß die Existenz-quantifizierte Variable l_1 mit Nil instantiiert werden, ein Problem, welches auf der abstrakten Ebene nicht aufgetaucht ist. Daraus resultiert das neue Gleichheitsproblem

$$(\mathsf{Th}) \qquad \mathsf{cons}(\mathsf{n}, \mathsf{cdr}(\mathsf{l})) \equiv \mathsf{cons}(\mathsf{n}, \mathsf{l}_2) \tag{6.20}$$

Die Abstraktion dieses Gleichheitsproblems entspricht gerade der abstrakten Gleichung (6.18). Das dies gilt, ist schon dadurch sichergestellt, daß die vorgenommene Veränderung der Gleichung (6.7) (die Instantiierung von l_1 durch Nil) den Kriterien aus Abschnitt 4.1 auf Seite 59 entspricht.

Das Gleichheitsproblem (6.20) läßt sich nun einfach lösen, indem die existenzquantifizierte Variable l_2 mit $\mathsf{cdr}(\mathsf{l})$ instantiiert wird, womit das Problem gelöst ist.

6.1.2 Komplexeres Beispiel

In diesem Abschnitt wird die Signatur Σ_{List} zum einen um eine Funktion $+$ auf den natürlichen Zahlen erweitert, die die Summe zweier natürlicher Zahlen "berechnet". Zum anderen wird sie um die Funktion length erweitert, die die Länge einer Liste über natürlichen Zahlen "berechnet", d. h. daß folgende Signatur betrachtet wird:

$$\Sigma_{\mathsf{List}} \cup \{+ : \mathsf{Nat} \times \mathsf{Nat} \rightarrow \mathsf{Nat}, \mathsf{length} : \mathsf{List} \rightarrow \mathsf{Nat}\}$$

Die Menge der Axiome $\mathsf{Ax}_{\mathsf{List}}$ wird um die Definitionsgleichungen von $+$ und length erweitert. Dies sind

$$(\mathsf{Ax}) \qquad \forall \mathsf{n} : \mathsf{Nat}\ 0 + \mathsf{n} \equiv \mathsf{n} \qquad (6.21)$$

$$(\mathsf{Ax}) \qquad \forall \mathsf{n}_1, \mathsf{n}_2 : \mathsf{Nat}\ \mathsf{s}(\mathsf{n}_1) + \mathsf{n}_2 \equiv \mathsf{s}(\mathsf{n}_1 + \mathsf{n}_2) \qquad (6.22)$$

$$(\mathsf{Ax}) \qquad \mathsf{length}(\mathsf{Nil}) \equiv 0 \qquad (6.23)$$

$$(\mathsf{Ax}) \qquad \forall \mathsf{n} : \mathsf{Nat}\ \forall \mathsf{l} : \mathsf{List}\ \mathsf{length}(\mathsf{cons}(\mathsf{n}, \mathsf{l})) \equiv \mathsf{s}(\mathsf{length}(\mathsf{l})) \qquad (6.24)$$

Die Abstraktionen dieser Gleichungen bezüglich der gemeinsamen Variablen sind:

$$(\mathsf{Ax}') \qquad \forall \mathsf{n} : \mathsf{Nat}\ @_{\mathsf{Nat}} + \mathsf{n} \equiv \mathsf{n} \qquad (6.25)$$

$$(\mathsf{Ax}') \qquad \forall \mathsf{n}_1, \mathsf{n}_2 : \mathsf{Nat}\ \mathsf{s}(\mathsf{n}_1) + \mathsf{n}_2 \equiv \mathsf{s}(\mathsf{n}_1 + \mathsf{n}_2) \qquad (6.26)$$

$$(\mathsf{Ax}') \qquad @_{\mathsf{Nat}} \equiv @_{\mathsf{Nat}} \qquad (6.27)$$

$$(\mathsf{Ax}') \qquad \forall \mathsf{l} : \mathsf{List}\ \mathsf{length}(\mathsf{cons}(@_{\mathsf{Nat}}, \mathsf{l})) \equiv \mathsf{s}(\mathsf{length}(\mathsf{l})) \qquad (6.28)$$

Zudem werden noch zwei weitere Gleichungen in die Axiomenmenge aufgenommen. Dies ist zum einen die Gleichung, die die Kommutativität von $+$ ausdrückt und zum anderen eine Gleichung, die das Verhalten der Funktion length auf der Konkatenation zweier Listen beschreibt:

$$(\mathsf{Ax}) \qquad \forall \mathsf{n}_1, \mathsf{n}_2 : \mathsf{Nat}\ \mathsf{n}_1 + \mathsf{n}_2 \equiv \mathsf{n}_2 + \mathsf{n}_1 \qquad (6.29)$$

$$(\mathsf{Ax}) \qquad \forall \mathsf{l}_1, \mathsf{l}_2 : \mathsf{List}\ \mathsf{length}(\mathsf{app}(\mathsf{l}_1, \mathsf{l}_2)) \equiv \mathsf{length}(\mathsf{l}_1) + \mathsf{length}(\mathsf{l}_2) \qquad (6.30)$$

Die Abstraktionen dieser beiden Gleichungen bezüglich der gemeinsamen Variablen sind wieder die ursprünglichen Gleichungen. Das zu beweisende Theorem ist

$$(\mathsf{Th}) \qquad \forall \mathsf{l}_1, \mathsf{l}_2 : \mathsf{List}\ \mathsf{length}(\mathsf{app}(\mathsf{l}_1, \mathsf{l}_2)) \equiv \mathsf{length}(\mathsf{app}(\mathsf{l}_2, \mathsf{l}_1)) \qquad (6.31)$$

Die Abstraktion des Gleichheitsproblems bezüglich der gemeinsamen Variablen l_1 und l_2 ist wieder das ursprüngliche Gleichheitsproblem. Die Abstraktionsinformation für dieses Gleichheitsproblem ist:

	$\mathsf{length}(\mathsf{app}(\mathsf{l}_1, \mathsf{l}_2))$	$\mathsf{length}(\mathsf{app}(\mathsf{l}_2, \mathsf{l}_1))$	Skelett
l_1	$[(\mathsf{length}, 1)(\mathsf{app}, 1)]$	$[(\mathsf{length}, 1)(\mathsf{app}, 2)]$	$[(\mathsf{length}, 1)]$
l_2	$[(\mathsf{length}, 1)(\mathsf{app}, 2)]$	$[(\mathsf{length}, 1)(\mathsf{app}, 1)]$	$[(\mathsf{length}, 1)]$

Abstraktionsinformation von (6.31)

Aus dieser Abstraktionsinformation läßt sich die Information gewinnen, daß die Kontexte, $(\mathsf{app}, 1)$ und $(\mathsf{app}, 2)$ in den jeweiligen Funktionssymbollisten innerhalb der jeweiligen betrachteten Skelette stehen. Die skelettorientierten Heuristiken aus Abschnitt 3.2.4.3 versuchen zunächst diese Kontexte innerhalb der Listen nach links zu verschieben. Dabei spielt es keine Rolle, ob es derselbe Kontext ist, der weiter links in der Liste auftaucht, oder ob ein beliebiger Kontext weiter links auftaucht. Wichtig ist nur, daß das Skelett erhalten bleibt. Dafür wird in der Datenbasis nach einer abstrakten Gleichung gesucht, aus deren Abstraktionsinformation abzulesen ist, daß sie einerseits das Skelett erhält und, andererseits, den Kontext nach links in den Funktionssymbollisten verschiebt. Dies entspricht auf der Term Seite einer Verschiebung des Kontextes nach oben relativ zu dem betrachteten Skelett. Die einzige (abstrakte) Gleichung, die sowohl das Skelett $[(\mathsf{length}, 1)]$ erhält, als auch den Kontext verschiebt, ist die Gleichung (6.30):

$$\forall \mathsf{l}_1, \mathsf{l}_2 : \mathsf{List}\ \mathsf{length}(\mathsf{app}(\mathsf{l}_1, \mathsf{l}_2)) \equiv \mathsf{length}(\mathsf{l}_1) + \mathsf{length}(\mathsf{l}_2).$$

Die Abstraktionsinformation dieser Gleichung bezüglich der gemeinsamen Variablen $\{\mathsf{l}_1, \mathsf{l}_2\}$ ist

	$\mathsf{length}(\mathsf{app}(\mathsf{l}_1, \mathsf{l}_2))$	$\mathsf{length}(\mathsf{l}_1) + \mathsf{length}(\mathsf{l}_2)$	Skelett
l_1	$[(\mathsf{length}, 1), (\mathsf{app}, 1)]$	$[(+, 1), (\mathsf{length}, 1)]$	$[(\mathsf{length}, 1)]$
l_2	$[(\mathsf{length}, 1), (\mathsf{app}, 2)]$	$[(+, 2), (\mathsf{length}, 1)]$	$[(\mathsf{length}, 1)]$

Abstraktionsinformation von (6.30)

Aus der Abstraktionsinformation der Gleichung (6.30) läßt sich ablesen, daß, wird diese Gleichung von links nach rechts angewendet, die Kontexte, $(\mathsf{app}, 1)$ und $(\mathsf{app}, 2)$ aus den respektiven Funktionssymbollisten des linken Termes eliminiert werden und durch relativ zum Skelett weiter oben vorkommende Kontexte $(+, 1)$ und $(+, 2)$ respektive ersetzt werden. Dies entspricht im Sinne von Abschnitt 3.2.4.3 einer Verschiebung der Kontexte nach oben. Die Anwendung der Gleichung (6.30) auf den linken Term des abstrakten Theorems (6.31) liefert das neue Theorem

$$(\mathsf{Th}') \qquad \forall \mathsf{l}_1, \mathsf{l}_2 : \mathsf{List}\ \mathsf{length}(\mathsf{l}_1) + \mathsf{length}(\mathsf{l}_2) \equiv \mathsf{length}(\mathsf{app}(\mathsf{l}_2, \mathsf{l}_1)) \qquad (6.32)$$

Das neue Gleichheitsproblem (6.32) ist bezüglich der Ordnung $<_{\mathsf{GO}}$ kleiner als ursprüngliche (6.31), da die Kontexte abgenommen haben und die Differenz in der Anzahl der Vorkommen von l_1 und l_2 gleich geblieben ist. Es gilt also:

$$(\mathsf{length}(\mathsf{l}_1)+\mathsf{length}(\mathsf{l}_2), \mathsf{length}(\mathsf{app}(\mathsf{l}_2,\mathsf{l}_1)), \{\mathsf{l}_1,\mathsf{l}_2\}) \\ <_{GO} (\mathsf{length}(\mathsf{app}(\mathsf{l}_1,\mathsf{l}_2)), \mathsf{length}(\mathsf{app}(\mathsf{l}_2,\mathsf{l}_1)), \{\mathsf{l}_1,\mathsf{l}_2\})$$

Die Abstraktionsinformation des neuen Gleichheitsproblems ist

	$\mathsf{length}(\mathsf{l}_1)+\mathsf{length}(\mathsf{l}_2)$	$\mathsf{length}(\mathsf{app}(\mathsf{l}_2,\mathsf{l}_1))$	Skelett
l_1	$[(+,1),(\mathsf{length},1)]$	$[(\mathsf{length},1),(\mathsf{app},2)]$	$[(\mathsf{length},1)]$
l_2	$[(+,2),(\mathsf{length},1)]$	$[(\mathsf{length},1),(\mathsf{app},1)]$	$[(\mathsf{length},1)]$

Abstraktionsinformation von (6.32)

Die in dieser Abstraktionsinformation enthaltenen Informationen sind zum einen, daß die Kontexte in dem linken Term schon ganz außen stehen und zum anderen, daß die Kontexte in dem rechten Term noch innerhalb des betrachteten Skelettes stehen. Nun gibt es zwei Möglichkeiten: Einerseits kann versucht werden, die außen stehenden Kontexte des linken Termes zu eliminieren oder die Kontexte im rechten Term nach außen zu schieben. Ersteres schlägt fehl, da keine adäquate Gleichung gefunden werden kann. Letzteres kann mit Hilfe der selben Gleichung (6.30) wie im vorherigen Schritt erledigt werden. Dies Anwendung der Gleichung (6.30) auf den rechten Term führt zu folgendem Gleichheitsproblem:

$$(\mathsf{Th}') \qquad \forall \mathsf{l}_1,\mathsf{l}_2 : \mathsf{List}\ \mathsf{length}(\mathsf{l}_1)+\mathsf{length}(\mathsf{l}_2) \equiv \mathsf{length}(\mathsf{l}_2)+\mathsf{length}(\mathsf{l}_1) \tag{6.33}$$

Auch in diesem Schritt gilt, daß das neue Gleichheitsproblem (6.33) bezüglich der Ordnung $<_{GO}$ kleiner ist als das usprüngliche (6.32), da außer einer Verringerung des Kontextes sich nichts verändert hat.

Die Abstraktionsinformation für das neue Gleichheitsproblem (6.33) ist

	$\mathsf{length}(\mathsf{l}_1)+\mathsf{length}(\mathsf{l}_2)$	$\mathsf{length}(\mathsf{l}_2)+\mathsf{length}(\mathsf{l}_1)$	Skelett
l_1	$[(+,1),(\mathsf{length},1)]$	$[(+,2),(\mathsf{length},1)]$	$[(\mathsf{length},1)]$
l_2	$[(+,2),(\mathsf{length},1)]$	$[(+,1),(\mathsf{length},1)]$	$[(\mathsf{length},1)]$

Abstraktionsinformation von (6.33)

Aus dieser Abstraktionsinformation ist ersichtlich, daß nun alle Kontexte relativ zu den betrachteten Skeletten außen stehen. Der Versuch, diese Kontexte zu eliminieren, scheitert wie schon im vorherigen Schritt daran, daß keine adäquate abstrakte Gleichung hierfür existiert. Hier kommt jetzt die spezielle Heuristik aus Abschnitt 3.2.4.4 zum Einsatz. Diese sucht sich einen Kontext auf einer Seite aus und versucht diesen auf der rechten Seite zu erzeugen, ohne dabei das bestehende Skelett zu verletzen. In diesem Fall würde sie zum Beispiel den Kontext $(+,1)$ aus der Funktionssymbolliste $[(+,1),(\mathsf{length},1)]$ aussuchen und versuchen diesen in der zugeordneten Funktionssymbolliste $[(+,2),(\mathsf{length},1)]$ zu erzeugen. Hierfür wird in der Datenbasis nach einer Gleichung gesucht, die den Kontext $(+,1)$ erzeugt und gleichzeitig das Skelett unverändert läßt. Die einzige hierfür in Frage kommende Gleichung ist die Kommutativitätsgleichung von + . Die Abstraktionsinformation dieser Gleichung (6.29) bezüglich der gemeinsamen Variablen n_1 und n_2 ist:

	$n_1 + n_2$	$n_2 + n_1$	Skelett
n_1	$[(+, 1)]$	$[(+, 2)]$	$[]$
n_2	$[(+, 2)]$	$[(+, 1)]$	$[]$

Abstraktionsinformation von (6.29)

Aus dieser Abstraktionsinformation ist ersichtlich, daß, wird die Gleichung von links nach rechts angewendet, in der Funktionssymbolliste zu dem zweiten Argument von + das Paar $(+, 1)$ erzeugt wird. Gleichzeitig läßt das Skelett des Gleichheitsproblems (6.33) unverändert. Durch Anwendung der (abstrakten) Gleichung (6.29) von links nach rechts auf den rechten Term des (abstrakten) Gleichheitsproblems (6.33) erhält man das triviale Gleichheitsproblem

$$(\mathsf{Th}') \qquad \forall \mathsf{n}_1, \mathsf{n}_2 : \mathsf{Nat}\ \mathsf{n}_1 + \mathsf{n}_2 \equiv \mathsf{n}_1 + \mathsf{n}_2 \tag{6.34}$$

Damit ist das abstrakte Gleichheitsproblem gelöst. Da in diesem Beispiel das abstrakte Gleichheitsproblem und das konkrete Gleichheitsproblem ein und dasselbe sind und dies auch für alle angewendeten abstrakten Gleichungen gilt, ist der berechnet Plan schon der Beweis für das konkrete Gleichheitsproblem.

6.1.3 Beispiel für Planung mit Termstrukturen

In diesem Abschnitt geht es um den Beweis einer Gleichung, die aussagt, daß die größte natürliche Zahl einer nichtleeren Liste auch die größte natürliche Zahl der umgekehrten Liste ist. Dafür werden die Signatur Σ_{Nat} um ein Prädikatssymbol $<$ und Σ_{List} um zwei Funktionssymbole Max und rev erweitert. Die Spezifikation des Prädikatssymbols wir dnicht angegeben, da sie für unsere Zwecke hier nicht relevant ist. Die folgenden Gleichungen werden zu $\mathsf{Ax}_{\mathsf{List}}$ hinzugefügt:

$$(\mathsf{Ax}) \qquad \mathsf{rev}(\mathsf{Nil}) \equiv \mathsf{Nil} \tag{6.35}$$

$$(\mathsf{Ax}) \qquad \forall \mathsf{n} : \mathsf{Nat}\ \ \forall \mathsf{l} : \mathsf{List}\ \ \mathsf{rev}(\mathsf{cons}(\mathsf{n}, \mathsf{l})) \equiv \mathsf{app}(\mathsf{rev}(\mathsf{l}), \mathsf{cons}(\mathsf{n}, \mathsf{Nil})) \tag{6.36}$$

$$(\mathsf{Ax}) \qquad \forall \mathsf{n} : \mathsf{Nat}\ \mathsf{Max}(\mathsf{cons}(\mathsf{n}, \mathsf{Nil})) \equiv \mathsf{n} \tag{6.37}$$

$$(\mathsf{Ax}) \qquad \begin{array}{c} \forall \mathsf{n}, \mathsf{m} : \mathsf{Nat}\ \forall \mathsf{l} : \mathsf{List}\ \mathsf{n} < \mathsf{Max}(\mathsf{cons}(\mathsf{m}, \mathsf{l})) \\ \Rightarrow \mathsf{Max}(\mathsf{cons}(\mathsf{n}, \mathsf{cons}(\mathsf{m}, \mathsf{l}))) \equiv \mathsf{Max}(\mathsf{cons}(\mathsf{m}, \mathsf{l})) \end{array} \tag{6.38}$$

$$(\mathsf{Ax}) \qquad \begin{array}{c} \forall \mathsf{n}, \mathsf{m} : \mathsf{Nat}\ \forall \mathsf{l} : \mathsf{List}\ \mathsf{Max}(\mathsf{cons}(\mathsf{m}, \mathsf{l})) < \mathsf{n} \\ \Rightarrow \mathsf{Max}(\mathsf{cons}(\mathsf{n}, \mathsf{cons}(\mathsf{m}, \mathsf{l}))) \equiv \mathsf{n} \end{array} \tag{6.39}$$

Weiterhin sei folgendes Axiom gegeben:

$$(\mathsf{Ax}) \qquad \forall \mathsf{n} : \mathsf{Nat}\ \forall \mathsf{l} : \mathsf{List}\ \mathsf{Max}(\mathsf{rev}(\mathsf{cons}(\mathsf{n}, \mathsf{l}))) \equiv \mathsf{Max}(\mathsf{cons}(\mathsf{n}, \mathsf{l})) \tag{6.40}$$

Die Axiome werden bezüglich der größten gemeinsamen variablen Unterterme zu Strukturgleichungen abstrahiert. Dies sind

$$(\mathsf{Ax}') \qquad \forall \mathsf{n} : \mathsf{Nat} \ \ \forall \mathsf{l} : \mathsf{List} \ \ \mathsf{rev}(\mathsf{cons}(\mathsf{n}, \mathsf{l})) \equiv \mathsf{app}(\mathsf{rev}(\mathsf{l}), \mathsf{cons}(\mathsf{n}, @_{\mathsf{List}})) \tag{6.41}$$

$$(\mathsf{Ax}') \qquad \forall \mathsf{n} : \mathsf{Nat} \ \mathsf{Max}(\mathsf{cons}(\mathsf{n}, @_{\mathsf{List}})) \equiv \mathsf{n} \tag{6.42}$$

$$(\mathsf{Ax}') \qquad \forall \mathsf{l} : \mathsf{List} \ \mathsf{Max}(\mathsf{cons}(@_{\mathsf{Nat}}, \mathsf{l})) \equiv \mathsf{Max}(\mathsf{l}) \tag{6.43}$$

$$(\mathsf{Ax}') \qquad \forall \mathsf{n} : \mathsf{Nat} \ \mathsf{Max}(\mathsf{cons}(\mathsf{n}, @_{\mathsf{List}})) \equiv \mathsf{n} \tag{6.44}$$

$$(\mathsf{Ax}') \qquad \forall \mathsf{l} : \mathsf{List} \ \mathsf{Max}(\mathsf{rev}(\mathsf{l})) \equiv \mathsf{Max}(\mathsf{l}) \tag{6.45}$$

Ersetzungssystem Die Ordnung auf den Funktionssymbolen sei die folgende:

$$\mathsf{Nil} < \mathsf{cons} < \left\{ \begin{array}{c} \mathsf{app} \\ \mathsf{Max} \end{array} \right\} < \mathsf{rev}$$

Mit Hilfe der Ordnung $<_{\mathsf{TO}}$ können die Gleichungen gerichtet werden. Nach dem in Abschnitt 3.3.4 beschriebenen Verfahren, werden die gerichteten Gleichungen sukzessive in ein Strukturtermersetzungssystem eingefügt, wobei darauf gerachtet wird, daß keine kritischen Paare entstehen. Das resultierende Ersetzungsystem ist das folgende:

$$\mathsf{app}(\mathsf{rev}(\mathsf{l}), \mathsf{cons}(\mathsf{n}, @_{\mathsf{List}})) \to \mathsf{rev}(\mathsf{cons}(\mathsf{n}, \mathsf{l})) \tag{6.46}$$

$$\mathsf{Max}(\mathsf{cons}(\mathsf{n}, @_{\mathsf{List}})) \to \mathsf{n} \tag{6.47}$$

$$\mathsf{Max}(\mathsf{cons}(\mathsf{n}, @_{\mathsf{List}})) \to \mathsf{n} \tag{6.48}$$

$$\mathsf{Max}(\mathsf{rev}(\mathsf{l})) \to \mathsf{Max}(\mathsf{l}) \tag{6.49}$$

Da per Konstruktion in diesem Termersetzungssystem nur konvergierende kritische Paare enthalten sind, und die zur Orientierung der Gleichungen verwendete Ordnung $<_{\mathsf{TO}}$ eine Simplifikationsordnung ist, ist dieses Termersetzungssystem noethersch und konfluent. Somit gibt es eindeutige Normalformen für jeden Strukturterm.

Das zu beweisende Theorem ist

$$(\mathsf{Th}) \qquad \begin{array}{l} \forall \mathsf{l}_1, \mathsf{l}_2 : \mathsf{List} \ \neg \mathsf{app}(\mathsf{l}_1, \mathsf{l}_2) \equiv \mathsf{Nil} \\ \qquad \qquad \Rightarrow \mathsf{Max}(\mathsf{rev}(\mathsf{app}(\mathsf{l}_1, \mathsf{l}_2))) \equiv \mathsf{Max}(\mathsf{app}(\mathsf{l}_1, \mathsf{l}_2)) \end{array} \tag{6.50}$$

Den beiden Termen des Gleichheitsproblems entsprechen, bezüglich des gemeinsamen Untertermes $\mathsf{app}(\mathsf{l}_1, \mathsf{l}_2)$, die Strukturterme

$$\mathsf{Max}(\mathsf{rev}(\mathsf{AC}_1)) \quad \text{und} \quad \mathsf{Max}(\mathsf{AC}_1)$$

Der rechte Strukturterm kann nicht reduziert werden und ist somit schon in Normalform. Der rechte Term kann mit Hilfe der Regel (6.49) auf die Normalform $\mathsf{Max}(\mathsf{AC}_1)$ reduziert werden. Da die Normalformen gleich sind kann nun die Ableitung auf den Strukturtermen als Plan für den Beweis des konkreten Gleichheitsproblems verwendet werden.

Die einzige Gleichung, die der Strukturregel (6.49) entspricht, ist die Gleichung (6.40). Um diese auf den Term

$$\mathsf{Max}(\mathsf{rev}(\mathsf{app}(\mathsf{l}_1, \mathsf{l}_2)))$$

anzuwenden, muß folgendes Gleichheitsproblem gelöst werden:

$$\forall \mathsf{l}_1, \mathsf{l}_2 : \mathsf{List}\ \neg \mathsf{app}(\mathsf{l}_1, \mathsf{l}_2) \equiv \mathsf{Nil} \Rightarrow \exists \mathsf{n} : \mathsf{Nat}\ \exists \mathsf{l}' : \mathsf{List}\ \mathsf{app}(\mathsf{l}_1, \mathsf{l}_2) \equiv \mathsf{cons}(\mathsf{n}, \mathsf{l})$$

Der Beweis hierfür ist relativ einfach. Somit kann die Gleichung angewendet werden und der resultierende Term, nach Anwendung der Termsubstitution $[\mathsf{cons}(\mathsf{n}, \mathsf{l}) \backslash \mathsf{app}(\mathsf{l}_1, \mathsf{l}_2)]$ ist:

$$\mathsf{Max}(\mathsf{app}(\mathsf{l}_1, \mathsf{l}_2)),$$

womit das Gleichheitsproblem gelöst ist.

6.2 Ein Parserproblem

Das in diesem Abschnitt beschrieben Problem stammt aus der Verifikation eines Parsers. Zunächst wird ein Teil der Spezifikation des Parsers angegeben, sowie deren Abstraktion. Anschließend wird detailliert gezeigt, wie das Gesamtverfahren das in der Verifikation des Parsers anfallende Gleichheitsproblem löst.

6.2.1 Spezifikation

Es gibt insgesamt sieben Datentypen: Die in Abschnitt 6.1 eingeführte Datenstruktur Nat und deren Gleichungsspezifikation $\mathsf{Ax}_{\mathsf{Nat}}$, zwei hier nicht näher spezifizierte Datentypen für Bäume (Tree) und Knotenmarkierungen (Label) sowie eine Datenstruktur Tag mit zweielementiger Trägermenge.

$$\Sigma_{\mathsf{Tree}} = (\emptyset, \emptyset, \{\mathsf{Tree}\}) \qquad \Sigma_{\mathsf{Label}} = (\emptyset, \emptyset, \{\mathsf{Label}\})$$

$$\Sigma_{\mathsf{Tag}} = (\emptyset, \{\mathsf{BR} :\to \mathsf{Tag}, \mathsf{BRANCHES} :\to \mathsf{Tag}\}, \{\mathsf{Tag}\})$$

Anschließend gibt es noch drei Datentypen für Listen von Bäumen ($\mathsf{TreeList}$), Listen von Knotenmarkierungen und Kellerspeicher (Stack).

$$\Sigma_{\mathsf{TreeList}} = (\emptyset \\ \{ \quad \mathsf{Nil} :\to \mathsf{TreeList}\ , \mathsf{tcons} : \mathsf{Tree}\ \times \mathsf{TreeList}\ \to \mathsf{TreeList} \\ \mathsf{tcar} : \mathsf{Tree}\ \to \mathsf{Tree}\ , \mathsf{tcdr} : \mathsf{TreeList}\ \to \mathsf{TreeList} \\ \mathsf{tapp} : \mathsf{TreeList}\ \times \mathsf{TreeList}\ \to \mathsf{TreeList}\ \} \\ \{ \quad \mathsf{TreeList}\ \})$$

$$\Sigma_{\mathsf{LabelList}} = (\emptyset \\ \{ \quad \mathsf{Nil} :\to \mathsf{LabelList}\ , \mathsf{lcons} : \mathsf{Tree}\ \times \mathsf{TreeList}\ \to \mathsf{TreeList} \\ \mathsf{lcar} : \mathsf{Tree}\ \to \mathsf{Tree}\ , \mathsf{lcdr} : \mathsf{TreeList}\ \to \mathsf{TreeList} \\ \mathsf{lapp} : \mathsf{TreeList}\ \times \mathsf{TreeList}\ \to \mathsf{TreeList}\ \} \\ \{ \quad \mathsf{TreeList}\ \})$$

$$\Sigma_{\mathsf{Stack}} = (\emptyset \\ \{ \quad \mathsf{push} : \mathsf{Tree}\ \times \mathsf{Stack}\ \to \mathsf{Stack} \\ \mathsf{top} : \mathsf{Stack}\ \to \mathsf{Tree}\ , \mathsf{pop} : \mathsf{Stack}\ \to \mathsf{Stack}\ \} \\ \{ \quad \mathsf{Stack}\ \})$$

Die Gleichungsspezifikation für die Datenstruktur TreeList ist $\mathsf{Ax}_{\mathsf{TreeList}}$:

$$(\mathsf{Ax}) \qquad \forall \mathsf{t} : \mathsf{Tree}\ \forall \mathsf{l} : \mathsf{TreeList}\ \mathsf{tcons}(\mathsf{n}, \mathsf{l}) \neq \mathsf{Nil} \tag{6.51}$$

$$(\mathsf{Ax}) \qquad \mathsf{tcdr}(\mathsf{Nil}) \equiv \mathsf{Nil} \tag{6.52}$$

$$(\mathsf{Ax}) \qquad \forall \mathsf{t} : \mathsf{Tree}\ \forall \mathsf{l} : \mathsf{TreeList}\ \mathsf{tcdr}(\mathsf{tcons}(\mathsf{t}, \mathsf{l})) \equiv \mathsf{l} \tag{6.53}$$

$$(\mathsf{Ax}) \qquad \forall \mathsf{l} : \mathsf{TreeList}\ \mathsf{tapp}(\mathsf{Nil}, \mathsf{l}) \equiv \mathsf{l} \tag{6.54}$$

$$(\mathsf{Ax}) \qquad \forall \mathsf{t} : \mathsf{Tree}\ \forall \mathsf{l}_1, \mathsf{l}_2 : \mathsf{TreeList}\ \mathsf{tapp}(\mathsf{tcons}(\mathsf{t}, \mathsf{l}_1), \mathsf{l}_2) \equiv \mathsf{tcons}(\mathsf{t}, \mathsf{tapp}(\mathsf{l}_1, \mathsf{l}_2)) \tag{6.55}$$

Die Gleichungsspezifikation für die Datenstruktur LabelList ist $\mathsf{Ax}_{\mathsf{LabelList}}$:

$$(\mathsf{Ax}) \qquad \forall \mathsf{k} : \mathsf{Label}\ \forall \mathsf{l} : \mathsf{LabelList}\ \mathsf{lcons}(\mathsf{n}, \mathsf{l}) \neq \mathsf{Nil} \tag{6.56}$$

$$(\mathsf{Ax}) \qquad \mathsf{lcdr}(\mathsf{Nil}) \equiv \mathsf{Nil} \tag{6.57}$$

$$(\mathsf{Ax}) \qquad \forall \mathsf{k} : \mathsf{Label}\ \forall \mathsf{l} : \mathsf{LabelList}\ \mathsf{lcdr}(\mathsf{lcons}(\mathsf{k}, \mathsf{l})) \equiv \mathsf{l} \tag{6.58}$$

$$(\mathsf{Ax}) \qquad \forall \mathsf{l} : \mathsf{LabelList}\ \mathsf{lapp}(\mathsf{Nil}, \mathsf{l}) \equiv \mathsf{l} \tag{6.59}$$

$$(\mathsf{Ax}) \qquad \forall \mathsf{k} : \mathsf{Label}\ \forall \mathsf{l}_1, \mathsf{l}_2 : \mathsf{LabelList}\ \mathsf{lapp}(\mathsf{lcons}(\mathsf{k}, \mathsf{l}_1), \mathsf{l}_2) \equiv \mathsf{lcons}(\mathsf{t}, \mathsf{lapp}(\mathsf{l}_1, \mathsf{l}_2)) \tag{6.60}$$

Die Gesamtsignatur der Spezifikation des Parsers besteht aus den Signaturen der Datenstrukturen Nat , Tag , Label , Tree , TreeList und LabelList , sowie neun weiteren Funktionen:

$$
\begin{aligned}
\Sigma_{\mathsf{Parser}} = (\emptyset \\
\{ \quad & \mathsf{top_n} : \mathsf{Nat} \times \mathsf{Stack} \to \mathsf{TreeList}, \\
& \mathsf{pop_n} : \mathsf{Nat} \times \mathsf{Stack} \to \mathsf{Stack}, \\
& \mathsf{mk_{tree}} : \mathsf{Label} \times \mathsf{TreeList} \to \mathsf{Tree}, \\
& \mathsf{stack} : \mathsf{Tree} \to \mathsf{Stack}, \mathsf{mk_{list}} : \mathsf{Tree} \to \mathsf{TreeList}, \\
& \mathsf{labels} : \mathsf{Tag} \times \mathsf{Tree} \to \mathsf{LabelList}, \\
& \mathsf{leaves} : \mathsf{TreeList} \to \mathsf{LabelList}, \\
& \mathsf{alltrees} : \mathsf{Stack} \to \mathsf{TreeList}, \\
& \mathsf{reduce} : \mathsf{Label} \times \mathsf{Nat}\ \mathsf{Stack} \to \mathsf{Stack} \}) \\
\cup \quad & \Sigma_{\mathsf{Nat}} \cup \Sigma_{\mathsf{Tag}} \cup \Sigma_{\mathsf{Stack}} \cup \Sigma_{\mathsf{TreeList}} \\
& \cup \Sigma_{\mathsf{LabelList}} \cup \Sigma_{\mathsf{Label}} \cup \Sigma_{\mathsf{Tree}}
\end{aligned}
$$

Die zusätzlichen Funktionen werden durch die folgenden Gleichungen spezifiziert:

$$(\mathsf{Ax}) \quad \mathsf{leaves}(\mathsf{Nil}) \equiv \mathsf{Nil} \tag{6.61}$$

$$(\mathsf{Ax}) \quad \forall t : \mathsf{Tree}\ \forall l : \mathsf{TreeList}\ \mathsf{leaves}(\mathsf{tcons}(t, l)) \equiv \mathsf{lapp}(\mathsf{labels}(\mathsf{BR}, t), \mathsf{leaves}(l)) \tag{6.62}$$

$$(\mathsf{Ax}) \quad \mathsf{alltrees}(\mathsf{empty}) \equiv \mathsf{Nil} \tag{6.63}$$

$$(\mathsf{Ax}) \quad \forall t : \mathsf{Tree}\ \forall s : \mathsf{Stack}\ \mathsf{alltrees}(\mathsf{push}(t, s)) \equiv \mathsf{tapp}(\mathsf{alltrees}(s), \mathsf{mk_{list}}(t)) \tag{6.64}$$

$$(\mathsf{Ax}) \quad \begin{gathered} \forall l : \mathsf{Label}\ n : \mathsf{Nat}\ s : \mathsf{Stack}\ \mathsf{reduce}(l, n, s) \\ \equiv \mathsf{push}(\mathsf{mk_{tree}}(l, \mathsf{top_n}(n, s)), \mathsf{pop_n}(n, s)) \end{gathered} \tag{6.65}$$

$$(\mathsf{Ax}) \quad \forall t : \mathsf{Tree}\ \forall s : \mathsf{Stack}\ \mathsf{alltrees}(\mathsf{push}(t, s)) \equiv \mathsf{tapp}(\mathsf{alltrees}(s), \mathsf{mk_{list}}(t)) \tag{6.66}$$

$$(\mathsf{Ax}) \quad \forall l_1, l_2 : \mathsf{TreeList}\ \mathsf{leaves}(\mathsf{tapp}(l_1, l_2)) \equiv \mathsf{lapp}(\mathsf{leaves}(l_1), \mathsf{leaves}(l_2)) \tag{6.67}$$

$$(\mathsf{Ax}) \quad \forall t : \mathsf{Tree}\ \mathsf{leaves}(\mathsf{mk_{list}}(t)) \equiv \mathsf{labels}(\mathsf{BR}, t) \tag{6.68}$$

$$(\mathsf{Ax}) \quad \forall k : \mathsf{Label}\ \forall l : \mathsf{TreeList}\ \mathsf{labels}(\mathsf{BR}, \mathsf{mk_{tree}}(k, l)) \equiv \mathsf{leaves}(l) \tag{6.69}$$

$$(\mathsf{Ax}) \quad \forall n : \mathsf{Nat}\ \forall s : \mathsf{Stack}\ \mathsf{alltrees}(s) \equiv \mathsf{tapp}(\mathsf{alltrees}(\mathsf{pop_n}(n, s)), \mathsf{top_n}(n, s)) \tag{6.70}$$

Die Menge $\mathsf{Ax}_{\mathsf{Parser}}$ der Gleichungen zur Spezifikation des Parsers besteht aus den Axiomenmengen $\mathsf{Ax}_{\mathsf{Nat}}$, $\mathsf{Ax}_{\mathsf{TreeList}}$, $\mathsf{Ax}_{\mathsf{LabelList}}$ und den Gleichungen (6.61)-(6.70).

6.2.1.1 Abstraktion

Die Abstraktionen der Gleichungen werden bezüglich der größten gemeinsamen Unterterme (siehe Abschnitt 3.2.1) der Gleichungen gebildet. Auf eine Auflistung der abstrakten Gleichungen wird an dieser Stelle verzichtet. Die im Beweis betrachteten abstrakten Gleichungen werden dort angegeben.

6.2.2 Gleichheitsproblem

Für die Verifikation des Parsers muß unter anderem folgende Gleichung bewiesen werden

$$(\mathsf{Th}) \qquad \begin{array}{l} \forall \mathsf{k} : \mathsf{Label}\ \forall \mathsf{n} : \mathsf{Nat}\ \forall \mathsf{s} : \mathsf{Stack} \\ \qquad \mathsf{leaves}(\mathsf{alltrees}(\mathsf{reduce}(\mathsf{k}, \mathsf{n}, \mathsf{s}))) \equiv \mathsf{leaves}(\mathsf{alltrees}(\mathsf{s})) \end{array} \qquad (6.71)$$

Die Abstraktion bezüglich des gemeinsamen Untertermes s dieses Gleichheitsproblems ist die folgende abstrakte Gleichung: Dabei werden wegfallende quantifizierte Variablen eliminiert:

$$(\mathsf{Th}') \qquad \begin{array}{l} \forall \mathsf{s} : \mathsf{Stack} \\ \qquad \mathsf{leaves}(\mathsf{alltrees}(\mathsf{reduce}(@_{\mathsf{Label}}, @_{\mathsf{Nat}}, \mathsf{s}))) \equiv \mathsf{leaves}(\mathsf{alltrees}(\mathsf{s})) \end{array} \qquad (6.72)$$

Die Abstraktionsinformation des abstrakten Gleichheitsproblems (6.72) bezüglich des betrachteten gemeinsamen Untertermes s ist:

	Unterterm s
$\mathsf{leaves}(\mathsf{alltrees}(\mathsf{reduce}(@_{\mathsf{Label}}, @_{\mathsf{Nat}}, \mathsf{s})))$	$[(\mathsf{leaves}, 1), (\mathsf{alltrees}, 1), (\mathsf{reduce}, 3)]$
$\mathsf{leaves}(\mathsf{alltrees}(\mathsf{s}))$	$[(\mathsf{leaves}, 1), (\mathsf{alltrees}, 1)]$
Skelett	$[(\mathsf{leaves}, 1), (\mathsf{alltrees}, 1)]$

Abstraktionsinformation von (6.72)

Aus dieser Abstraktionsinformation wird die Information gewonnen, daß zunächst versucht werden muß den Kontext $(\mathsf{reduce}, 3)$ entweder in der Funktionssymbolliste zu dem Vorkommen von s in dem linken Term nach außen zu schieben, oder diesen zu elimieren. Beides muß unter Erhaltung des Skelettes erfolgen. Die Suche nach einer Gleichung, die unter Erhaltung des Skelettes entweder den Kontext nach außen schiebt oder eliminiert, ist erfolglos. Die einzige Gleichung, in der überhaupt das Funktionssymbol reduce vorkommt, erhält nicht das Skelett. Somit ist das Verfahren an dieser Stelle zunächst einmal blockiert.

An dieser Stelle kommt die in Abschnitt 3.2.4.3 beschriebene Variante der skelettverändernden Heuristiken zum tragen. Diese stellt fest, daß, würde die Gleichung (6.65) angewendet werden, würde der Kontext $(\mathsf{reduce}, 3)$ ersetzt werden durch neue Kontexte. Für die Funktionssymbole in diesen neuen Kontexten gibt es aber in der Datenbasis mehr abstrakte Gleichungen, die diese Kontexte bearbeiten, als es Gleichungen gab, die die Kontexte, in denen das Funktionssymbol reduce auftrat, bearbeiteten. Deshalb wird die Gleichung (6.65) angewendet, auch wenn das resultierende Gleichheitsproblem bezüglich der Ordnung $<_{GO}$ nicht kleiner ist als das Gleichheitsproblem (6.72). Das neue Gleichheitsproblem ist

$$(\mathsf{Th'}) \quad \begin{array}{l} \mathsf{leaves}(\mathsf{alltrees}(\mathsf{push}(\mathsf{mk}_{\mathsf{tree}}(@_{\mathsf{Label}}, \mathsf{top_n}(@_{\mathsf{Nat}}, \mathsf{s})), \mathsf{pop_n}(@_{\mathsf{Nat}}, \mathsf{s})))) \\ \quad \equiv \mathsf{leaves}(\mathsf{alltrees}(\mathsf{s})) \end{array} \tag{6.73}$$

Die Abstraktionsinformation für das neue Gleichheitsproblem (6.73) ist:

	Unterterm s
$\mathsf{leaves}(\mathsf{alltrees}(\mathsf{push}\ldots))$	$[(\mathsf{leaves}, 1), (\mathsf{alltrees}, 1), (\mathsf{push}, 2), (\mathsf{pop_n}, 2)]$
$\mathsf{leaves}(\mathsf{alltrees}(\mathsf{s}))$	$[(\mathsf{leaves}, 1), (\mathsf{alltrees}, 1)]$
Skelett	$[(\mathsf{leaves}, 1), (\mathsf{alltrees}, 1)]$

Abstraktionsinformation von (6.73)

Daß dem Vorkommen von s im rechten Term das zweite Vorkommen von s im linken Term zugeordnet wird, liegt daran, daß die beiden das größte gemeinsame Skelett bezüglich ihrer Funktionssymbollisten haben und die Größe des Kontextes bezüglich dieses Skelettes kleiner ist als der Kontext für das Vorkommen von s im rechten Term und dem ersten Vorkommen von s im linken Term.

Aus der Abstraktionsinformation erhält man die Information, daß die Kontexte $(\mathsf{push}, 2)$ und $(\mathsf{pop_n}, 2)$ entweder relativ zum Skelett $[(\mathsf{leaves}, 1), (\mathsf{alltrees}, 1)]$ nach außen geschoben werden sollen oder eliminiert werden sollen. Mit Hilfe der Gleichung (6.66) kann dies realisiert werden. Da die Abstraktion dieser Gleichung bezüglich deren gemeinsamen Variablen wieder dieselbe Gleichung ergibt, wird im folgenden nur von der Gleichung (6.66) gesprochen und nicht von deren Abstraktion. Die Abstraktionsinformation der Gleichung (6.66) ist die folgende

	Unterterm t	Unterterm s
$\mathsf{alltrees}(\mathsf{push}(\mathsf{t}, \mathsf{s}))$	$[(\mathsf{alltrees}, 1), (\mathsf{push}, 1)]$	$[(\mathsf{alltrees}, 1), (\mathsf{push}, 2)]$
$\mathsf{tapp}(\mathsf{alltrees}(\mathsf{s}), \mathsf{mk_{list}}(\mathsf{t}))$	$[(\mathsf{tapp}, 2), (\mathsf{mk_{list}}, 1)]$	$[(\mathsf{tapp}, 1), (\mathsf{alltrees}, 1)]$
Skelett	$[]$	$[(\mathsf{alltrees}, 1)]$

Abstraktionsinformation von (6.66)

Aus dieser Abstraktionsinformation wird sichtbar, daß die Gleichung (6.66), wenn von links nach rechts angewendet, den Kontext $(\mathsf{push}, 2)$ bezüglich des Vorkommen von s im linken Term nach oben verschiebt. Gleichzeitig erhält sie das Skelett $[(\mathsf{alltrees}(,)1)]$. Da sie an der Termposition $[1]$ auf den linken Term des Gleichheitsproblems angewendet wird, bleibt außerdem auch der andere Skelettteil $[(\mathsf{leaves}, 1)]$ erhalten. Somit erhält man das folgende neue Gleichheitsproblem:

$$(\mathsf{Th}') \quad \begin{array}{l}\mathsf{leaves}(\mathsf{tapp}(\mathsf{alltrees}(\mathsf{pop_n}(@_{\mathsf{Nat}}, \mathsf{s})),\\ \qquad \mathsf{mk_{list}}(\mathsf{mk_{tree}}(@_{\mathsf{Label}},\\ \qquad\qquad \mathsf{top_n}(@_{\mathsf{Nat}}, \mathsf{s}))))\end{array} \equiv \mathsf{leaves}(\mathsf{alltrees}(\mathsf{s})) \qquad (6.74)$$

Es gilt auch, daß das neue Gleichheitsproblem (6.74) bezüglich der Ordnung $<_{GO}$ kleiner ist als das Gleichheitsproblem (6.73), da die Größe des Kontextes in der Abstraktionsinformation kleiner geworden ist. Die Abstraktionsinformation des neuen Gleichheitsproblems ist:

	Unterterm s
$\mathsf{leaves}(\mathsf{tapp}(\mathsf{alltrees}(\ldots), \ldots))$	$[(\mathsf{leaves}, 1), (\mathsf{tapp}, 1), (\mathsf{alltrees}, 1), (\mathsf{pop_n}, 2)]$
$\mathsf{leaves}(\mathsf{alltrees}(\mathsf{s}))$	$[(\mathsf{leaves}, 1), (\mathsf{alltrees}, 1)]$
Skelett	$[(\mathsf{leaves}, 1), (\mathsf{alltrees}, 1)]$

Abstraktionsinformation von (6.74)

Jetzt gilt es die Kontexte $(\mathsf{tapp}, 1)$ und $(\mathsf{pop_n}, 2)$ entweder weiter nach außen zu schieben oder zu eliminieren. Beides schlägt jedoch fehl, da es keine geeignete Gleichung in der Datenbasis gibt. An dieser Stelle setzt die vorkommenorientierte Heuristik aus Abschnitt 3.2.4.2 an: Die Differenz in der Anzahl der Vorkommen von s in den beiden Termen des Gleichheitsproblems (6.74) ist 1. Diese Heuristik sucht nach einer Gleichung, die diese Differenz verringert und das bestehende Skelett erhält. Gelingt dies, ist das neue Gleichheitsproblem per Definition bezüglich der Ordnung $<_{GO}$ kleiner als das jetzige Gleichheitsproblem (6.74). Die einzige anwendbare abstrakte Gleichung ist die Abstraktion der Gleichung (6.70). Deren Abstraktion bezüglich der gemeinsamen Variablen ist:

$$(\mathsf{Ax}') \quad \forall \mathsf{s} : \mathsf{Stack}\ \mathsf{alltrees}(\mathsf{s}) \equiv \mathsf{tapp}(\mathsf{alltrees}(\mathsf{pop_n}(@_{\mathsf{Nat}}, \mathsf{s})), \mathsf{top_n}(@_{\mathsf{Nat}}, \mathsf{s})) \qquad (6.75)$$

Diese abstrakte Gleichung erhöht, wenn von links nach rechts angewendet, die Anzahl der Vorkommen des Untertermes s um 1. Somit wird die Differenz der Anzahl der Vorkommen im Gleicheitsproblem kleiner, nämlich 0. Das neue Gleichheitsproblem ist

$$(\mathsf{Th}') \quad \begin{array}{l}\mathsf{leaves}(\\ \quad \mathsf{tapp}(\mathsf{alltrees}(\mathsf{pop_n}(@_{\mathsf{Nat}}, \mathsf{s})), \mathsf{mk_{list}}(\mathsf{mk_{tree}}(@_{\mathsf{Label}}, \mathsf{top_n}(@_{\mathsf{Nat}}, \mathsf{s})))))\\ \equiv \mathsf{leaves}(\mathsf{tapp}(\mathsf{alltrees}(\mathsf{pop_n}(@_{\mathsf{Nat}}, \mathsf{s})), \mathsf{top_n}(@_{\mathsf{Nat}}, \mathsf{s})))\end{array} \qquad (6.76)$$

Die Abstraktionsinformation des Gleichheitsproblems (6.76) ist:

	Vorkommen von s
$\mathsf{leaves}(\mathsf{tapp}(\ldots, \mathsf{mk}_{\mathsf{list}}(\ldots)))$	$[(\mathsf{leaves}, 1), (\mathsf{tapp}, 1), (\mathsf{alltrees}, 1), (\mathsf{pop}_{\mathsf{n}}, 2)]$
$\mathsf{leaves}(\mathsf{tapp}(\ldots, \mathsf{top}_{\mathsf{n}}(\ldots)))$	$[(\mathsf{leaves}, 1), (\mathsf{tapp}, 1), (\mathsf{alltrees}, 1), (\mathsf{pop}_{\mathsf{n}}, 2)]$
Skelett	$[(\mathsf{leaves}, 1), (\mathsf{tapp}, 1), (\mathsf{alltrees}, 1), (\mathsf{pop}_{\mathsf{n}}, 2)]$
hline	Vorkommen von s
$\mathsf{leaves}(\mathsf{tapp}(\ldots, \mathsf{mk}_{\mathsf{list}}(\ldots)))$	$[(\mathsf{leaves}, 1), (\mathsf{tapp}, 2), (\mathsf{mk}_{\mathsf{list}}, 1), (\mathsf{mk}_{\mathsf{tree}}, 2), (\mathsf{top}_{\mathsf{n}}(, , 2))]$
$\mathsf{leaves}(\mathsf{tapp}(\ldots, \mathsf{top}_{\mathsf{n}}(\ldots)))$	$[(\mathsf{leaves}, 1), (\mathsf{tapp}, 2), (\mathsf{top}_{\mathsf{n}}(, , 2))]$
Skelett	$[(\mathsf{leaves}, 1), (\mathsf{tapp}, 2), (\mathsf{top}_{\mathsf{n}}(, , 2))]$

Abstraktionsinformation von (6.76)

Aus dieser Abstraktionsinformation folgt, daß die Kontexte bezüglich des zweiten Vorkommen von s im linken Term nach außen verschoben oder eliminiert werden müssen. Wieder gibt es dafür keine geeignete abstrakte Gleichung. Die einzige abstrakte Gleichung, die das Funktionssymbol $\mathsf{mk}_{\mathsf{list}}$ eliminieren würde, ist die Abstraktion der Gleichung (6.68):

$$(\mathsf{Ax}') \qquad \forall \mathsf{t} : \mathsf{Tree}\ \mathsf{leaves}(\mathsf{mk}_{\mathsf{list}}(\mathsf{t})) \equiv \mathsf{labels}(@_{\mathsf{Tag}}, \mathsf{t}) \tag{6.77}$$

Doch diese ist nicht anwendbar auf den linken Term. Deshalb versucht die Skelettverändernde Heuristik aus Abschnitt 3.2.4.3 das Skelett so zu transformieren, daß die abstrakte Gleichung (6.77) anwendbar wird.

Es läßt sich feststellen, daß, um die abstrakte Gleichung auf den linken Term anzuwenden, ein Funktionssymbol leaves oberhalb des Funktionssymbols $\mathsf{mk}_{\mathsf{list}}$ eingeführt werden muß. Damit aber das Skelett nicht kleiner wird, muß auf dem rechten Term die gleiche Transformation ausgeführt werden. Die skelettverändernde Heuristik sucht also nach einer Gleichung, die sie auf den Term anwenden kann, um das Funktionssymbol $\mathsf{leaves}(\ )$an geeigneter Stelle zu erzeugen. Die einzige Gleichung, die dies leistet und auch gleichzeitig auf dem rechten Term anwendbar ist, ist die Gleichung (6.67):

$$\mathsf{leaves}(\mathsf{tapp}(\mathsf{l}_1, \mathsf{l}_2)) \equiv \mathsf{lapp}(\mathsf{leaves}(\mathsf{l}_1), \mathsf{leaves}(\mathsf{l}_2))$$

Nach Anwendung dieser Gleichung auf beide Terme des Gleichheitsproblems erhält man folgende Gleichung:

$$(\mathsf{Th}') \qquad \begin{gathered} \mathsf{lapp}(\mathsf{leaves}(\mathsf{alltrees}(\mathsf{pop}_{\mathsf{n}}(@_{\mathsf{Nat}}, \mathsf{s}))), \\ \mathsf{leaves}(\mathsf{mk}_{\mathsf{list}}(\mathsf{mk}_{\mathsf{tree}}(@_{\mathsf{Label}}, \mathsf{top}_{\mathsf{n}}(@_{\mathsf{Nat}}, \mathsf{s})))) \\ \equiv \mathsf{lapp}(\mathsf{leaves}(\mathsf{alltrees}(\mathsf{pop}_{\mathsf{n}}(@_{\mathsf{Nat}}, \mathsf{s}))), \mathsf{leaves}(\mathsf{top}_{\mathsf{n}}(@_{\mathsf{Nat}}, \mathsf{s}))) \end{gathered} \tag{6.78}$$

Das Skelett dieses Gleichheitsproblems ist genau so groß, wie das von (6.76). Das Gleichheitsproblem (6.78) muß nicht bezüglich der Ordnung $<_{\mathsf{GO}}$ kleiner sein als

(6.76), da es nur als Zwischenschritt für die Anwendung der abstrakten Gleichung (6.77) dient. Auf den linken Term kann jetzt die ausgewählte Gleichung (6.77) angewendet werden. Die Anwendung dieser Gleichung würde aber das Skelett zerstören, bzw. verkleinern. Deshalb versucht die Heuristik auf dem rechten Term eines der durch die Gleichung (6.77) im linken Term erzeugten Funktionssymbole im rechten Term an der korrespondierenden Stelle zu erzeugen. Dies gelingt durch Anwendung von rechts nach links der Abstraktion der Gleichung (6.69):

$$(\mathsf{Ax}) \qquad \forall \mathsf{l} : \mathsf{TreeList}\ \mathsf{labels}(@_{\mathsf{Tag}}, \mathsf{mk}_{\mathsf{tree}}(@_{\mathsf{Label}}, \mathsf{l})) \equiv \mathsf{leaves}(\mathsf{l}) \tag{6.79}$$

Durch Anwendung der abstrakten Gleichung (6.77) auf den linken Term und der abstrakten Gleichung (6.79) auf den rechten Term des Gleichheitsproblems (6.78), erhält man das folgende Gleichheitsproblem:

$$(\mathsf{Th}') \qquad \begin{array}{l} \mathsf{lapp}(\mathsf{leaves}(\mathsf{alltrees}(\mathsf{pop}_\mathsf{n}(@_{\mathsf{Nat}}, \mathsf{s}))), \\ \qquad \mathsf{labels}(@_{\mathsf{Tag}}, \mathsf{mk}_{\mathsf{tree}}(@_{\mathsf{Label}}, \mathsf{top}_\mathsf{n}(@_{\mathsf{Nat}}, \mathsf{s})))) \\ \equiv \mathsf{lapp}(\mathsf{leaves}(\mathsf{alltrees}(\mathsf{pop}_\mathsf{n}(@_{\mathsf{Nat}}, \mathsf{s}))), \\ \qquad \mathsf{labels}(@_{\mathsf{Tag}}, \mathsf{mk}_{\mathsf{tree}}(@_{\mathsf{Label}}, \mathsf{top}_\mathsf{n}(@_{\mathsf{Nat}}, \mathsf{s})))) \end{array} \tag{6.80}$$

Diese Gleichheitsproblem ist bezüglich der Ordnung $<_{\mathcal{GO}}$ kleiner als das Gleichheitsproblem (6.76) und es ist ein triviales Gleichheitsproblem.

Der berechnete Plan sieht folgendermaßen aus: Auf die beiden Terme des ursprünglichen Gleichheitsproblems (6.71) müssen sukzessive Gleichungen an vorgegebenen Termpositionen angewendet werden, von denen die Gleichungen (6.65), (6.66), (6.67) und (6.77) respektive eine Abstraktion sind. In diesem Fall sind die gesuchten konkreten Gleichungen eindeutig bestimmt. Für den linken Term sieht der Plan vor, Gleichungen anzuwenden, von denen die Gleichungen (6.70), (6.67) und (6.79 respektive Abstraktionen sind. Auch hier sind die anzuwendenden konkreten Gleichungen eindeutig bestimmt. Die während der Planausführung entstehenden Unterprobleme bei der Anwendung einer Gleichung beschränken sich auf reine Unifikationsprobleme. Nach der Planausführung erhält man das folgende Gleichheitsproblem

$$(\mathsf{Th}) \qquad \begin{array}{l} \mathsf{lapp}(\mathsf{leaves}(\mathsf{alltrees}(\mathsf{pop}_\mathsf{n}(\mathsf{n},))\mathsf{s}), \mathsf{labels}(\mathsf{BR}, \mathsf{mk}_{\mathsf{tree}}(\mathsf{k}, \mathsf{top}_\mathsf{n}(\mathsf{n}, \mathsf{s})))) \equiv \\ \mathsf{lapp}(\mathsf{leaves}(\mathsf{alltrees}(\mathsf{pop}_\mathsf{n}(\mathsf{n}',))\mathsf{s}), \mathsf{labels}(\mathsf{BR}, \mathsf{mk}_{\mathsf{tree}}(\mathsf{k}', \mathsf{top}_\mathsf{n}(\mathsf{n}', \mathsf{s})))) \end{array} \tag{6.81}$$

Die Variablen n' und k' sind frei instantiierbare Variablen, die durch die Anwendung der Gleichungen (6.70) und (6.69) in den rechten Term kamen. Dadurch ist das Gleichheitsproblem durch Instantiierung von n' durch n und k' durch k gelöst.

Kapitel 7

Zusammenfassung

In der vorliegenden Arbeit wurde das in der KI verwendete Prinzip der Abstraktion mit dem Paradigma der *Differenz Reduktion* kombiniert. Zur Erprobung dieser Kombination wurde das reine Geichheitsbeweisen als Anwendungsgebiet gewählt. Dafür wurden zwei Abstraktionsarten für Gleichungen eingeführt, nämlich die Abstraktionsebene der “abstrakten Terme” und die der “Strukturterme”. Der Grund, warum Abstraktionen verwendet wurden, ist der, daß es durch den Wegfall von Informationen durch die Abstraktion leichter ist einen Beweis zu finden. Leichter ist es deshalb, da sich nicht mehr um alle Details gekümmert werden muß. Dies setzt allerdings voraus, daß die für den Beweis wichtigen Informationen durch die Abstraktion erhalten bleiben. In dieser Arbeit wurden die gemeinsamen Unterterme einer Gleichung sowie deren Positionen innerhalb der beiden Terme der Gleichung als Kriterium dafür genommen, was “wichtige” Informationen einer Gleichung sind. Dementsprechend sind die beiden Abstraktionen so gewählt, daß sie alle nicht gemeinsamen Unterterme “wegabstrahieren”. Anhand von speziellen Heuristiken wird dann versucht, die Abstraktion eines Gleichheitsproblems zu beweisen. Gelingt der abstrakte Beweis, soll dieser als Plan für das konkrete Gleichheitsproblem dienen. Hierfür muß dieser allerdings zu einem konkreten Beweis verfeinert werden. Ein gegebenes Gleichheitsproblem wird also zunächst zu einem abstrakten Gleichheitsproblem abstrahiert. Auf dieser ersten Abstraktionsebene der “abstrakten Terme”, wird dann versucht einen abstrakten Beweis für das abstrakte Gleichheitsproblem zu finden. Scheitert dies, kann das abstrahierte Gleichheitsproblem ein weiteres Mal zu einem Strukturgleichheitsproblem abstrahiert werden. Gelingt der Beweis auf dieser zweiten Abstraktionsstufe, muß der dort erhaltene Strukturbeweis verfeinert werden. Dieser kann entweder direkt in einen konkreten Beweis transformiert werden oder erst in einen abstrakten Beweis, der dann anschließend zu einem konkreten Beweis verfeinert wird.

Für beide Abstraktionsebenen wurden spezielle Heuristiken entwickelt, die nach dem Paradigma der *Differenz Reduktion* vorgehen. Der Grund, warum spezielle Heuristiken entworfen wurden und beispielsweise nicht ein bekanntes widerlegungsvollständiges Verfahren auf den abstrakten Ebenen verwendet wurde, ist der, daß die Abstraktion einer erfüllbaren Menge von Gleichungen unerfüllbar werden kann; dies wurde in Kapitel 2 gezeigt und hat zur Folge, daß ein widerlegungsvollständiges Verfahren schon einen Widerspruch zwischen den Abstraktionen der Axiome herleiten könnte, ohne daß dabei die Abstraktion der zu beweisenden Gleichung involviert ist. Ein solcher abstrakter Widerspruchsbeweis kann aber kein Plan für den Beweis der konkreten Gleichung sein. Dieses Ergebnis ordnet sich in die von Giunchiglia und Walsh [GW92a] gemachten allgemeinen Analysen diverser Abstraktionsarten ein. In [GW92a] stellen sie fest, daß praktisch alle in der KI entwickelten Abstraktionen einen konsistenten Raum zu einem inkonsistenten Raum abstrahieren. Wichtig ist allerdings, daß die Beweisbarkeit erhalten bleibt. Das bedeutet, daß jede Abstraktion eines Theorem des "Grund"-Raumes auch Theorem des "abstrakten"-Raumes ist. Für die in dieser Arbeit entwickelten Abstraktionsarten gilt dies nur insofern, daß jedes Gleichheitsproblem zu einem trivialen Gleichheitsproblem abstrahiert werden kann. Eine richtige Einordnung der hier definierten Abstraktionen in die von Giunchiglia und Walsh definierten Klassen ist aber nicht möglich, da sie nur solche Abstraktionen betrachten, die nach einem festen Schema abstrahieren, wohingegen bei den hier beschriebenen Abstraktionen die Abstraktion einer Gleichung nicht eine eindeutig bestimmte abstrakten Gleichung oder Strukturgleichung ist. Vielmehr ist die Abstraktion einer Gleichung parametrisiert über die betrachteten Vorkommen von gemeinsamen Untertermen dieser Gleichung, wodurch einer Gleichung eine Klasse von abstrakten Gleichungen und Strukturgleichungen entspricht.

Die entwickelten Abstraktionen und Heuristiken wurden im INKA-System implementiert und auf ihre praktische Tauglichkeit hin überprüft. Einige ausgewählte Beispiele wurden in Kapitel 6 aufgeführt. Für alle Axiome werden in dem Moment, in dem sie in die Datenbasis aufgenommen werden, die Abstraktionen berechnet. Die daraus resultierenden abstrakten Gleichungen und Strukturgleichung werden analysiert und nach bestimmten Kriterien sortiert unter verschiedenen Typen in den abstrakten Datenbasen abgelegt. Dies ermöglicht den beschriebenen Heuristiken unmittelbar auf abstrakte Gleichungen oder Strukturgleichungen zuzugreifen, was einen positiven Effekt auf das Zeitverhalten bei der Suche nach geeigneten Gleichungen hat. Die Strukturgleichungen werden zudem mit Hilfe der beschriebenen Terminierungsordnung gerichtet und gegebenenfalls in das in Kapitel 3 beschriebene "Strukturterm"-Ersetzungssystem eingefügt. Für die Reduktion eines Strukturtermes gestaltet sich die Suche nach anwendbaren Reduktionsregeln allerdings noch sehr aufwendig, was sich insbesondere im Fall großer Datenbasen bemerkbar macht. Hauptsächlich für die Reduktion eines Strukturtermes aber auch für eine genauere Auswahl geeigneter Gleichungen eines bestimmten Typs wird die in Zukunft im INKA-System zur Verfügung stehende Technik des *Term-Indexing* [GM93] eine starke Verbesserung des Laufzeitverhaltens bringen. Dennoch lassen sich mit der bisherigen Implementierung schon praktisch taugliche Ergebnisse erzielen. So beträgt zum Beispiel die Planungszeit, d. h. das Finden eines abstrakten Beweises, für das Parser-Beispiel aus Kapitel 6 unter 1 CPU Sekunde. Ermittelt wurde dieser Wert auf einer SPARCStation 20 mit 256 MB Hauptspeicher.

Bei der Entwicklung der vorgestellten Abstraktionen und Heuristiken stellte sich heraus, daß die Entwicklungen von Heuristiken und Abstraktionen eng miteinander gekoppelt sind. Dies bedeutet, daß es nicht sinnvoll ist, zuerst eine Abstraktion zu definieren und dann anschließend Heuristiken für diese Abstraktionen zu entwickeln. Vielmehr scheinen die Arbeiten zur Entwicklung von Abstraktionen und Heuristiken stark miteinander verwoben zu sein.
Darüber hinaus ergaben sich einige Fragestellungen die im Rahmen dieser Arbeit unbeantwortet geblieben sind: Eine erste allgemeine Fragestellung ist folgende: Ist es sinnvoller, in einem Beweissystem einige wenige Abstraktionen zur Verfügung zu haben und diese mit mächtigen heuristischen Komponenten zu versehen, oder ist es sinnvoller viele Abstraktionen zu entwickeln und diese mit sehr speziellen Heuristiken auszustatten? Wird eine Entscheidung zu Gunsten des ersteren gefällt, muß die meiste Arbeit in die Entwicklung der heuristischen Apparate zu den einzelnen Abstraktionen investiert werden. Wird der zweite Ansatz verfolgt, wird eine Fülle spezieller Abstraktionen und Verfahren entwickelt, deren Zusammenspiel in irgendwie koordiniert werden muß, was im ersten Ansatz relativ leichter sein müßte. Dabei ist die Koordinierung der einzelnen Abstraktionen und Heuristiken ein zweites offenes Problem, insbesondere im Zusammenhang mit Hierarchien von Abstraktionen. Dabei tauchen ähnliche Probleme wie bei der Verfeinerung eines Planes zu einem konkreten Beweis oder zu einem Beweis einer "niedrigeren" Abstraktionsstufe auf.

Aus diesen offenen Fragestellungen ergeben sich einige Anregungen für weitere Forschungen auf dem Gebiet der Verwendung von Abstraktionen zum Planen in automatischen Beweissystemen. Zum Beispiel können neue Abstraktionen und Heuristiken für andere Problemklassen als die durch diese Arbeit abgedeckten Probleme entwickelt werden. Dies ist wohl insbesondere im Hinblick auf die Klärung der ersten obigen Fragestellung wichtig. Desweiteren könnte versucht werden, den in dieser Arbeit gewählten Ansatz auf allgemeine prädikatenlogische Probleme zu übertragen. Dies könnte für Beweise der Äquivalenz zweier Literale, in denen gemeinsame Unterterme enthalten sind, oder für Beweise der Äquivalenz zweier Formeln, in denen gemeinsame Teilformeln vorkommen, untersucht werden. Bezüglich der zweiten oben erwähnten Fragestellung ergibt sich ein breites Untersuchungsfeld im Bereich der Verfeinerung von Plänen und der Koordinierung der verschiedenen Heuristiken verschiedener Abstraktionen. Hierfür wird aber eine Reihe von Abstraktionen mitsamt Heuristiken benötigt, die eventuell zunächst entwickelt werden müssen. Im Gegensatz zu diesen eher praktischen Perspektiven weiterer Forschungen auf diesem Gebiet, stellt die Erweiterung der von Giunchiglia und Walsh betrachteten Klasse von Abstraktionen eine mehr theoretische Perspektive dar. In einer Erweiterung dieser Klasse müßten dann auch solche Fragestellungen erörtert werden, wie die von Giunchiglia und Walsh definierten Typen von Abstraktionen mitsamt der daraus folgenden Resultate in sinnvoller Art und Weise übernommen werden könnten.

Danksagungen

An erster Stelle möchte ich meinem Betreuer Dieter Hutter danken, der mir immer wieder wichtige Ratschläge für die Vollendung dieser Arbeit geben konnte. Auch dafür, daß er mich immer wieder vor der Verfolgung vielfältiger Fährten mit ungewissen Ergebnis bewahrt hat, gebührt ihm meine Dankbarkeit. Weiterhin möchte ich Herrn Prof. Dr. Siekmann für die Vergabe des Themas danken. Danken möchte ich auch Claus Sengler, der mich in vielen Gesprächen immer wieder motivierte und dessen moralische Unterstützung mir eine große Hilfe war, sowie den restlichen Mitarbeitern des VSE Projektes, Bruno Langenstein, Andreas Wolpers und Werner Stephan, die mich so freundlich in ihrer Runde aufgenommen haben. Dem ganzen VSE Projekt und der Infrastrukturgruppe der DFKI GmbH gilt mein Dank für die hervorragenden Arbeitsbedingungen, die man nur jedem Diplomanden wünschen kann. Weiterhin danke ich Christoph Benzmüller für die Diskussionen über diverse Probleme im Bereich des automatischen Beweisens, die mich immer wieder Dinge etwas klarer haben erkennen lassen. In der letzten Phase der Diplomarbeit gilt mein besonderer Dank meiner Freundin Sandra Veeck, die mich desöfteren entbehren oder meine schlechte Laune ertragen mußte, sowie dafür, daß sie mich immer wieder unterstützt und die Arbeit korrekturgelesen hat. Für das Korrekturlesen danke ich auch Christoph Benzmüller und Claus Sengler. Ein Dankeschön gebührt auch Mike Fischer und Sacha Luthe, die mich während der Anfertigung dieser Arbeit immer wieder motiviert haben. Nicht zuletzt möchte ich meinen Eltern danke, die mir das Studium ermöglichten und den Spaß am wissenschaftlichen Arbeiten immer gefördert haben.

Literaturverzeichnis

[BI95] Alan Bundy and Andrew Ireland. Productive use of failure in inductive proof. Technical report, Department of Artificial Intelligence, University of Edinburgh, 80 South Bridge, Edinburgh EH1 1HN, Scotland, 1995.

[BS88] Karl Hans Bläsius and Jörg Siekmann. Partial unification for graph based equational reasoning. In Ewing L. Lusk and Ross A. Overbeek, editors, *Proceedings 9^th International Conference on Automated Deduction (CADE), Lecture Notes in Computer Science (LNCS) 310*, pages 397–414, Argonne, Illinois, USA, 1988. Springer-Verlag, Berlin, Germany.

[BSvH+93] Alan Bundy, Andrew Stevens, Frank van Harmelen, Andrew Ireland, and Alan Smaill. Rippling: a heuristic for guiding inductive proofs. *Artificial Intelligence, North Holland Publishing Company, Amsterdam, The Netherlands*, 62:185–253, 1993.

[BvHHS90] A. Bundy, F. van Harmelen, C. Horn, and A. Smaill. The Oyster-Clam System. In Mark E. Stickel, editor, *Proceedings 10^th International Conference on Automated Deduction (CADE), Lecture Notes in Artificial Intelligence (LNAI) 449*, pages 647 – 648, Kaiserslautern, Germany, July 1990. Springer-Verlag, Berlin, Germany.

[BvHSI90] Alan Bundy, Frank van Harmelen, Alan Smaill, and Andrew Ireland. Extensions to the rippling-out tactic for guiding inductive proofs. In Mark E. Stickel, editor, *Proceedings 10^th International Conference on Automated Deduction (CADE), Lecture Notes in Artificial Intelligence (LNAI) 449*, pages 132–146, Kaiserslautern, Germany, July 1990. Springer-Verlag, Berlin, Germany.

[BW81] A. Bundy and B. Welham. Using meta-level inference for selective application of multiple rewrite rule sets in algebraic manipulation. *Artificial Intelligence*, 16:189–211, 1981.

[BW92] David Basin and Toby Walsh. Difference matching. In Deepak Kapur, editor, *Proceedings 11^th International Conference on Automated*

Deduction (CADE), Lecture Notes in Artificial Intelligence (LNAI) 607, pages 295–309, Saratoga Springs, New York, USA, June 1992. Springer-Verlag, Berlin, Germany.

[BW93] David Basin and Toby Walsh. Difference unification. In Ruzena Bajcsy, editor, *Proceedings 13^{th} International Joint Conference on Artificial Intelligence (IJCAI)*, pages 116–122, San Mateo, California, USA, 1993. Morgan Kaufmann.

[CAB$^+$86] Robert L. Constable, S. Allen, H. Bromly, W. Cleaveland, J. Cremer, R. Harper, D. Howe, T. Knoblock, N. Mendler, P. Panangaden, J. Sasaki, and S. Smith. *Implementing Mathematics with the Nuprl Proof Development System.* Prentice-Hall, Englewood Cliffs, New Jersey, 1986.

[CHS94] J. Cleve, D. Hutter, and C. Sengler. *INKA-Handbuch.* Deutsches Forschungszentrum für künstliche Intelligenz GmbH, Saarbrücken, March 1994.

[Chu36] Alonzo Church. A note on the entscheidungsproblem. *Journal of Symbolic Logic*, 1, 1936.

[Com89] *Computational Logic – Essays in Honor of Alan Robinson.* MIT Press, Cambridge, 1989.

[dB80] Nicolaas Govert de Bruijn. A survey of the project automath. In *[HS80]*, pages 579–606, 1980.

[Dig79] Vincent J. Digricoli. Resolution by unification and equality. In Joyner [Joy79].

[EFT86] Heinz-Dieter Ebbinghaus, Jörg Flum, and Wolfgang Thomas. *Einführung in die mathematische Logik.* Wissenschaftliche Buchgesellschaft, Darmstadt, Germany, 2^{nd} edition, 1986. The first edition was published 1978.

[Fit90] Melvin Fitting. *First-Order Logic and Automated Threorem Proving.* Springer Verlag, 1990.

[Fre78] G Frege. Begriffsschrift, eine der arithmetischen nachgebildete formelsprache, 1878.

[GM93] Peter Graf and Christoph Meyer. Extended path-indexing. Technical Report MPI-I-93-253, Max-Planck-Institut für Informatik, Saarbrücken, December 1993.

[GMW79] M. Gordon, R. Milner, and C. Wadsworth. *Edinburgh LCF: A Mechanized Logic of Computation.* Number 78 in LNCS. Springer Verlag, 1979.

[Göd31] Kurt Gödel. Über formal unentscheidbare Sätze der Principia Mathematica und verwandter Systeme I. *Monatshefte der Mathematischen Physik*, 38:173–198, 1931. English Version in [vH67].

[GW89] Fausto Giunchiglia and Toby Walsh. Abstract theorem proving. In *Proceedings of 11^{th} International Joint Conference on Artificial Intelligence (IJCAI)*, 1989. Also available as DAI Research Paper No 430, Department of Artificial Intelligence, University of Edinburgh.

[GW92a] F. Giunchiglia and T. Walsh. The inevitability of inconsistent abstract spaces. *Journal of Automated Reasoning*, 11:23–41, 1992. Also IRST-Technical Report 9006-16, IRST, Trento, Italy.

[GW92b] Fausto Giunchiglia and Toby Walsh. A theory of abstraction. *Journal of Artificial Intelligence*, 57(2-3):323–390, 1992. Also IRST-Technical Report 9001-14, IRST, Trento, Italy.

[Hel92] Christoph Helferich. *Geschichte der Philosophie: Von den Anfängen bis zur Gegenwart und östliches Denken.* J. B. Metzlersche Verlagsbuchhandlung und Carl Ernst Poeschel Verlag GmbH, Stuttgart, Deutschland, zweite auflage edition, 1992.

[Her30] Jaques Herbrand. *Recherches sur la théorie de la démonstration.* PhD thesis, Université de Paris, 1930. Englisch translation in [vH67].

[HO80] Gérard P. Huet and Derek C. Oppen. Equations and rewrite rules: A survey. Technical Report CSL-111, SRI International, 1980.

[HRS90] Maritta Heisel, Wolfgang Reif, and Werner Stephan. Tactical Theorem Proving in Program Verification. In Stickel [Sti90], pages 115–131.

[HRS91] Maritta Heisel, Wolfgang Reif, and Werner Stephan. *Automating Software Design*, chapter 21, pages 547–574. AAAI Press, 1991.

[HS80] R. Hindeley and J. Seldin, editors. *To H. B. Curry: Essays in Combinator Logic, Lambda Calculus and Formalisms.* Academic Press, 1980.

[Hua89] Xiaorong Huang. Proof transformation towards human reasoning style. In Dieter Metzing, editor, *Proceedings of the of 13th GWAI*, number 216 in Informatik-Fachberichte, pages 37–42. Springer Verlag, 1989.

[Hua95] Xiaorong Huang. Transforming resolution proofs into natural deduction proofs at the assertion level. Seki-report, Fachbereich Informatik, Universität des Saarlandes, Saarbrücken, Germany, 1995. Forthcoming.

[Hut91] Dieter Hutter. *Mustergesteuerte Strategien zum Beweisen von Gleichheiten*. PhD thesis, Universität Karlsruhe, Karlsruhe, 1991.

[Joy79] William H. Joyner, editor. *Proceedings of the 4th Workshop on Automated Deduction*, Austin, Texas, USA, 1979.

[Jut79] L.S. van Benthem Jutting. *Checking Landau's "Grundlagen" in the AUTOMATH System*, volume 83 of *Mathematical Centre Tracts*. Mathematisch Centrum, Amsterdam, Netherlands, 1979.

[KB70] Donald E. Knuth and Peter B. Bendix. Simple word problems in universal algebras. In J. Leech, editor, *Computational Problems in Abstract Algebra*, pages 263–297. Pergamon Press, 1970.

[Lan30] Edmund Landau. *Grundlagen der Analysis*. Wissenschaftliche Buchgesellschaft, Darmstadt, Deutschland, zweite auflage edition, 1930.

[Lei86] G. W. Leibniz. Projêt et essais pour arriver à quelques certitudes pour finir une bonne partie des disputes et pour avancer l'art d'inventer, 1686. Deutsche Übersetzung aus G. W. Leibniz, Fragmente zur Logik, Akademie Verlag, Berlin, 1960.

[LO88] Ewing L. Lusk and Ross A. Overbeek, editors. *Proceedings of the 9th Conference on Automated Deduction*, number 310 in LNCS, Argonne, Illinois, USA, 1988.

[Mat86] H. Matsumura. Commutative ring theory. In *Cambridge Studies in Advanced Mathematics*, volume 8. Cambridge University Press, 1986. Translated from Japanese 'Kakan kan ron' by Miles Reid.

[McC90] W. W. McCune. Otter 2.0 users guide. Technical Report ANL-90-9, Argonne National Laboratory, Argonne, Illinois 60439, USA, 1990.

[Men87] Elliott Mendelson. *Introduction to Mathematical Logic*. The Wadsworth & Brooks/Cole, Advanced Books and Software, Mathematics Series. Wadsworth, Monterey, California 93940, USA, 3^{rd} edition, 1987. The first edition was published 1964.

[Mor69] James B. Morris. *E*-resolution. In Walker and Norton [WN69], pages 287–294.

[NSS56] Allen Newell, J. C. Shaw, and Herbert A. Simon. The logic theory machine. *IRE Transactions Information Theory, IT-2(3)*, 1956.

[NSS63] Allen Newell, J. C. Shaw, and Herbert A. Simon. *The Logic Theory Machine*, volume Computer and Thought. McGraw-Hill, Amsterdam, The Netherlands, Feigenbaum and Feldman edition, 1963.

[OS89] Hans Jürgen Ohlbach and Jörg Siekmann. The Markgraf Karl Refutation Procedure. In Complog89 [Com89], pages 41–112.

[Pea89] G. Peano. Arithmetices principia, novo methodo exposita. *Fratres Bocca, Turin*, 1889.

[Pla81] David Plaisted. Theorem proving with abstractions. *Artificial Intelligence, North Holland Publishing Company, Amsterdam, The Netherlands*, 16:47–108, 1981.

[Pla86] D. A. Plaisted. Abstractions using generalization functions. In *Proceedings 8^{th} International Conference on Automated Deduction (CADE)*, pages pages 365–376. Springer-Verlag, Berlin, Germany, 1986.

[Plu87] D. Plummer. *Gazing: Controlling the Use of Rewrite Rules*. PhD thesis, Department of Artificial Intelligence, University of Edinburgh, 80 South Bridge, Edinburgh EH1 1HN, Scotland, 1987.

[PN90] Lawrence C. Paulson and Tobias Nipkow. Isabelle tutorial and user's manual. Technical Report 189, Computer Laboratory, University of Cambridge, January 1990.

[Prä92] Axel Präcklein. Rewriting, narrowing, compilation, and difference reduction heuristics for equality reasoning in resolution theorem proving. Seki-report, Fachbereich Informatik, Universität Kaiserslautern, Postfach 3049, 67653 Kaiserslautern, Germany, 1992.

[Ric78] Michael M. Richter. *Logikkalküle*, volume 43 of *Leitfäden der angewandten Mathematik und Mechanik*. B. G. Teubner, Stuttgart, Germany, 1978.

[Rob65] J. A. Robinson. A machine-oriented logic based on the resolution principle. *Journal of the Association for Computing Machinery (ACM), ACM, Inc., 211 East 43^{rd} Street, New York, 10017, USA*, 12(1):23–41, 1965.

[RW10] B. Russel and A. N. Whitehead. *Principia Mathematica*, volume 1. Cambridge University Press, Großbritanien, zweite auflage edition, 1910.

[RW69] Arthur Robinson and Larry Wos. Paramodulation and TP in first order theories with equality. *Machine Intelligence*, 4:135–150, 1969.

[SA91] Volker Sperschneider and Grigorius Antoniou. *LOGIC: A Foundation for Computer Science*. Addison Wesley, 1st ed. edition, 1991.

[Sac74] E.D. Sacerdoti. Planning in a hierarchy of abstraction spaces. *Artificial Intelligence*, 5:115–135, 1974.

[SBB+82] L. Sterling, A. Bundy, L. Byrd, R. O'Keefe, and B. Silver. Solving symbolic equations with PRESS. In J. Calmet, editor, *Computer Algebra*, volume 144 of *Lecture Notes in Computer Science*, pages 109–116. Springer-Verlag, 1982.

[Sti90] Mark Stickel, editor. *Proceedings of the 10th Conference on Automated Deduction*, number 449 in LNCS, Kaiserslautern, Germany, 1990.

[Tur37] Alan Turing. On computable numbers, with an application to the entscheidungsproblem, 1937.

[UR89] A. Unruh and P. Rosenbloom. Abstraction in problem solving and learning. In *Proceedings of the 11th IJCAI*, 1989.

[vH67] Jean van Heijenoort, editor. *From Frege to Gödel A Source Book in Mathematical Logic, 1879-1931*. Source Books in the History of the Sciences. Harvard University Press, 1967.

[WN69] Donald E. Walker and Lewis Norton, editors. *Proceedings of the 1st International Joint Conference on Artificial Intelligence*, 1969.

[ZK88] Hantao Zhang and Deepak Kapur. First order theorem proving using conditional rewrite rules. In Lusk and Overbeek [LO88], pages 1–20.

Index