Integriertes automatisches und interaktives Beweisen: die Kalkülebene

Martin Giese

Diplomarbeit

Institut für Logik,
Komplexität und Deduktionssysteme
Universität Karlsruhe

Betreuer:
Prof. Dr. W. Menzel
Dipl.-Inform. Wolfgang Ahrendt

30. Juni 1998

Überarbeitete Version vom 25. September 1998

Überarbeitete Version

Dies ist die erste überarbeitete Version der Arbeit. Sie unterscheidet sich von der abgegebenen Version in erster Linie durch die Korrektur etlicher Tippfehler. Außerdem wurde das Seiten-Layout geändert, wodurch einige Seiten gespart werden konnten. Für die vielen Hinweise auf Fehler sei Reiner Hähnle herzlich gedankt.

Danksagung

Mein besonderer Dank gilt Wolfgang Ahrendt für die sehr intensive Betreuung bei der Erstellung dieser Arbeit. Ferner sei Bernhard Beckert für seine Diskussionsbereitschaft gedankt. Schließlich möchte ich meinen Eltern für die geduldige Sicherung meines Lebensunterhalts danken.

Erklärung

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

Karlsruhe, den 30. Juni 1998

Martin Giese

Inhaltsverzeichnis

1. Einleitung **5**

2. Anforderungen **9**
2.1. Beschränkung auf analytische Nicht-Normalform-Kalküle . . . 9
2.2. Anforderungen des interaktiven Beweisens . . . 10
2.3. Anforderungen des automatischen Beweisens . . . 11
2.4. Entwicklungsmethodik für einen integrierten Kalkül . . . 11
2.5. Die Präsentationsebene . . . 13

3. Sequenzenkalkül vs. Tableaukalkül **14**
3.1. Sequenzenkalkül . . . 14
3.2. Tableaukalkül . . . 16

4. Metavariablen statt freier Variablen **19**
4.1. Die Problematik der freien Variablen . . . 19
4.2. Metavariablen . . . 20
4.3. Beispiel . . . 21
4.4. Trennung von Beweisbaum und Substitution . . . 22
4.5. Einschränkungen an die Substitution . . . 24
4.6. Zusammenfassung . . . 26

5. ϵ-Terme statt Skolemfunktionen **27**
5.1. Skolemisierung . . . 27
5.2. Motivation einer ϵ-Logik . . . 29
5.3. Syntax . . . 33
5.4. Semantik . . . 36
5.5. Einbettung der Prädikatenlogik in die ϵ-Logiken . . . 42
5.6. Grundkalküle für die ϵ-Logik . . . 45
5.7. Umbenennung von gebundenen Variablen . . . 61
5.8. Erweiterungen der ϵ-Logik . . . 63
5.9. Zusammenfassung . . . 67

6. Gleichheitsbehandlung mit Constraints **68**
6.1. Gleichheitsbehandlung mit Metavariablen . . . 69
6.2. Constraintbasierte Verfahren zur automatischen Gleichheitsbehandlung . . . 71
6.3. Constraintsprachen . . . 73
6.4. Ein Kalkül mit Metavariablen . . . 76
6.5. Termordnungen für ϵ-Terme . . . 88

6.6. Präsentation von Beweisen mit Constraints . . . 90
6.7. Steuerung des automatischen Beweisers mit Constraints . . . 95
6.8. Zusammenfassung . . . 99

7. Universeller Präfix statt universeller Variablen **101**
7.1. Universelle Variablen in Tableaukalkülen . . . 101
7.2. Beweisen mit universellem Präfix . . . 103
7.3. Korrektheit von $\mathcal{U}_E$ . . . 115
7.4. Vollständigkeit von $\mathcal{U}_S$ . . . 117
7.5. Verhältnis zum Verfahren von Beckert . . . 121
7.6. Zusammenfassung . . . 126

8. Hypergraphenkalkül **127**
8.1. Metavariablen . . . 128
8.2. ϵ-Terme . . . 129
8.3. Constraints . . . 129
8.4. Universelle Präfixe . . . 132
8.5. Zusammenfassung . . . 132

9. Fazit **133**

A. ι und η: Die Vorfahren der ϵ-Terme **135**

B. Zur Implementierung der ϵ-Logik **138**
B.1. Repräsentation durch Bäume und Graphen . . . 138
B.2. Zusammenfassung . . . 140

C. Verzeichnis der beschriebenen Kalküle **141**

1. Einleitung

Thema der Arbeit

Es soll ein Kalkül zur Behandlung der Prädikatenlogik erster Stufe mit Gleichheit entworfen werden, der für die gemeinsame Verwendung vom interaktiven und automatischen Teil eines integrierten Beweissystems geeignet ist. Dabei soll besonderer Wert darauf gelegt werden, daß

- fortgeschrittene Verfahren aus dem Bereich des automatischen Beweisens innerhalb des Kalküls darstellbar sind,
- die interaktive Benutzung des Kalküls durch derartige Verfeinerungen nicht zu sehr kompliziert wird, und
- der Kalkül so homogen wie möglich ist, also nicht etwa eine bloße Vereinigung von Konzepten des automatischen und interaktiven Beweisens ist.

Weiterhin sollen Möglichkeiten zur Kommunikation zwischen dem menschlichen Benutzer und dem automatischen Teil des Systems im Auge behalten werden. Die Implementierung eines Systems, das diesen Kalkül verwendet, ist nicht Teil der Aufgabenstellung

Kontext der Arbeit

Diese Arbeit ist Teil des Bestrebens, ein maschinelles Beweissystem zu schaffen, das die Stärken von interaktiven und automatischen Systemen vereint.

Interaktive Systeme bieten die Möglichkeit, einen Beweis formal streng und lückenlos zu führen. Das System läßt keine Schlüsse der Art „wie man leicht sieht“ zu, es zwingt zu einer vollständigen, detaillierten Beweisführung. Erfahrungsgemäß ist es möglich, für prädikatenlogisch formalisierte Aussagen mit Hilfe interaktiver Systeme Beweise zu finden, deren Struktur einer vom Benutzer zuvor erarbeiteten Beweisidee folgt. Das heißt, wenn es gelungen ist, eine Aussage zu formalisieren, macht auch die Formalisierung eines dazugehörigen Beweises meist keine allzu großen Probleme. Die große Schwäche bestehender interaktiver Systeme liegt jedoch gerade in jenen Schritten, die „man leicht sieht“. Durch die Notwendigkeit, diese Trivialitäten in allen Einzelheiten zu zeigen, geht viel Zeit verloren. Die zahlreichen Teilbeweise von Aussagen, die für den Menschen offensichtlich sind, nehmen oft einen größeren Raum ein als die eigentliche Beweisidee. Zwar verfügen interaktive Systeme meist über die Möglichkeit, gewisse Abfolgen von Beweisschritten automatisch durchzuführen. Diese Automatiken sind jedoch deterministisch: bestehen mehrere Alternativen, einen Beweis fortzusetzen, wird nicht nach der richtigen gesucht, sondern der Benutzer gefragt.

Automatische Beweiser sind hingegen auf die automatische *Suche* von Beweisen spezialisiert. Der Benutzer manipuliert nur noch gewisse Parameter, die die Beweissuche beeinflussen. Sind die Parameter festgelegt, fängt der Automat an, nach einem Beweis zu suchen. Im wesentlichen lauten die möglichen Antworten dieses Prozesses „Erfolg“ und „Mißerfolg“. In bestehenden Systemen hat der Benutzer nicht die Möglichkeit, auf die Beweissuche lokal einflußzunehmen, etwa durch Angabe einzelner Regelanwendungen oder Instantiierungen. Seine einzige Möglichkeit, dem System bei der Suche zu helfen, besteht in der Festlegung einiger globaler Parameter *a priori*. Aufgrund der großen Suchräume versagen automatische Verfahren aber bei größeren Problemen schnell, wären also auf Hilfe vom Anwender durchaus angewiesen.

Ideal wäre ein Beweissystem, das dem Benutzer erlaubt, einen Beweis bis zu dem Detaillierungsgrad einzugeben, an dem die Einzelschritte für ihn trivial erscheinen, und die Validierung dieser Einzelschritte einem automatischen Beweiser zu überlassen.

In [1] wurde ein Schritt in diese Richtung getan, indem eine Verbindung zwischen dem interaktiven Verifikations-System KIV [18, 12] und dem automatischen Tableaubeweiser $_3T^AP$ [6] geschaffen wurde. Diese Verbindung bietet in KIV die Möglichkeit, den *Abschluß* eines Astes dem automatischen Beweiser zu übertragen. Dazu werden die prädikatenlogischen[1] Formeln des Beweisziels an $_3T^AP$ übergeben. Wenn der Automat einen Beweis findet, wird der entsprechende Beweisast abgeschlossen, sonst geschieht nichts. Effektiv erhält der KIV-Sequenzenkalkül ein neues Axiomenschema

$$\frac{}{\Gamma \vdash \Delta}\ {_3T^AP}$$

falls $\Gamma \vdash \Delta$ von $_3T^AP$ bewiesen werden kann.

An diesem Ansatz ist jedoch unbefriedigend, daß auf KIV Seite nicht nachzuvollziehen ist, *welcher* Beweis für die Aussage vom Automaten gefunden wurde. Da die Einflußnahme auf den automatischen Beweiser auch bei diesem Ansatz auf die Vorgabe einer Anfangsbedingung, nämlich die in der Sequenz stehenden Formeln, beschränkt bleibt, wäre es insbesondere nützlich, im Falle des Mißerfolgs zu analysieren, an welcher Stelle der Automat Schwierigkeiten hat.

Darum wurde in [19] ein Verfahren entwickelt, um von $_3T^AP$ gefundene Tableau-Beweise in Beweise des Sequenzenkalküls von KIV zu übersetzen. Doch, wie ein italienisches Sprichwort sagt, *traduttore, traditore,* wer übersetzt ist ein Verräter! Während $_3T^AP$ einen Tableaukalkül mit freien Variablen verwendet, ist der KIV-Kalkül ein Grundkalkül. Dieser wesentliche Aspekt des automatischen Beweissystems muß bei der Übersetzung also zwangsläufig verloren gehen. Die δ^{++}-Skolemisierung von $_3T^AP$ läßt Beweise zu, deren Struktur für eine Übersetzung in den KIV-Kalkül derart geändert werden muß, daß exponentiell größere Beweise entstehen. Auch die Verwendung von universellen Variablen in $_3T^AP$ kann erheblich größere Übersetzungen von Beweisen zur Folge haben. Zwar kann also aus einem geschlossenen $_3T^AP$-Beweis ein KIV-Beweis gewonnen werden. Letzterer gibt allerdings nach wie vor nicht unbedingt darüber Aufschluß, was $_3T^AP$ tatsächlich getan hat.

Aus dieser Problematik heraus entstand der Wunsch nach einem System, in dem keine Übersetzung nötig ist, da die interaktive und die automatische Komponente mit demselben Kalkül arbeiten. Der Kalkül sollte *mindestens* stark genug sein, um die Prädikatenlogik erster Stufe mit Gleichheit vollständig zu behandeln, da algebraische Spezifikationen

[1] KIV verwendet die dynamische Logik, die von $_3T^AP$ nicht behandelt werden kann. Daher können nicht alle Formeln übergeben werden.

(erster Stufe), auf die sich die wichtigsten Anwendungen maschineller Beweissysteme beziehen, diese Logik verwenden. Wie im letzten Absatz angedeutet, unterscheiden sich Kalküle für automatische Beweissysteme teilweise recht drastisch von solchen für den interaktiven Gebrauch. Beim Versuch, die beiden Ansätze zu vereinen, stößt man auf eine ganze Reihe von Schwierigkeiten.

Da der gemeinsame Kalkül die zentrale Komponente eines integrierten Beweissystems ist, muß er sorgfältig entworfen werden, wenn das System schließlich für die praktische Anwendung geeignet sein soll. Daher widmet sich diese Arbeit der Untersuchung von Entwurfsmöglichkeiten für einen solchen Kalkül.

Eine besondere Stellung nimmt bei dieser Untersuchung die automatische Behandlung der Gleichheit ein: Diese bereitet noch viel größere Probleme als die reine Prädikatenlogik, da die auftretenden Suchräume schon bei kleinen Problemen sehr groß werden. Es ist daher zu erwarten, daß auf diesem Gebiet künftig noch erhebliche Fortschritte gemacht werden. Der Kalkül sollte so angelegt sein, daß von derartigen Entwicklungen mit möglichst wenigen Änderungen profitiert werden kann. Exemplarisch wird gezeigt, wie sich das Verfahren von Beckert [4] in den erarbeiteten Rahmen einfügen läßt.

Eine Implementierung sollte nicht Teil der Arbeit sein, da vor einer praktisch verwendbaren Implementierung, außer der Kalkülebene, noch etliche andere Fragestellungen zu behandeln sind. Zum Beispiel beschränken sich die Vorschläge zur Steuerung des automatischen Beweisers durch den Benutzer in dieser Arbeit auf Methoden, die eng mit dem Entwurf des Kalküls zusammenhängen. Für ein leistungsfähiges integriertes System müssen jedoch auf jeden Fall noch weitere Möglichkeiten der Einflußnahme erforscht werden. Auch zur Präsentation, also zur benutzerfreundlichen Darstellung von Beweisen und Regeln werden in dieser Arbeit nur einige Vorschläge gemacht, die sich aus dem entwickelten Formalismus ergeben. Diese haben lediglich den Zweck, zu zeigen, daß die vorgestellten Kalküle in angemessener Weise präsentiert werden *können* und einige Alternativen aufzuzeigen. Es geht hier noch nicht darum, festzustellen, welche dieser Alternativen *optimal* ist.

Aufbau

In Kapitel 2 werden zunächst die Anforderungen an einen Kalkül für das integrierte automatische und interaktive Beweisen konkretisiert. Es wird eine Entwicklungsmethodik für den Kalkül vorgestellt und gerechtfertigt, die sich an verschiedenen Aspekten eines Nicht-Normalform-Tableaukalküls für das automatische Beweisen orientiert. Der Hauptteil der Arbeit ist gemäß dieser Methodik gegliedert. Zuerst wird jedoch in Kapitel 3 ein kurzer Überblick über die Verwandtschaft zwischen dem Nicht-Normalform-Tableaukalkül mit freien Variablen und dem Sequenzenkalkül gegeben. Dieses Kapitel wurde eingefügt, um die sprachlichen Barrieren und häufig auftretenden Mißverständnisse zwischen Verwendern der beiden Kalküle abzubauen.

In Kapitel 4 beginnt der Hauptteil der Arbeit mit der Einführung des Konzepts der Metavariablen, die an die Stelle der freien Variablen des Tableaukalküls treten sollen. In Kapitel 5 werden die ϵ-Terme vorgestellt, die eine intuitivere und mächtigere Alternative zu Skolemfunktion bieten. Deren Semantik wird ausführlich diskutiert, und es werden Grundkalküle für deren Behandlung eingeführt.

In Kapitel 6 werden Constraints als Hilfsmittel zur automatischen Gleichheitsbehandlung eingeführt. Es werden Kalküle mit Metavariablen, ϵ-Termen und Constraints angegeben. Außerdem wird gezeigt, wie Constraints für die Beweispräsentation und die

Steuerung des automatischen Beweisers benutzt werden können. Schließlich wird in Kapitel 7 eine Methode zur Behandlung von universellen Variablen vorgeschlagen. Es werden wieder dementsprechende Kalküle angegeben, und gezeigt, wie das von Beckert in [4] beschriebene Verfahren zur automatischen Gleichheitsbehandlung in diese einzubetten ist.

In Kapitel 8 wird diskutiert, wie sich die vorgestellten Ansätze mit dem Konzept der Hypergraphen von Preiß [17] vereinen lassen. Kapitel 9 enthält eine Zusammenfassung der Ergebnisse dieser Arbeit, sowie einen Ausblick auf mögliche weitere Forschung. In Anhang A wird kurz die Einführung von ϵ-Termen in Hilberts Werk diskutiert. Anhang B enthält einige Bemerkungen zur Implementierbarkeit von ϵ-Logiken. Da eine ganze Reihe verschiedener Kalküle beschrieben wird, auf die auch des öfteren Bezug genommen wird, gibt Anhang C einen Überblick über alle verwendeten Kalküle.

Zum Stil

Zu den Eigenheiten dieser Arbeit gehört der Wechsel zwischen etwas vagen Passagen einerseits, und aufwendigen, technisch detaillierten Beweisen andererseits. Diese stilistische Diskrepanz ergibt sich aus der Aufgabenstellung: Ein Kalkül, der für den interaktiven Gebrauch geeignet ist, muß für den Menschen handhabbar sein. Dies ist eine psychologische Komponente, die schwer abzuwägen ist, *da es noch kein integriertes System auf der Basis eines gemeinsamen Kalküls gibt.* Am ehesten kann man sich auf Erfahrungen mit bestehenden interaktiven Systemen stützen, aber die dort auftretenden Probleme sind eben andere, als die eines integrierten Systems. Man ist also auf Vermutungen und Schätzungen angewiesen, wenn es um die interaktive Seite der Studie geht. Dennoch wurde versucht, Intuitionen bezüglich der interaktiven Verwendung soweit wie möglich zu formalisieren, um mit Hilfe dieser Formalisierung präzise Schlüsse zu ziehen.

Andererseits werden in Kapitel 5 die sogenannten ϵ-Terme in die Logik eingeführt, durch die Formeln syntaktisch in Terme eingebettet werden. Dies führt zu Phänomenen, die in der klassischen Prädikatenlogik nicht auftreten, wie Terme mit gebundenen Variablen. Viele Aussagen, die für die Logik ohne ϵ-Terme sonst als selbstverständlich hingenommen werden, müssen nun genau unter die Lupe genommen werden, und mitunter mit Einschränkungen versehen werden. Bei der Erarbeitung der Korrektheits- und Vollständigkeitsbeweise (siehe etwa Abschnitt 5.6.2) wurde klar, daß an dieser Stelle die größte Vorsicht geboten ist, da allzu oft Beweisansätze verworfen werden mußten, weil sich einige, vermeintlich triviale, Aussagen als falsch herausstellten!

2. Anforderungen

In dieser Arbeit wird ein Kalkül für homogenes, integriertes interaktives und automatisches Beweisen gesucht. Es sollte also versucht werden, alle für einen Beweis wichtige Information innerhalb eines einheitlichen Rahmens darzustellen. Nun wird ein Teil der Vorgänge immer implizit bleiben: Die syntaktische Unifikation beim Abschluß von Ästen könnten beispielsweise durchaus als deduktiver Vorgang betrachtet werden. Diese deshalb im Rahmen des Kalküls mitzubehandeln, hieße aber sicherlich, über das Ziel hinauszuschießen. Vielmehr sollte der Kalkül bis zu der Ebene hinuntergehen, auf der ein Mensch bei einem detaillierten, technisch exakten Beweis auf dem Papier arbeitet. Ab dieser Ebene sollte aber jeder Beweisschritt im Rahmen des Kalküls darstellbar sein, und nicht etwa für Teilprobleme den Übergang zu einem anderen Formalismus erfordern.

Sucht man nach einem Kalkül, der für automatisches und interaktives Beweisen geeignet ist, so muß man zunächst untersuchen, welche Anforderungen diese beiden Felder an den Kalkül stellen. Da diese Anforderungen aber zum großen Teil von der Grundstruktur des Kalküls abhängen, kann man kaum konkrete Aussagen treffen, ohne sich hierin zu beschränken.

2.1. Beschränkung auf analytische Nicht-Normalform-Kalküle

Interaktive Beweissysteme arbeiten mit Kalkülen, die nicht auf Normalformen operieren. Eine Einschränkung auf Normalformen wäre für den menschlichen Beweiser nicht einsichtig. Beim interaktiven Beweisen ist es nötig, die Herkunft der betrachteten Formeln aus dem ursprünglichen Beweisziel nachvollziehen zu können, um eine problemnahe Beweisplanung zu ermöglichen. Ein für das *integrierte* Beweisen geeigneter Kalkül sollte also auch auf Normalformen verzichten.

Zum interaktiven Beweisen wurden bislang hauptsächlich der Sequenzenkalkül (KIV) und „natürliches Schließen“ (*natural deduction* in Isabelle) verwendet. Bei der *natural deduction* ergibt sich für das interaktive Beweisen das Problem, daß in ihrer üblichen Darstellung an einem fertigen Beweis nicht mehr sehr gut zu sehen ist, welche der zusätzlichen Prämissen, die etwa beim Beweisen von Implikationen auftauchen, an welchen Stellen im Beweis benutzt werden und benutzt werden dürfen. Verwaltet man die momentan zur Verfügung stehenden Prämissen hingegen explizit, so gelangt man zu einer eingeschränkten Form des Sequenzenkalküls. Letzterer ist jedoch symmetrischer und dadurch strukturell leichter durchschaubar.

Für das automatische Beweisen haben unter den *Nicht-Normalform* Kalkülen sicherlich die analytischen Tableaux mit freien Variablen die größte Bedeutung. Vergleicht man nun analytische Tableaux mit freien Variablen mit dem Sequenzenkalkül, so beobachtet man eine große strukturelle Ähnlichkeit. Betrachtet man nämlich die Grundversion der

Tableaux, instantiiert man also die freien Variablen mit den zum Abschluß verwendeten Substitutionen, so erhält man einen Kalkül, der sehr eng mit dem Sequenzenkalkül zusammenhängt: Einer Sequenz entspricht die Menge aller noch zu betrachtenden Formeln auf einem Ast des Tableaux.

Diese Verwandtschaft legt es nahe, sich in der folgenden Untersuchung auf analytische Nicht-Normalform-Kalküle vom Sequenz/Tableautyp zu beschränken. Insbesondere kann bei den zu stellenden Anforderungen auf die besonderen Eigenschaften dieser Kalküle eingegangen werden.

Wenn an verschiedenen Stellen Regeln oder Beweise angegeben werden, so wird die Sequenzenschreibweise benutzt. Diese hat den Vorteil, daß sie alle relevante Information für ein Beweisziel enthält.

2.2. Anforderungen des interaktiven Beweisens

Die Anforderungen, die ein interaktives Beweissystem an den Kalkül stellt, sind nicht exakt formalisierbar. Dies verwundert nicht, da man sich hier letztlich auch mit der Kommunikation zwischen Mensch und Maschine befaßt. Man kann sich allerdings auf Erfahrungen mit bestehenden interaktiven Beweissystemen (etwa KIV) stützen, und aus diesen Erfahrungen extrahieren, welche Eigenschaften des Kalküls sich als wichtig herausgestellt haben.

Der Grund für die Suche nach einem Kalkül für ein integriertes System besteht allerdings darin, daß die für das automatische Beweisen benutzten Kalküle verschiedene Eigenschaften haben, die sich beim interaktiven Gebrauch störend auswirken. Die Negation jener Eigenschaften ergibt offenbar Anforderungen an einen Kalkül für den interaktiven Gebrauch.

So können etwa folgende Anforderungen zusammengestellt werden:

- Das aktuelle Beweisziel muß dem Benutzer immer deutlich sein. Der Kalkül kann dies unterstützen, wie etwa im Sequenzenkalkül, bei dem jedes Blatt eines Beweisbaums ein Beweisziel darstellt und dieses Ziel durch die Sequenz vollständig gegeben ist.

- Die „Beweisgeschichte“ sollte dem Benutzer in einer Form zugänglich sein, die es ihm ermöglicht, Fehlentscheidungen zurückzuverfolgen und zu korrigieren. Aus einem geschlossenen Beweis sollte klar hervorgehen, welche Schritte ausgeführt wurden, um ihn zu erhalten.

- Mit Informationen, die aus der Sicht des Benutzers für den Beweis nicht wesentlich sind, sollte dieser auch nicht konfrontiert werden. Dazu gehören etwa Informationen, die der automatische Teil zur Suchraumeinschränkung verwendet: diese soll der Benutzer nur sehen, wenn es ihm darum geht, das Verhalten des Automaten zu verstehen. Daher sollten diese Informationen schon im Kalkül von den für den Benutzer wichtigen Informationen getrennt sein.

- Jeder im Kalkül mögliche Beweisschritt sollte für den Benutzer einsichtig sein. Er sollte das Ergebnis der Anwendung einer Beweisregel im wesentlichen vorhersehen können. Es mögen durchaus Regeln mit komplizierten Voraussetzungen, etwa Variablenbedingungen, existieren. Aber wenn der Benutzer intuitiv meint, eine Regel anwenden zu können, und dies in irgendeiner korrekten Weise möglich ist, sollte die Regel auch formal anwendbar sein und das Ergebnis der Intuition entsprechen. Nur so kann nämlich der Benutzer seine Beweisführung im voraus planen, also zielgerichtet arbeiten.

- Solange der Benutzer einen Beweis ohne den Automaten, also rein interaktiv führt, sollte er möglichst wenig von denjenigen Eigenheiten des Kalküls merken, die in erster Linie dem automatischen Beweisen dienen. Der Mensch sollte natürlich *auch* die in erster Linie für den Automaten vorgesehenen Mechanismen nützen können, wenn dies sinnvoll erscheint. Aber erst dann sollte er mit deren Spezifika konfrontiert werden.

Wie schon angedeutet, sind diese Anforderungen alle nicht formal exakt formulierbar oder überprüfbar. Die Brauchbarkeit eines bestimmten Kalküls läßt sich nur anhand einer Implementierung, also durch die praktische Anwendung bestätigen.

2.3. Anforderungen des automatischen Beweisens

Die Anforderungen an einen Kalkül für das automatische Beweisen sich natürlich anderer Art. Bei einem analytischen Nicht-Normalform-Kalkül sind in erster Linie folgende Eigenschaften wichtig:

- Der Kalkül muß effizient implementierbar sein. Insbesondere müssen Datenstrukturen gefunden werden, die die syntaktischen Konstrukte des Kalküls ökonomisch repräsentieren und schnelle Implementierungen der Regeln erlauben. Diese Anforderung ist hier viel wichtiger als bei den interaktiven Systemen. Dort ist es vertretbar, wenn die Anwendung einer Regel ein bis zwei Sekunden dauert. Für einen Automatischen Beweiser wäre dies völlig inakzeptabel, da ja erheblich mehr Regelanwendungen für die Beweissuche erforderlich sind, als der eigentliche Beweis benötigt.

- Der Kalkül muß eine Möglichkeit bieten, um Indeterminismen bei der Beweissuche zu vermeiden. Dies ist zum Beispiel die Rolle der freien Variablen im Tableaukalkül.

- Die speziellen Verfahren zur Gleichheitsbehandlung, die für automatische Beweissysteme existieren, sollten in dem Kalkül darstellbar sein. Dazu gehören vervollständigungs- und constraintbasierte Verfahren. Auch das Konzept der universellen Variablen wird hier besonders wichtig.

Der letzte Punkt ergibt sich aus der Forderung am Anfang dieses Kapitels: Bei einem reinen automatischen Beweiser kann es durchaus sinnvoll sein, Teilprobleme wie die Gleichheitsbehandlung im Hintergrund mit völlig anderen Formalismen zu behandeln. Für einen *homogenen, integrierten* Kalkül, wie er hier gesucht wird, ist dies unerwünscht.

2.4. Entwicklungsmethodik für einen integrierten Kalkül

Die Anforderungen aus den letzten beiden Abschnitten unterscheiden sich in ihrer Natur: die Anforderungen auf der automatischen Seite verlangen das Vorhandensein von Mechanismen, die relativ konkrete Probleme lösen. Für die interaktive Seite ist eher die Darstellung dieser Mechanismen wichtig.

Die Gliederung dieser Arbeit orientiert sich daher in erster Linie an den verschiedenen Problemlösungen, die im Bereich des automatischen Beweisens gefunden wurden. Es sollen Wege vorgeschlagen werden, die jeweiligen Probleme in einer Weise zu lösen, die sich an den in automatischen Beweisern erprobten Lösungen orientiert, aber mit den Anforderungen an einen interaktiven Gebrauch vereinbar ist.

Es soll versucht werden, die Vorschläge, die den einzelnen Problemen entsprechen, so weit voneinander unabhängig zu halten, daß es auch möglich ist, nur einen Teil derselben in einer Implementierung zu berücksichtigen. Auch wenn sich ein Teil der vorgeschlagenen Konzepte in der Praxis als unvorteilhaft erweisen sollte, wird dadurch sichergestellt, daß der Rest beibehalten werden kann.

Die diskutierten Problemlösungen sind im einzelnen:

- **Freie Variablen:** Diese dienen dazu, dem Automaten das Finden von Instantiierungen für Quantoren zu erleichtern. Im wesentlichen wird hier die Wahl einer Instanz auf den Zeitpunkt des Astabschlusses verschoben.

- **Skolemisierung:** Im Prinzip wird durch Skolemisierung einem Objekt, dessen Existenz bekannt ist, ein Name gegeben. Dies geschieht auch in einem Grundkalkül. In einem Kalkül mit freien Variablen ergeben sich jedoch zusätzliche Probleme.

- **Universelle Variablen und Formeln:** Diese sind eine Optimierung, die den Zweck hat,die Zahl der für einen Beweis benötigten Instantiierungen zu reduzieren. Sie sind insbesondere für eine effiziente Gleichheitsbehandlung von großer Bedeutung.

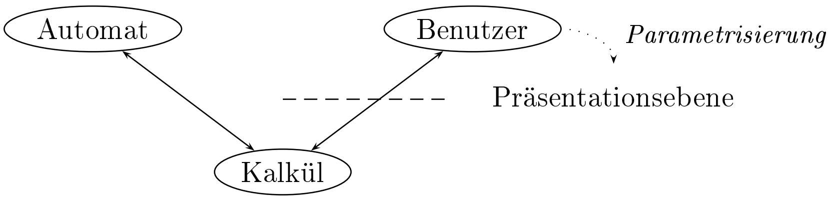
- **Gleichheitsbehandlung, Constraints:** Die Symmetrie der Gleichheit macht ihre automatische Behandlung sehr schwer. Hier sind die Verwendung von Constraints und vervollständigungsbasierte Verfahren wichtig.

Es gibt natürlich noch eine Reihe weiterer Optimierungen von analytischen Tableaux. Dazu zählen etwa Konnektions- und Regularitätsbedingungen. Diese Verfahren sollen hier nicht behandelt werden. Sie beziehen sich auf die Form von Beweisen, also die Reihenfolge, in der Regeln auf Formeln angewandt werden. Solche Verfahren betreffen nicht die Kalkülregeln selber, sondern sie schränken die Anwendung dieser Regeln ein. Eine Beweisprozedur (also ein Verfahren, das automatisch nach einem Beweis sucht) kann diese Einschränkungen auf einen hier vorgeschlagenen Kalkül anwenden. Zu überprüfen ist dann die Vollständigkeit dieser *Beweisprozedur*. Diese ist aber keine Eigenschaft des *Kalküls*.

Auch das als Lemmagenerierung bezeichnete Verfahren und ähnliche Varianten von Tableauregeln werden nicht näher betrachtet, da sie ohne weiteres auf die vorgestellten Konzepte übertragbar sein sollten.

Die zuerst genannte Anforderung an einen Kalkül für automatische Beweiser, nämlich die Effizienz, steht zunächst nicht im Mittelpunkt. Der Entwurf einer effizienten Implementierung muß genau berücksichtigen, welche von den vorgeschlagen Konzepten übernommen werden sollen. Es wäre also nicht möglich, hier die Orthogonalität der verschiedenen Ansätze zu wahren. In Anhang B sollen trotzdem einige Betrachtungen zur effizienten Implementierbarkeit von ϵ-Termen angestellt werden.

Abbildung 2.1.: Rolle der Präsentationsebene



2.5. Die Präsentationsebene

Die in dieser Arbeit vorgestellten Konzepte führen teilweise zu Kalkülen, die für die direkte Verwendung durch den Benutzer zu unhandlich sind. Beim Betrachten des Beweises auf Seite 31 gewinnt man zum Beispiel nicht den Eindruck, der Sequenzen-Kalkül sei zum interaktiven Beweisen besonders geeignet: So viel redundante Information lenkt vom eigentlichen Beweisgeschehen ab. Die Erfahrung mit KIV zeigt jedoch, daß der Sequenzenkalkül sehr wohl geeignet ist: Der Benutzer sieht im Normalfall nur *eine* Sequenz, in der alle Formeln dargestellt sind, die für das aktuelle Beweisziel relevant sind. Die Darstellung des Kalküls für den Benutzer ist also viel übersichtlicher als die Darstellung „auf dem Papier“, da zwischen Kalkül und Benutzer durch eine *Präsentationsebene* vermittelt wird, siehe Abb. 2.5. Der Benutzer hat die Möglichkeit, die Präsentation zu parametrisieren, um den von ihm gewünschten Abstraktionsgrad einzustellen.

Da zur Lösung der identifizierten Probleme oft Mechanismen nötig sind, die die Benutzung des Kalküls für den Benutzer erschweren würden, werden bei der Einführung dieser Mechanismen jeweils auch Möglichkeiten zur Gestaltung der Präsentation diskutiert.

3. Grund-Sequenzenkalkül und Tableaukalkül mit freien Variablen

Trotz aller Ähnlichkeiten zwischen Sequenzen- und Tableaukalkül gibt es zwischen typischen Ausprägungen einige subtile Unterschiede, die für die folgenden Abschnitte von Bedeutung sind, und häufig zum Anlaß von Verwechslungen und Mißverständnissen werden, wenn es um einen Vergleich der beiden Kalküle geht.

Wenn in diesem und den folgenden Abschnitten zwischen Sequenzen- und Tableaukalkül unterschieden wird, so geht es, sofern nichts anderes gesagt wird, *nicht* um die Darstellung in Sequenzen- oder Tableau*schreibweise*: Der Sequenzenkalkül wird stellvertretend für einen Grundkalkül betrachtet, der Tableaukalkül für einen Kalkül mit freien Variablen im Sinne des automatischen Beweisens.

3.1. Sequenzenkalkül

Im Sequenzenkalkül werden Beweisziele durch *Sequenzen* $\Gamma \vdash \Delta$ dargestellt. Γ und Δ sind dabei Mengen[1] von nicht notwendigerweise geschlossenen prädikatenlogischen Formeln. Ist eine solche Sequenz als Beweisziel gegeben, so soll aus der Konjunktion der Formeln in Γ die Disjunktion der Formeln in Δ abgeleitet werden. Man definiert also folgende Semantik: Eine Sequenz $\Gamma \vdash \Delta$ ist in einer prädikatenlogischen Struktur $\mathcal{S}$ unter einer Variablenbelegung β gültig, falls $\mathcal{S}, \beta \not\models \phi$ für eine Formel $\phi \in \Gamma$, oder $\mathcal{S}, \beta \models \psi$ für eine Formel $\psi \in \Delta$. Eine Sequenz heißt *allgemeingültig*, falls sie in *allen* Strukturen $\mathcal{S}$ unter *allen* Variablenbelegungen β gültig ist.

Ein Sequenzenbeweis ist ein Baum, der in jedem Knoten eine Sequenz enthält. Die Wurzel wird in der graphischen Darstellung üblicherweise unten dargestellt. Um die Allgemeingültigkeit einer Formel ϕ nachzuweisen, fängt man mit einer Sequenz $\vdash \phi$ als Wurzel an. Der Beweis wird erweitert durch Anwendung von Regeln (d.h. Instanzen von Regelschemata) der Form

$$\frac{\Gamma_1 \vdash \Delta_1 \quad \ldots \quad \Gamma_m \vdash \Delta_m}{\Gamma \vdash \Delta}$$

die folgende Korrektheitseigenschaft haben:

[1] Klassischerweise verwendet die Definition Listen statt Mengen. Für unsere Zwecke kommt es allerdings auf die Reihenfolge oder mehrfaches Auftreten von Formeln nicht an. Um die Notation übersichtlich zu halten, werden die Mengenklammern weggelassen, und die Formeln links und rechts vom Symbol $\vdash$ durch Kommata getrennt.

Sind alle Prämissen $\Gamma_i \vdash \Delta_i$ *allgemeingültig, so ist auch die Konklusion* $\Gamma \vdash \Delta$ *allgemeingültig.*

Geschlossen werden Sequenzenbeweise durch das *Axiom*

$$\frac{}{\phi, \Gamma \vdash \phi, \Delta}\ ax$$

Sequenzen dieser Form sind offenbar allgemeingültig. Eine einfache Induktion über die Baumstruktur des Beweises mit der angegebenen Korrektheitseigenschaft ergibt die Allgemeingültigkeit der Sequenz in der Wurzel, und damit der Formel ϕ.

Die erwähnten subtilen Unterschiede betreffen hauptsächlich die Semantik der freien Variablen, sowie die damit zusammenhängende Behandlung von Quantoren. Die folgenden Absätze sind inhaltlich nicht sehr anspruchsvoll. Ihr Zweck ist es, durch Hervorhebung verschiedener Details, sowie durch angedeutete Korrektheitsbeweise, die Entsprechung zwischen den Regeln im Tableaukalkül und denen im Sequenzenkalkül deutlich herauszustellen.

Allquantoren[2] in der linken Hälfte der Sequenz (also in Γ) werden mit Hilfe des folgenden Regelschemas behandelt:

$$\frac{[x/t](\phi), \forall x.\phi, \Gamma \vdash \Delta}{\forall x.\phi, \Gamma \vdash \Delta}\ \forall l$$

wobei t ein beliebiger Term ist, und $[x/t](\phi)$ entsteht, indem t in ϕ für x substituiert wird. Zur Anwendung der Regel muß dieser Term ausgewählt werden. Durch ihn wird angegeben, welche Instanz der allquantifizierten Formel benötigt wird. Zum Beweis der angegebenen Korrektheitseigenschaft für diese Regel ist im wesentlichen zu zeigen, daß falls $\mathcal{S}, \beta \models \forall x.\phi$, dann auch $\mathcal{S}, \beta \models \phi^t_x$. Dies ist mit Hilfe der Semantik des Allquantors und dem Substitutionslemma der Prädikatenlogik leicht zu zeigen.

Für Existenzquantoren links gibt es die Regel

$$\frac{[x/y](\phi), \Gamma \vdash \Delta}{\exists x.\phi, \Gamma \vdash \Delta}\ \exists l$$

wobei y eine Variable ist, die in der Sequenz noch nicht frei vorkommt.[3] Im weiteren Beweis bezeichnet diese Variable das Element, dessen Existenz durch die Formel $\exists x.\phi$ zugesichert wird. Die Korrektheit ergibt sich hier aus folgendem Argument: Angenommen, die obere Sequenz sei allgemeingültig, die untere aber nicht. Dann existieren eine Struktur $\mathcal{S} = (\mathcal{D}, \mathcal{I})$, sowie eine Belegung β, die die untere Sequenz nicht erfüllen. Es gilt also $\mathcal{S}, \beta \models \exists x.\phi$, sowie $\mathcal{S}, \beta \models \psi$ für alle $\psi \in \Gamma$ und $\mathcal{S}, \beta \not\models \psi$ für alle $\psi \in \Delta$. Also existiert ein Element $d \in \mathcal{D}$ der Trägermenge mit $\mathcal{S}, \beta\{x \leftarrow d\} \models \phi$, und durch Variablenumbenennung $\mathcal{S}, \beta\{y \leftarrow d\} \models [x/y](\phi)$. In der Struktur $\mathcal{S}$ und unter der *modifizierten* Belegung $\beta\{y \leftarrow d\}$ ist damit aber die obere Sequenz ungültig, womit der Widerspruch hergestellt ist.

[2]Im Sequenzenkalkül gibt es zu fast jeder Regel eine duale Variante. Die Regel für Existenzquantoren rechts ist zum Beispiel völlig analog. Zur Diskussion der zugrundeliegenden Prinzipien reicht es, jeweils immer nur eine der beiden Regeln zu betrachten.

[3]Falls x in der Ausgangssequenz nicht frei vorkommt, wird man natürlich $y = x$ verwenden, um das Einführen neuer Variablen zu vermeiden. Im allgemeinen kann man darauf aber nicht verzichten.

Aus der eingangs angegebenen Semantik folgt leicht, daß eine Sequenz $\phi_1, \ldots, \phi_m \vdash \psi_1, \ldots, \psi_n$ genau dann allgemeingültig ist, wenn die abgeschlossene prädikatenlogische Formel

$$\forall x_1. \ldots \forall x_k.((\phi_1 \wedge \ldots \wedge \phi_m) \rightarrow (\psi_1 \vee \ldots \vee \psi_n))$$

allgemeingültig ist, wobei $x_1, \ldots, x_k$ die freien Variablen in den Formeln ϕ_i und ψ_i der Sequenz seien. Kommen also im Sequenzenkalkül freie Variablen vor, so sind diese als implizit über die Sequenz allquantifiziert anzusehen. Man sagt auch, die freien Variablen im Sequenzenkalkül seien *universell*.

Es ist ohne weiteres möglich (allerdings nicht üblich), nur mit geschlossenen Formeln zu arbeiten: Man kann nämlich freie Variablen einfach der Signatur als Konstantensymbole hinzufügen. Die Regel $\exists l$ führt dann statt einer Variable ein neues Konstantensymbol ein, erweitert also die Signatur. Dies zu tun ist möglich, weil in der Definition der allgemeingültigen Sequenz über die Belegung, ebenso wie über die Strukturen, universell quantifiziert wird. Macht man die freien Variablen zu Konstanten, so wird die Belegung unnötig. Sie wird ersetzt durch die Interpretation der entsprechenden Konstantensymbole in der prädikatenlogischen Struktur.

3.2. Tableaukalkül

Auch Tableaubeweise haben Baumstruktur, wobei es hier allerdings üblich ist, die Wurzel am oberen Ende darzustellen. Die Knoten enthalten prädikatenlogische Formeln.[4] Ein Tableau heißt *erfüllbar*, falls es eine Struktur $\mathcal{S}$ gibt, so daß für *alle* Belegungen β ein Ast existiert, so daß alle Formeln auf dem Ast unter $\mathcal{S}, \beta$ gültig sind.

Der Tableaukalkül ist ein Widerlegungskalkül. Um die Allgemeingültigkeit einer prädikatenlogischen Formel ϕ zu zeigen, muß diese geschlossen sein. Gegebenenfalls wird der universelle Abschluß der Formel genommen, der genau dann allgemeingültig ist, wenn die ursprüngliche Formel es ist. Die geschlossene Formel wird negiert, und die Unerfüllbarkeit der Negation gezeigt. Dazu beginnt man mit einem Tableau, das nur aus der Wurzel mit dieser negierten Formel besteht. Das Tableau wird aufgebaut entweder durch Erweiterungsregeln der Form

$$\frac{\phi}{\begin{array}{ccc} \phi_{11} & & \phi_{l1} \\ \vdots & \ldots & \vdots \\ \phi_{1k_1} & & \phi_{lk_l} \end{array}}$$

wobei ϕ eine Formel ist, die in einem Ast des Tableaus vorkommt, oder durch Anwendung von Substitutionen auf das ganze Tableau. Die Korrektheitseigenschaft lautet hier:

> *Wird auf ein erfüllbares Tableau eine Regel angewandt, so ist das resultierende Tableau weiterhin erfüllbar.*

Ein Tableaubeweis ist geschlossen, falls jedes Ast ein Paar von komplementären Literalen $\phi, \neg\phi$ enthält. Ein solches Tableau ist trivialerweise unerfüllbar, da in jeder Struktur unter jeder Belegung jeder Ast mindestens eine ungültige Formel enthält. Durch Induktion mit der Korrektheitsbedingung folgt, daß dann das initiale Tableau, das nur die Negation der zu beweisenden Formel enthielt, ebenfalls unerfüllbar sein mußte. Da die Formel als abgeschlossen vorausgesetzt wurde, folgt daraus deren Allgemeingültigkeit.

[4] Häufig werden signierte Formeln verwendet. Das ist für diese Betrachtung nicht nötig.

Um den Vergleich zum Sequenzenkalkül besser ziehen zu können, bietet es sich an, die Korrektheitsbedingung so umzuformulieren, daß mit der Unerfüllbarkeit gearbeitet wird, die ja der Allgemeingültigkeit im Sequenzenkalkül entspricht. Kontraposition liefert: Ein Tableau ist *unerfüllbar*, falls für alle Strukturen $\mathcal{S}$ eine Belegung β *existiert*, so daß *jeder* Ast eine unter $\mathcal{S}, \beta$ ungültige Formel enthält. Die entsprechende Umformulierung der Korrektheitseigenschaft der Regeln verlangt, daß aus der Unerfüllbarkeit des erweiterten Tableaus die Unerfüllbarkeit des ursprünglichen Tableaus folgt. Diese Formulierung unterscheidet sich von Definition allgemeingültiger Sequenzen primär durch die existenzielle Quantisierung der Belegung.

Die Regel für Allquantoren lautet

$$\frac{\forall x.\phi}{[x/X](\phi)}\ \gamma$$

wobei X eine Variable ist, die im Tableau noch nicht vorkommt. Für den Nachweis der Korrektheitseigenschaft wird im wesentlichen die Semantik des Allquantors ausgenutzt.

Diese Regel entspricht *nicht* der Regel $\exists l$, die zwar auch einen Quantor von der Formel entfernt und eine neue Variable einführt, aber semantisch eine völlig andere Bedeutung hat. Die γ-Regel entspricht vielmehr der Regel $\forall l$ des Sequenzenkalküls, wobei allerdings der eingesetzte Term offengelassen wird. Wenn später Substitutionen auf das Tableau angewandt werden, wird dieser Term gewissermaßen konkretisiert. Die Korrektheit dieser Substitution von allen Vorkommen einer freien Variablen im Tableau ergibt sich aus der Quantisierung über *alle* Belegungen in der Definition eines erfüllbaren Tableaus, beziehungsweise aus der existenziellen Quantisierung in der Charakterisierung der Unerfüllbarkeit. Die eingeführte freie Variable bezeichnet im übrigen im ganzen Tableau das gleiche Objekt. Das äußert sich schon daran, daß Substitutionen immer auf das ganze Tableau angewandt werden müssen. Im Gegensatz dazu ist der Skopus von freien Variablen im Sequenzenkalkül jeweils auf eine Sequenz beschränkt.

Es gibt mehrere Varianten der Regel für Existenzquantoren. Dies wird in Abschnitt 5 noch von Bedeutung sein. Im Moment genügt es, die folgende, sogenannte δ^+-Regel zu betrachten:

$$\frac{\exists x.\phi}{[x/f(X_1,\ldots,X_n)](\phi)}\ \delta^+$$

wobei f ein neues Funktionssymbol, und $X_1,\ldots,X_n$ die in $\exists x.\phi$ frei vorkommenden Variablen sind. Die Korrektheitseigenschaft wird nachgewiesen, indem aus einem Modell für das ursprüngliche Tableau (das f nicht interpretiert) eine Interpretation für f konstruiert wird, die für jede Belegung von $X_1,\ldots,X_n$ ein x liefert, so daß ϕ wahr wird, sofern ein solches existiert. Mit dem so *modifizierten* Modell wird das erweiterte Tableau erfüllbar.

Die δ^+ Regel entspricht der $\exists l$ Regel des Sequenzenkalküls, wenn man die am Ende von 3.1 erwähnte Variante mit Konstanten statt freien Variablen betrachtet: Dann existieren im Sequenzenkalkül überhaupt keine freien Variablen im Sinne des Tableaukalküls, deswegen entfällt die bei letzterem notwendige Parametrisierung des hinzugefügten Signatursymbols, und ein Konstantensymbol genügt.

Man kann sich fragen, was geschieht, wenn man einen Tableaubeweis mit einer nicht geschlossenen Formel beginnt. Es zeigt sich, daß ein Tableau mit nur einer Formel genau dann erfüllbar ist, wenn das Tableau mit dem All-Abschluß dieser Formel erfüllbar ist. Da das Tableau mit der Negation einer Formel begonnen wird, deren Allgemeingültigkeit zu zeigen ist, verhalten sich die freien Variablen in dieser, als seien sie existentiell quantifiziert. Deshalb werden freie Variablen im Tableaukalkül als *existentiell* bezeichnet.

Zusammenfassend sei also festgestellt: Die namensgebenden freien Variablen in einem Tableaukalkül mit ebensolchen verhalten sich semantisch völlig anders als die freien Variablen im Sequenzenkalkül.

4. Metavariablen statt freier Variablen

Da der größte Unterschied zwischen dem Sequenzenkakül, wie er etwa in KIV verwendet wird, und einem Tableaukalkül für das automatische Beweisen, wie im letzten Abschnitt erläutert, in der Semantik der freien Variablen liegt, soll dieser Aspekt eines integrativen Kalküls zuerst betrachtet werden.

4.1. Die Problematik der freien Variablen

Der Grund für die existentielle Semantik der freien Variablen beim Automaten-Kalkül liegt im Indeterminismus der vorgestellten Regel $\forall l$ des interaktiven Kalküls. Der einzige Weg, wie ein automatischer Beweiser mit dieser Regel arbeiten könnte, bestünde darin, alle Terme aufzuzählen, und die Regel für jeden anzuwenden. Indem die Instantiierung der freien Variablen auf den Astabschluß verschoben wird, verschwindet der Indeterminismus zwar nicht völlig, wird aber doch deutlich verringert. Ein Kalkül für integriertes Beweisen muß jedenfalls ähnliche Möglichkeiten bieten.

Dem menschlichen Beweiser dürfte der Verzicht auf die freien Variablen mit *universeller* Semantik nicht allzu schwer fallen: Ob die $\exists l$ Regel nun neue Konstantensymbole oder neue Variablen einführt, macht für Benutzung und Verständnis keinen großen Unterschied. Problematischer ist der Effekt der verzögerten Instantiierung. Ein Blick auf die Arbeitsweise eines menschlichen Beweisers zeigt nämlich, daß dieser zumeist keine Probleme mit der Anwendung der Instantiierungsregel $\forall l$ hat: Da er seine Beweisführung plant, weiß er meistens genau, welche Instanzen er benötigt. Würde man dem Benutzer einfach eine Regel wie die γ-Regel des Tableaukalküls zur Verfügung stellen, müßte dieser sich entweder die intendierte Instantiierung bis zum Abschluß des Beweises merken, oder aber sofort die entsprechende Substitution auf den Beweis anwenden.

Die globale Substitutionsregel, die einen Beweis ändert, indem alle Instanzen einer oder mehrerer freier Variablen im gesamten Beweis ersetzt werden, ist aber ebenfalls unvorteilhaft für einen integrierten Kalkül. In Abschnitt 2.2 wurde gefordert, daß aus einem Beweis die Beweisgeschichte hervorgeht, um die Korrektur von eventuellen Fehlentscheidungen zu erleichtern, und den fertigen Beweis nachvollziehbar zu halten. Die Anwendung einer Substitution auf einen gesamten Beweis ist aber eine destruktive Operation. Auch hier muß also eine Alternative gefunden werden.

Ebenso scheidet natürlich die Möglichkeit aus, den Automaten eine Beweissuche mit freien Variablen durchführen zu lassen, und nach gefundener abschließender Substitution dem Benutzer den entstandenen Grund-Beweis zu präsentieren. Das würde dem Gedanken eines gemeinsamen Kalküls widersprechen: der Benutzer würde nur eine Übersetzung der Struktur erhalten, auf der der Automat arbeitet.

Die einfachste Lösung wäre die Bereitstellung von *zwei* Instantiierungsregeln: eine für den Automaten und eine für den Menschen. Das Ergebnis wäre dann allerdings kaum mehr als *homogen* zu bezeichnen.

Die im Rest dieses Abschnitts vorgeschlagene Lösung betrifft nicht nur die hier angesprochenen Probleme, sondern hat auch positive Auswirkungen bezüglich der anderen Anforderungen aus 2.2.

4.2. Metavariablen

Formal werden freie Variablen im Tableaukalkül normalerweise mittels einer Variablenbelegung interpretiert, die bei einer gegebenen prädikatenlogischen Struktur jeder Variablen ein Element der Trägermenge zuordnet. In 3.2 wurde erläutert, daß für die Unerfüllbarkeit eines Tableaus – also den Abschluß eines Beweises – für jede Interpretation eine Belegung existieren muß, die alle Äste ungültig macht, also alle Äste abschließt. Wie wird aber die Existenz einer solchen Belegung für jede Interpretation gezeigt?

Es ist möglich, die Anwendung von Substitutionen auf den Beweis bis ans Ende zu verschieben, die Substitutionen also zu sammeln. So erhält man eine einzige Substitution, die alle Äste gleichzeitig abschließt. Beschränkt man sich hier auf Grundsubstitutionen,[1] so liefert diese abschließende Substitution für jede Variable einen Grundterm. Die Auswertung dieser Grundterme ergibt dann für jedes Modell eine Belegung der freien Variablen, die alle Äste ungültig macht. Es ist also die abschließende Substitution, die für jede Interpretation die benötigte Belegung liefert.

Freilich kann eine Substitution nur solche Belegungen angeben, die jeder Variable einen Wert der Trägermenge zuordnen, der Auswertung eines Grundterms ist. Das ist im allgemeinen nicht für jede Belegung der Fall. Die Vollständigkeit des Tableaukalküls besagt aber unter anderem, daß solche Belegungen zum Abschluß von Beweisen ausreichen.

Da nun aber die Behandlung von freien Variablen immer nur durch die Vermittlung von Substitutionen geschieht, ist es möglich, auf Belegungen und freie Variablen im bisherigen Sinne völlig zu verzichten. Dazu führt man die neue syntaktische Sorte der *Metavariablen* ein. Diese sollen disjunkt sein von den Objektvariablen, die von Quantoren gebunden werden.[2] Metavariablen werden als Platzhalter für Terme behandelt. Die Beweisregeln werden so formuliert, daß nur Metavariablen frei in Formeln vorkommen. Umgekehrt darf nicht über Metavariablen quantifiziert werden. Wird auf eine solche Formel eine Grundsubstitution der Metavariablen angewandt, entsteht eine geschlossene Formel ohne Metavariablen. Deren Semantik in einer prädikatenlogischen Struktur ist dann unabhängig von einer Belegung.

In diesem Abschnitt soll auf eine formale Darstellung dieses Ansatzes verzichtet werden. In Abschnitt 6.4 werden Korrektheit und Vollständigkeit eines Kalküls gezeigt, der mit Metavariablen arbeitet. Die Verwendung von Metavariablen in einem Kalkül für das integrierte Beweisen hat unter anderem folgende Vorteile:

- Es lassen sich relativ natürlich Constraints in den Kalkül einbetten. Dies sind syntaktische Einschränkungen an die Instantiierung von Metavariablen, die für die

[1] Dies ist möglich, da jede Verfeinerung einer abschließenden Substitution den Beweis immer noch abschließt

[2] Sie werden zur besseren Unterscheidung im folgenden mit Großbuchstaben bezeichnet

Gleichheitsbehandlung von größter Wichtigkeit sind. Sie beziehen sich auf Substitutionen für die in einer Formel frei vorkommenden Variablen, und ihre Semantik ist viel leichter zu definieren, wenn diese Variablen von vornherein Terme bezeichnen, also Metavariablen sind.

- Das Konzept der Metavariablen als Platzhalter für Terme ist näher an der menschlichen Sicht- und Arbeitsweise als die implizit existentiellen freien Variablen. Was der Mensch bei der Instantiierung eines Quantors eingibt, sind keine Belegungen, sondern Substitutionen.

- In der formalen Definition des Kalküls tritt eine deutlichere Trennung zwischen den lokal gültigen quantifizierten Variablen und den globalen, also zwischen mehreren Ästen vermittelnden Metavariablen, hervor, als dies beim Tableaukalkül sonst der Fall ist.

- Wenn freie Variablen in einem automatischen Beweissystem letztlich „instantiiert“ werden, so geschieht dies beim Astabschluß durch Unifikation. Auch hier wird also eine Substitution gesucht.

- Auch beim Hypergraphenkalkül (siehe Kapitel 8) bieten Metavariablen eine elegante Möglichkeit zur Verschiebung von Quantorinstantiierungen.

Dies sind alles angenehme Eigenschaften, die für die Verwendung von Metavariablen sprechen. Die ursprüngliche Frage nach der Präsentation des Kalküls für den Benutzer ist jedoch immer noch offen. Es wird sich allerdings zeigen, daß auch hier eine Möglichkeit besteht, deren Formulierung vom Konzept der Metavariablen profitiert.

4.3. Beispiel

Zur Verdeutlichung der Vorteile von Metavariablen, sowie zur Einleitung der im nächsten Abschnitt vorgestellten Strukturierung eines Beweissystems mit Metavariablen, soll ihre Benutzung hier kurz anhand eines Beispiels gezeigt werden. Die $\forall l$-Regel zur Instantiierung von Quantoren in einem Sequenzenkalkül mit Metavariablen lautet:

$$\frac{[x/X](\phi), \forall x.\phi, \Gamma \vdash \Delta}{\forall x.\phi, \Gamma \vdash \Delta} \; \forall l$$

wobei X eine Metavariable ist, die im Beweis noch nicht vorkommt. Ein Beweis ist genau dann geschlossen, wenn eine Grundsubstitution der Metavariablen existiert, unter der alle Äste geschlossen sind.

Es soll nun ein Beweis für die Formel

$$(\forall x.p(x, a) \land \forall y, z.(p(y, z) \to p(z, y))) \to p(a, b)$$

geführt werden. Dabei werden nicht mehr benötigte $\forall$-Formeln weggelassen, um den Beweis übersichtlicher zu machen. Mit den üblichen aussagenlogischen Regeln des Sequenzenkalküls und den neuen Quantorregeln ergibt sich folgende Ableitung:

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{p(X,a) \vdash p(Y,Z), p(a,b) \qquad p(X,a), p(Z,Y) \vdash p(a,b)}{p(X,a), p(Y,Z) \to p(Z,Y) \vdash p(a,b)}\,{\to}l}{p(X,a), \forall z.(p(Y,z) \to p(z,Y)) \vdash p(a,b)}\,\forall l}{p(X,a), \forall y,z.(p(y,z) \to p(z,y)) \vdash p(a,b)}\,\forall l}{\forall x.p(x,a), \forall y,z.(p(y,z) \to p(z,y)) \vdash p(a,b)}\,\forall l}{\forall x.p(x,a) \wedge \forall y,z.(p(y,z) \to p(z,y)) \vdash p(a,b)}\,\wedge l}{\vdash (\forall x.p(x,a) \wedge \forall y,z.(p(y,z) \to p(z,y))) \to p(a,b)}\,{\to}r$$

Dieser Beweis ist geschlossen mit der Grundsubstitution $\{X/b, Y/b, Z/a\}$, da nach dieser Substitution im linken Ast $p(b,a)$ links und rechts des $\vdash$ Zeichens auftaucht, und $p(a,b)$ im rechten Ast. Wendet man diese Substitution auf den ganzen Beweis an, erhält man:

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{p(b,a) \vdash p(b,a), p(a,b) \qquad p(b,a), p(a,b) \vdash p(a,b)}{p(b,a), p(b,a) \to p(a,b) \vdash p(a,b)}\,{\to}l}{p(b,a), \forall z.(p(b,z) \to p(z,b)) \vdash p(a,b)}\,\forall l}{p(b,a), \forall y,z.(p(y,z) \to p(z,y)) \vdash p(a,b)}\,\forall l}{\forall x.p(x,a), \forall y,z.(p(y,z) \to p(z,y)) \vdash p(a,b)}\,\forall l}{\forall x.p(x,a) \wedge \forall y,z.(p(y,z) \to p(z,y)) \vdash p(a,b)}\,\wedge l}{\vdash (\forall x.p(x,a) \wedge \forall y,z.(p(y,z) \to p(z,y))) \to p(a,b)}\,{\to}r$$

Dies ist der Beweis, der sich mit der $\forall l$-Regel des Grund-Sequenzenkalküls ergeben hätte. Die Anwendung einer anderen Substitution für die Metavariablen würde ebenfalls zu einem korrekten Beweisbaum führen. Bei diesem wären jedoch nicht alle Äste geschlossen.

4.4. Trennung von Beweisbaum und Substitution

Der eingeführte Sequenzenkalkül mit Metavariablen unterscheidet sich zunächst kaum vom Tableaukalkül mit freien Variablen. Lediglich die Sichtweise und Bezeichnung der freien Variablen hat sich geändert. Zum automatischen Beweisen wird dieser Kalkül also bestens geeignet sein. Bei der interaktiven Benutzung bleiben die am Anfang dieses Kapitels erwähnten Probleme. Das am Beispiel gezeigte Verhältnis zwischen dem Sequenzenkalkül mit Metavariablen und einem Sequenzenkalkül mit sofortiger Instantiierung, wie er für den menschlichen Beweiser vorteilhaft ist, legt folgendes Konzept nahe:

Ein Beweis bestehe aus einem Beweisbaum *und* einer Grundsubstitution. Der Beweis ist geschlossen, falls nach Anwendung der Substitution jeder Ast des Baums geschlossen ist. Der Benutzer manipuliert beide Strukturen: Erstens den Beweisbaum, durch die Anwendung von Regeln, oder auch durch Abschneiden von Ästen, falls sich diese als fehlerhaft herausstellen. Zweitens wird ihm ein Mechanismus zur Verfügung gestellt, um die für seinen Beweisplan interessanten Substitutionen auf den Metavariablen zu beschreiben. Instantiiert er einen Quantor, kann er angeben, welche Instantiierung er für die Variable intendiert. Auch diese Beschreibung relevanter Grundsubstitutionen soll er jederzeit ändern können. In welcher Form dies geschehen kann, wird in Abschnitt 4.5 erörtert.

Bei dieser Sichtweise ist es sinnvoll, auf die im Sequenzenkalkül üblichen Axiome zum Abschluß von Beweisästen zu verzichten. Im klassischen Sequenzenkalkül wird ein Ast durch Anwendung eines Axioms geschlossen. Ein Axiom wird dabei als Regel ohne

Prämissen aufgefaßt. Bei einer Trennung von Beweisbaum und Substitution werden diese Axiome nicht angewandt. Ein Ast ist hier nur bezüglich einer bestimmten Instantiierung der Metavariablen abgeschlossen, und zwar genau dann, wenn auf die Sequenz, *nach* Anwendung der Substitution, ein Axiom anwendbar *wäre*. Etwa ist ein Ast $p(X, a) \vdash p(Y, Z)$ unter Substitutionen σ mit $\sigma(X) = \sigma(Y)$ und $\sigma(Z) = a$ geschlossen, da unter genau diesen Substitutionen auf die Sequenz $\sigma(p(X, a)) \vdash \sigma(p(Y, Z))$ das Axiom $\phi, \Gamma \vdash \phi, \Delta$ anwendbar ist. Eine direkte Anwendung des Axioms, also ohne Substitutionen auf dem Beweisbaum vorzunehmen, wäre nicht möglich. Insgesamt ist diese Formulierung des Astabschlusses näher an der Sichtweise des Tableaukalküls.

Die vom Benutzer angegebenen Einschränkungen an die Metavariablen haben zweierlei Funktion, nämlich die Steuerung der Beweisdarstellung für den Benutzer einerseits, sowie die Steuerung des automatischen Beweisers durch Suchraumeinschränkung andererseits:

- Wenn das Beweissystem den Beweisbaum für den Benutzer anzeigt, wird es, soweit möglich, auf dessen Angaben eingehen, und etwa anstelle einer Metavariablen einen für diese angegebenen Term anzeigen. Wenn der Benutzer *keine* ausreichend konkreten Angaben zur Instantiierung einer Variablen macht, oder diese vom Automaten eingeführt wurde, so kann einfach die Variable angezeigt werden.
- Bei der Suche nach abschließenden Grundsubstitutionen werden nur solche berücksichtigt werden, die seinen Angaben entsprechen. Auch bei der automatischen Anwendung von Regeln wie der Gleichheitseinsetzung werden nur solche Instantiierungen berücksichtigt.

Dieses Vorgehen führt dazu, daß der Benutzer mit dem Kalkül arbeiten kann wie mit einem klassischen Sequenzenkalkül: er kann Instanzen für Quantoren explizit angeben und bekommt diese im Beweisbaum auch angezeigt. Auch der Automat verhält sich so, als wären tatsächlich konkrete Terme eingesetzt worden. Die freie Variable bleibt auf der *Kalkülebene* unangetastet. Von der Präsentationsebene *kann* sie in einer Weise dargestellt werden, die der Intention des Benutzers entspricht.

Weitere Vorteile können sich spezifisch für ein integratives System ergeben, wenn man etwa dem Benutzer die Möglichkeit gibt, Instantiierungen einzuschränken, zu denen die Metavariablen noch gar nicht eingeführt wurden. Auf diesem Wege wäre es möglich, dem Automaten suchraumeinschränkende Informationen zur Verfügung zu stellen, die ihm die Suche nach der richtigen Instantiierung vereinfachen. Auf derartige Aspekte der Mensch-Maschine-Kooperation soll hier jedoch nicht näher eingegangen werden.

Außerdem hat das Konzept der Trennung von Beweisbaum und Substitutionseinschränkung einen großen Vorteil: Die beiden Komponenten können separat voneinander geändert, bzw. zurückgesetzt werden. Selbst wenn der Benutzer für einen Quantor eine konkrete Instanz angegeben hat, kann er diese nachträglich (etwa im Falle eines Fehlers) ändern, ohne dadurch den Beweisbaum zu invalidieren. Eine Änderung der Substitutionseinschränkung kann lediglich zur Folge haben, daß danach einige Äste nicht mehr geschlossen sind, die es vorher waren. Diese Möglichkeit einer getrennten Rücksetzbarkeit von Instantiierungsentscheidungen ist für das interaktive Beweisen neu. Wie nützlich diese Möglichkeit in der Praxis ist, kann nur anhand einer Implementierung festgestellt werden.

Auf die Fragen der Beweispräsentation wird im folgenden Abschnitt auf informaler Ebene eingegangen. Eine ausführlichere Diskussion erfolgt in Abschnitt 6.6. Zur Steuerung des automatischen Beweisers durch die vom Benutzer angegebene Einschränkung an die abschließende Substitution, siehe Abschnitt 6.7.

4.5. Einschränkungen an die Substitution

Während es wohl erprobte Methoden gibt, dem Benutzer die Manipulation von Beweisbäumen in übersichtlicher Weise zu ermöglichen, ist noch unklar, welcher Art die vom Benutzer anzugebenden Einschränkungen an die abschließende Substitution sein sollen.

In Abschnitt 6 wird ausführlich das Konzept der Constraints behandelt. Constraints werden dort als formales Mittel benutzt, um Mengen von Grundsubstitutionen zu beschreiben, und sind für die automatische Behandlung der Gleichheit unverzichtbar. Es liegt also nahe, sie außerdem auch zur Einschränkung der abschließenden Substitution zu verwenden. Dieser Ansatz führt zu einer sehr zwanglosen Einbettung in das Gesamtbild: In einem Kalkül mit Constraints, wie er im folgenden erarbeitet wird,[3] sind die Formeln in den Sequenzen mit Constraints versehen, die angeben, unter welchen Substitutionen für die Metavariablen diese Formeln zum Astabschluß verwendet werden dürfen. Zum Beweisabschluß muß also eine Substitution der Metavariablen gefunden werden, unter der jeder Ast des Beweisbaums geschlossen wird, und zwar durch Formeln, deren Constraint von dieser Substitution auch erfüllt wird. Dazu müssen in jedem Blatt potentiell abschließende Formeln ausgewählt werden. Dann schließt eine Grundsubstitution den Beweis ab, falls sie die Constraints von allen ausgesuchten Formeln erfüllt. Zu überprüfen ist also insgesamt, ob eine solche Substitution existiert, das heißt, ob die Konjunktion der zu den Formeln gehörigen Constraints erfüllbar ist, siehe Satz 6.7.

Wenn nun der Benutzer eine Einschränkung der abschließenden Grundsubstitution in Form eines Constraints angibt, so muß dieser Constraint einfach konjunktiv zu denjenigen hinzugefügt werden, die von den abschließenden Formeln stammen! Für die Einbettung in den automatischen Teil des Systems ist dieser Ansatz also sehr vorteilhaft. Hierauf wird in Abschnitt 6.7 näher eingegangen.

Wie sieht es nun mit der interaktiven Seite aus? In jeder Constraintsprache, die mächtig genug zur Gleichheitsbehandlung ist, ist zumindest die syntaktische Gleichheit von Termen ausdrückbar. Wünscht der Benutzer also eine konkrete Instanz für eine Metavariable, so kann er, etwa bei der Quantor-Instantiierung einen Constraint $X = f(a)$ angeben,[4] der konjunktiv zum bisherigen Constraint hinzugefügt wird. Er kann sich jedoch auch, etwa durch einen Constraint der Form $Z = g(X, Y)$ auf die Belegung anderer Metavariablen beziehen. Falls in diesem Beispiel die Metavariablen X und Y noch nicht existieren, gibt der Benutzer auf diesem Weg keine konkrete Instanz an, sondern verlangt nur eine bestimmte Form für die Instanz. Dies kann es dem Benutzer ersparen, alle Details einer Instantiierung anzugeben, wenn das System diese auch selber finden kann. Die Constraintsprache braucht sich jedoch nicht auf syntaktische Gleichheiten zu beschränken. Welche Möglichkeiten hier für den Anwender nützlich sind, wird sich wohl nur in der Praxis feststellen lassen.

Die im letzten Absatz angegebenen Constraints ließen sich auch anders modellieren: Sie lassen sich alle durch die Forderung ausdrücken, daß die abschließende Grundsubsti-

[3] Es ist hier leider nicht möglich, alle Aspekte des Konzepts gesammelt vorzustellen, ohne gelegentlich auf spätere Abschnitte vorzugreifen.

[4] Das Symbol „=" bezeichnet hier die syntaktische Gleichheit von Termen, und nicht die semantische Gleichheit der Objektsprache.

tution in gewissem Sinne Instanz einer vom Benutzer angegebenen Substitution sei. Im ersten Fall wäre dies die Grundsubstitution $[X/f(a)]$, im zweiten Fall eine Nicht-Grund-Substitution der Form $[Z/g(X,Y), X/\ldots, Y/\ldots]$.

Bei letzterem Beispiel ist jedoch das Problem zu erkennen, daß entweder genau darauf geachtet werden muß, daß die Substitution keine Zyklen, wie $[X/f(Y), Y/X]$ enthält (überdies genügt der übliche Instanzbegriff nicht für Substitutionen der Art $[X/f(Y), Y/\ldots]$) oder von der Substitution verlangt werden muß, daß sie idempotent ist – dann müßten X und Y in der Belegung für Z durch die ihnen zugewiesenen Terme ersetzt werden, die recht groß werden können. Diese *konzeptionellen* Probleme ergeben sich bei der Benutzung von Constraints nicht: der Constraint $X = f(Y) \sqcup Y = X$ ist zum Beispiel schlicht unerfüllbar,[5] er beschreibt also die leere Menge von Substitutionen. Unter diesen Umständen ist es sicherlich vorzuziehen, das ohnehin vorhandene, und überdies allgemeinere Konzept der Constraints zu verwenden.

Einen Vorteil, hätte der Ansatz mit der Substitution allerdings: Eine geeignete Darstellung für den Benutzer ergäbe sich durch einfaches Anwenden der von ihm angegebenen Substitution auf den Beweis. Es ist allerdings, wie in Abschnitt 6.6 erläutert wird, ebenso möglich, aus einem vom Benutzer eingegebenen Constraint abzuleiten, welchen Variablen ein konkreter Term zugewiesen wird. Diese Arbeit muß beim Erfüllbarkeitstest für syntaktische Gleichheit durch das Eliminationsverfahren sogar ohnehin ausgeführt werden. Die beim Abschlußtest gewonnene Information kann also bei der Entscheidung über eine geeignete Darstellung weiterverwendet werden.

Eine weitere neue Fragestellung, die sich durch die Trennung von Beweisbaum und Substitution ergibt, betrifft die Darstellung der Abgeschlossenheit von Ästen. In einem interaktiven Beweissystem, das auf dem klassischen Sequenzenkalkül beruht, sind Beweisäste unabhängig von irgendeiner externen Einschränkung offen oder geschlossen. So war es einfach möglich, zum Beispiel durch unterschiedliche Farbgebung, offene und geschlossene Teile des Beweises zu kennzeichnen. Dies ist nun nicht mehr ohne weiteres möglich, da der Status der Äste von einer Grundsubstitution abhängt die nicht direkt zur Verfügung steht. Es ist allerdings möglich, drei wichtige Fälle zu unterscheiden, die zu einer sinnvollen Darstellung des Beweisbaums führen können:

- Ein Ast ist unter *jeder* Substitution geschlossen, die den vom Benutzer angegebenen Constraint erfüllt. Dies entspricht einem klassischen, durch Angabe einer konkreten Instantiierung abgeschlossenen Ast.

- *Es existiert* eine Substitution, die den vom Benutzer angegebenen Constraint erfüllt, und den Ast abschließt. Dabei ist natürlich nicht garantiert, daß diese Substitution auch zum Abschluß der anderen Äste geeignet ist. Dieser Fall ist nur bei der Verwendung von Metavariablen ohne konkrete Instantiierung von Bedeutung.

- Ein Ast ist unter *keiner* Substitution geschlossen, die den vom Benutzer angegebenen Constraint erfüllt. Dieser Fall entspricht einem klassischen offenen Ast.

Eine Kennzeichnung der Beweisäste durch ihren Status bezüglich dieser Einteilung unter dem aktuellen Constraint kann dem Benutzer helfen, die Auswirkungen einer Änderung des Constraints zu verfolgen. Auf die Frage der Beweispräsentation wird in Abschnitt 6.6 noch einmal näher eingegangen werden.

[5] $\sqcup$ bezeichnet die Konjunktion zwischen Constraints, siehe Def. 6.1.

Deutlich konkretere Vorschläge für die Beweispräsentation können an dieser Stelle leider nicht gemacht werden, da die Constraintsprache offengelassen werden soll, um Anforderungen künftiger Verfahren, etwa zur Gleichheitsbehandlung, genügen zu können.

4.6. Zusammenfassung

Es wurde das Konzept der Metavariablen motiviert, die in einem integrativen Kalkül die Funktion der freien Variablen des Tableaukalküls übernehmen sollen. Es wurde argumentiert, daß dieses Konzept Vorteile sowohl für die theoretische Behandlung des Kalküls und dessen konzeptionelle Homogenität, als auch für dessen interaktive Anwendbarkeit hat. Andererseits ergeben sich für die automatische Anwendung keine Nachteile, da Handhabung und Regeln weitgehend den freien Variablen im Tableaukalkül entsprechen.

Ferner wurde das Konzept einer Trennung von Beweisbaum und abschließender Substitution vorgestellt, die orthogonal manipulierbar sind. Es wurde gezeigt, daß Constraints einen natürlichen Weg darstellen, um dem Benutzer die Einschränkung von abschließenden Substitutionen zu erlauben, und dem System ermöglichen, eine Darstellung des Beweises zu finden, die auf die Bedürfnisse des Benutzers zugeschnitten ist.

5. ϵ-Terme statt Skolemfunktionen

Während im letzten Abschnitt im wesentlichen die Instantiierung von Allquantoren, also die γ-Regel im Sinne des Tableaukalküls, erörtert wurde, soll nun Behandlung von Existenzquantoren betrachtet werden. In einem klassischen Sequenzenkalkül bringen diese wenig Probleme mit sich, aber in einem Tableaukalkül mit freien Variablen müssen kompliziertere Regeln verwendet werden, um einen korrekten Kalkül zu erhalten. Diese Komplikationen haben ihren Ursprung in der Möglichkeit, Instantiierungen offenzulassen, und übertragen sich daher auf einen Kalkül mit Metavariablen.

5.1. Skolemisierung

Im klassischen Sequenzenkalkül lautet die Regel zur Behandlung von Existenzquantoren auf der linken Seite

$$\frac{[x/y](\phi), \Gamma \vdash \Delta}{\exists x.\phi, \Gamma \vdash \Delta}$$

wobei y eine neue Variable ist. In einem Kalkül mit Metavariablen sollen aber alle freien Variablen Metavariablen sein. Wird diese Regel so geändert, daß eine neue Metavariable eingeführt wird, ist sie aber offenbar nicht mehr korrekt, da die Metavariablen eine andere Semantik haben. Am Ende von Abschnitt 3.1 wurde auf die Möglichkeit hingewiesen, anstelle einer neuen Variable ein neues Konstantensymbol einzuführen. Das würde zu folgender Regel führen:

$$\frac{[x/c](\phi), \Gamma \vdash \Delta}{\exists x.\phi, \Gamma \vdash \Delta}$$

wobei c ein neues Konstantensymbol ist. Auch diese Regel ist aber nicht korrekt, wenn Metavariablen ins Spiel kommen, da sie folgenden „Beweis“ zuläßt:

$$\cfrac{\cfrac{\cfrac{\cfrac{\cfrac{p(X,c) \vdash p(d,Y)}{p(X,c), \neg p(d,Y) \vdash}\,5}{p(X,c), \exists x.\neg p(x,Y) \vdash}\,4}{p(X,c), \forall y.\exists x.\neg p(x,y) \vdash}\,3}{\exists y.p(X,y), \forall y.\exists x.\neg p(x,y) \vdash}\,2}{\forall x.\exists y.p(x,y), \forall y.\exists x.\neg p(x,y) \vdash}\,1$$

Der Beweis ist mit $[X/d, Y/c]$ geschlossen, obwohl in der Wurzel keine allgemeingültige Sequenz steht. Wenn man die Substitution auf den Beweis anwendet, also zu einem Beweis mit sofortiger Instantiierung übergeht, wird deutlich, wo das Problem liegt. In Schritt 1 wird dann nämlich der Allquantor mit der Konstante d instantiiert, die erst in Schritt 4 in die Signatur aufgenommen wird. Dieser Beweis läßt sich nicht derart umstellen, daß solche Vorgriffe nicht mehr vorkommen.

In [10] wird für den Tableaukalkül mit freien Variablen eine δ-Regel benutzt, die anstelle eines Konstantensymbols ein neues Funktionssymbol erzeugt:[1]

$$\frac{[x/f(X_1,\ldots,X_n)](\phi),\Gamma\vdash\Delta}{\exists x.\phi,\Gamma\vdash\Delta}\ \delta$$

wobei X_1 bis X_n alle *auf dem Beweisast* vorkommenden freien Variablen sind. Um Vorgriffe auf noch nicht eingeführte Signatursymbole zu vermeiden, wird ausgenutzt, daß kein Term echter Teilterm seiner selbst sein kann: Die Parametrisierung des Funktionssymbols hat den Effekt, daß die abschließende Substitution keine der Variablen X_i mit einem Term belegen kann, der den eingeführten Term $f(X_1,\ldots,X_n)$ enthält.

Diese Regel ist korrekt. Sie hat allerdings den Nachteil, sehr restriktiv zu sein: es gibt durchaus Fälle, in denen Vorgriffe erlaubt werden können, etwa wenn der Beweis so umgestellt werden könnte, daß der Vorgriff verschwände. In [11] wird eine liberalisierte Variante vorgestellt, die als δ^+-Regel bekannt ist: Sie parametrisiert das neue Funktionssymbol mit den freien Variablen $X_1,\ldots,X_n$, die *in der Formel* $\exists x.\phi$ vorkommen. In [11] wird die Korrektheit der δ^+-Regel nachgewiesen, und gezeigt, daß sie kürzere Beweise ermöglicht als die obige δ-Regel.

Die Elimination von Existenzquantoren mittels einer Funktion, die mit den freien Variablen der Formel parametrisiert wird, ist als Skolemisierung bekannt. Sie wird zum Beispiel verwendet, um eine beliebige Formel in eine erfüllbarkeitsäquivalente Formel in Klauselnormalform zu transformieren. Für Normalformbeweiser war Skolemisierung also schon immer ein wichtiges Konzept. Die Beweisschritte

$$\cfrac{\cfrac{p(X,f(X)),\forall x.\exists y.p(x,y),\Gamma\vdash\Delta}{\exists y.p(X,y),\forall x.\exists y.p(x,y),\Gamma\vdash\Delta}\ \delta^+}{\forall x.\exists y.p(x,y),\Gamma\vdash\Delta}\ \forall l$$

vermitteln die intuitive Bedeutung der eingeführten Skolemfunktion: Da für alle x die Existenz eines y mit einer bestimmten Eigenschaft behauptet wird, ist es möglich, für jedes x ein solches y *auszuwählen*, und eine Funktion zu definieren, die jedem x das ausgewählte y zuordnet. Diese Sichtweise legt eine weitere Optimierung nahe: Falls zwei Formeln $\exists x.\phi$ und $\exists y.\psi$ äquivalent sind, kann für beide dasselbe Skolemfunktionssymbol verwendet werden! Um eine implementierbare Regel zu erhalten, muß jedoch ein schwächeres Kriterium verwendet werden, da die Äquivalenz unentscheidbar ist. In [7] wird die δ^{+^+}-Regel vorgestellt, die zwei Formeln das gleiche Skolemfunktionssymbol zuordnet, falls diese bis auf Umbenennung der freien und gebundenen Variablen gleich sind. Sie ermöglicht etwa folgende Ableitung:

$$\cfrac{\cfrac{\cfrac{\cfrac{p(X,f(X)),p(X',f(X')),\forall x.\exists y.p(x,y),\Gamma\vdash\Delta}{p(X,f(X)),\exists y.p(X',y),\forall x.\exists y.p(x,y),\Gamma\vdash\Delta}\ \delta^{+^+}}{p(X,f(X)),\forall x.\exists y.p(x,y),\Gamma\vdash\Delta}\ \forall l}{\exists y.p(X,y),\forall x.\exists y.p(x,y),\Gamma\vdash\Delta}\ \delta^{+^+}}{\forall x.\exists y.p(x,y),\Gamma\vdash\Delta}\ \forall l$$

[1] Die vorgestellten Regeln entstammen Tableaukalkülen, wurden aber in Sequenzenschreibweise übersetzt.

In [7] wird nachgewiesen, daß die δ^{++}-Regel wiederum exponentiell kürzere Beweise als die δ^{+}-Regel erlaubt. Die Verwaltung der Skolemfunktionssymbole wird allerdings dadurch erschwert, daß bei jeder Anwendung der δ^{++}-Regel überprüft werden muß, ob schon ein Skolemfunktionssymbol für eine bis auf Variablenumbenennungen gleiche Formel existiert.[2] Jede einzelne Regelanwendung ist also aufwendiger, die Beweise können aber insgesamt kleiner werden.

Die Behandlung von Existenzquantoren in einem integrativen Kalkül sollte *mindestens* die von der δ^{++}-Regel ermöglichten Beweise zulassen, da sonst für den automatischen Gebrauch ein Rückschritt zu verbuchen wäre. Die vorgestellten Skolemisierungsregeln haben jedoch eine Eigenschaft, die für den interaktiven Gebrauch unangenehm ist: Sie führen neue Signatursymbole ein, für die ein Name gebraucht wird. Dieser Name muß entweder vom Beweissystem – mehr oder weniger willkürlich – gewählt werden, woraufhin der Benutzer die Herkunft des Symbols oft nicht nachvollziehen kann. Oder der Benutzer muß ausdrücklich nach einem Namen gefragt werden, was im automatischen Betrieb undenkbar ist. Dieses Phänomen tritt übrigens auch schon bei der klassischen $\exists l$-Regel in interaktiven Systemen auf. Es wird bei der interaktiven Erarbeitung größerer Beweise mit KIV gelegentlich recht schwierig, die verschiedenen, vom System durch Anhängen von Nummern erzeugten, neuen Variablen auseinanderzuhalten.

5.2. Motivation einer ϵ-Logik

Hält man an der Idee der Skolemisierung fest, so muß das Beweissystem Methoden zur Verfügung stellen, um dem Benutzer den Umgang mit den eingeführten Funktionssymbolen zu erleichtern. Insbesondere muß es ihm möglich sein, Namen für Skolemfunktionen festzulegen. Sucht der Automat einen Beweis, wird er zunächst willkürliche Namen wählen. Wenn der Benutzer einen automatisch gefundenen Beweis analysiert, muß er aber die Möglichkeit haben, diese Namen nachträglich zu ändern. Jedenfalls sollte auch zu jedem Skolemsymbol die Information verfügbar sein, für welche Formeln $\exists x.\phi$ es eingeführt wurde. Letztere Information muß bei einer Implementierung der δ^{++}-Regel schon alleine für die automatische Behandlung gehalten werden. Nur anhand der zugehörigen Formeln kann ja entschieden werden, ob ein neues Skolemsymbol benötigt wird, oder ein bereits eingeführtes weiterverwendet werden kann.

Will man die Herkunft, und damit die Bedeutung eines Skolemsymbols nicht nur als Hintergrundinformation mitführen, so bietet es sich an, diese Information bei der automatischen Namensgebung zu nutzen, indem der erzeugte Name die zugehörige Formel textuell enthält. Zum Beispiel könnte die zu $\exists x.p(x,Y)$ gehörige Skolemfunktion den Namen $Sk[\exists x.p(x,Y)]$ erhalten. Die in diesem Namen enthaltene Formel hätte keinerlei semantische Funktion, sondern würde nur zur Dokumentation dienen. Daher würden auch nachträgliche Namensänderungen keine Probleme bereiten. Auch Substitutionen auf Termen, die dieses Symbol enthalten, würden sich nicht auf die namentlich genannte Metavariable Y auswirken. Bei großen Formeln wären diese Namen natürlich unnötig lang, aber der Benutzer kann sie ja bei Bedarf kürzen.

Etwas problematisch wäre bei diesem Ansatz die Parametrisierung: Enthält die Formel mehrere Metavariablen, muß man eine Konvention für die Reihenfolge der Parameter wählen. Ist zum Beispiel die Formel $\exists x.p(f(x,Y),f(Z,Y))$ gegeben, wird eine zweistellige

[2] Mit Hilfe eines geschickten Hashing-Verfahrens mag dies relativ effizient machbar sein. Einen gewissen Mehraufwand bedeutet es aber auf jeden Fall.

Skolemfunktion namens $Sk[\exists x.p(f(x,Y),f(Z,Y))](\ldots,\ldots)$ erzeugt. Welcher der beiden Parameter entspricht nun Y, und welcher Z? Bei größeren Formeln kann dies das Verständnis sehr erschweren, zumal für die Parameter (etwa nach Gleichungsanwendungen) eingesetzte Terme dann von den im Skolemfunktionsnamen genannten Metavariablen räumlich getrennt auftreten.

Weiterhin sollte ein homogener, integrierter Kalkül möglichst keine Hintergrundinformation verwenden, und das Kriterium für die δ^{++}-Regel beruht gerade auf der an Funktionssymbole als Hintergrundinformation gekoppelten Formeln.

Sicherlich ließen sich diese Probleme in angemessener Weise lösen. Es gibt jedoch eine andere Variante, die nicht zu diesen Problemen führt, und eine Reihe von Vorteilen hat, die sie für einen integrativen Kalküls besonders geeignet machen. Zur Motivation mag man den eben beschriebenen Ansatz so variieren, daß die Formeln nicht mehr nur textuell in den Skolemfunktionsnamen vorkommen, sondern nun doch eine Semantik erhalten. Beschränkt man sich zunächst auf den Fall ohne Metavariablen, so kann man die Skolemkonstante $Sk[\exists x.\phi]$ interpretieren als „dasjenige x, für das ϕ gilt". Das so natürlichsprachlich beschriebene Objekt soll formal durch den Term $\epsilon x.\phi$ dargestellt werden.

Dabei muß natürlich festgelegt werden, wie dieser ϵ-Term zu interpretieren ist, wenn es in einer gegebenen Struktur kein, oder mehrere x gibt, für das ϕ wahr wird. Schließlich soll ein Term in jeder Struktur genau eine Auswertung besitzen. Um dies zu gewährleisten, verlangt die Semantik der ϵ-Terme, daß bei der Existenz mehrerer erfüllender Trägerelemente eines von diesen ausgewählt wird. Falls kein erfüllendes Element existiert, wird ein beliebiges Element der Trägermenge ausgewählt. Eine korrektere natürlichsprachliche Umschreibung von $\epsilon x.\phi$ lautet also: „ein x, für das ϕ gilt, vorausgesetzt ein solches existiert".

Um zu einer Modellsemantik zu kommen, wird die Termauswertungsfunktion mit der Gesamtheit der Entscheidungen über die ausgewählten Elemente zu allen möglichen ϵ-Termen parametrisiert. Ein ϵ-Term wird also immer nur ausgewertet bezüglich einer *a priori* getroffenen Auswahl, die ihm einen eindeutigen Wert zuweist.

Die ϵ-Terme wurden ursprünglich von Hilbert eingeführt, siehe etwa [13]. In Anhang A wird ein kleiner Überblick gegeben über den Kontext, in dem die ϵ-Terme von Hilbert eingeführt werden, obwohl dieser Kontext für unsere Betrachtungen nicht erheblich ist.

In Hilberts Arbeit fehlt die heute für die Prädikatenlogik übliche Modellsemantik, siehe Anhang A. Eine solche Modellsemantik für ϵ-Terme wird etwa in [14] und [15] angegeben.

[14] ist ein Lehrbuch zur mathematischen Logik, in dem von Anfang an ϵ-Terme verwendet werden, und zwar mit extensionaler Semantik, siehe Def. 5.12. Auf andere mögliche Interpretationen von ϵ-Termen wird nur am Rande eingegangen.

In [15] werden umfassend Varianten von klassischen, intuitionistischen und nichtmonotonen Logiken mit ϵ-Termen untersucht. Der Schwerpunkt liegt dabei auf der Behandlung im Rahmen von Kalkülen des *natural deduction* Typs. Auch die Anwendung in der Linguistik, speziell die formale Darstellung von natürlichsprachlichen Sätzen durch logische Formeln, wird eingehend erörtert. Nicht behandelt wird dort die Anwendung von ϵ-Termen für automatisches, bzw. integriertes Beweisen. Auch die folgenden ausführlichen Korrektheits- und Vollständigkeitsbeweise fehlen dort.

Um zur Anwendung von ϵ-Termen als Ersatz für Skolemisierung zurückzukommen, soll nun eine entsprechende Regel vorgestellt werden. In den folgenden Abschnitten wird näher auf die genaue Semantik der ϵ-Terme eingegangen werden, hier soll nur informell

Abbildung 5.1.: Ein Beweis mit ϵ-Termen

$$
\begin{array}{l}
X/\epsilon x.\forall y.\neg p(x,y)\\
Y/\epsilon y.\neg q(\epsilon x.\forall y.\neg p(x,y),y)\\
U/\epsilon x.\forall y.\neg p(x,y)\\
V/\epsilon y.\neg q(\epsilon x.\forall y.\neg p(x,y),y)
\end{array}
$$

$$
\begin{prooftree}
\AxiomC{$p(U,V) \ \vdash \ p(\epsilon x.\forall y.\neg p(x,y),Y),\ q(X,\epsilon y.\neg q(X,y))$}
\AxiomC{$q(U,V) \ \vdash \ p(\epsilon x.\forall y.\neg p(x,y),Y),\ q(X,\epsilon y.\neg q(X,y))$}
\RightLabel{$\vee l$ 9}
\BinaryInfC{$p(U,V)\vee q(U,V) \ \vdash \ p(\epsilon x.\forall y.\neg p(x,y),Y),\ q(X,\epsilon y.\neg q(X,y))$}
\RightLabel{$\forall l$ 8}
\UnaryInfC{$\forall v.(p(U,v)\vee q(U,v)) \ \vdash \ p(\epsilon x.\forall y.\neg p(x,y),Y),\ q(X,\epsilon y.\neg q(X,y))$}
\RightLabel{$\forall l$ 7}
\UnaryInfC{$\forall u.\forall v.(p(u,v)\vee q(u,v)) \ \vdash \ p(\epsilon x.\forall y.\neg p(x,y),Y),\ q(X,\epsilon y.\neg q(X,y))$}
\RightLabel{$\neg l$ 6}
\UnaryInfC{$\neg q(X,\epsilon y.\neg q(X,y)),\ \forall u.\forall v.(p(u,v)\vee q(u,v)) \ \vdash \ p(\epsilon x.\forall y.\neg p(x,y),Y)$}
\RightLabel{$\exists l$ 5}
\UnaryInfC{$\exists y.\neg q(X,y),\ \forall u.\forall v.(p(u,v)\vee q(u,v)) \ \vdash \ p(\epsilon x.\forall y.\neg p(x,y),Y)$}
\RightLabel{$\forall l$ 4}
\UnaryInfC{$\forall x.\exists y.\neg q(x,y),\ \forall u.\forall v.(p(u,v)\vee q(u,v)) \ \vdash \ p(\epsilon x.\forall y.\neg p(x,y),Y)$}
\RightLabel{$\neg l$ 3}
\UnaryInfC{$\neg p(\epsilon x.\forall y.\neg p(x,y),Y),\ \forall x.\exists y.\neg q(x,y),\ \forall u.\forall v.(p(u,v)\vee q(u,v)) \ \vdash$}
\RightLabel{$\forall l$ 2}
\UnaryInfC{$\forall y.\neg p(\epsilon x.\forall y.\neg p(x,y),y),\ \forall x.\exists y.\neg q(x,y),\ \forall u.\forall v.(p(u,v)\vee q(u,v)) \ \vdash$}
\RightLabel{$\exists l$ 1}
\UnaryInfC{$\exists x.\forall y.\neg p(x,y),\ \forall x.\exists y.\neg q(x,y),\ \forall u.\forall v.(p(u,v)\vee q(u,v)) \ \vdash$}
\end{prooftree}
$$

ein Gefühl für deren Verwendung vermittelt werden. Die neue $\exists l$-Regel lautet also:

$$\frac{[x/\epsilon x.\phi](\phi), \Gamma \vdash \Delta}{\exists x.\phi, \Gamma \vdash \Delta} \; \exists l$$

Um Anwendung und Probleme dieser Regel zu verdeutlichen, wird in Abbildung 5.1 ein Beweis für die allgemeingültige Sequenz

$$\exists x.\forall y.\neg p(x,y),\ \forall x.\exists y.\neg q(x,y),\ \forall x.\forall y.(p(x,y) \vee q(x,y))\ \vdash$$

vorgestellt. Da die Sequenzen recht lang werden, werden wieder einige Formeln, die nicht mehr benötigt werden, stillschweigend weggelassen. Die angegebenen Beweisregeln werden dabei jeweils auf die ganz links stehende Formel angewandt. Der Beweis ist mit der angegebenen Grundsubstitution für die Metavariablen geschlossen.

Wie in Schritt 5 deutlich wird, können in ϵ-Termen auch Metavariablen vorkommen. Die abschließende Substitution bezieht sich dann auch auf diese. Außerdem können die in ϵ-Termen vorkommenden Formeln durchaus weitere ϵ-Terme enthalten.

Die Verwendung von ϵ-Termen mit dieser Regel hat folgende Vorzüge:

- Wie in 6.4.3 gezeigt werden wird, ist die $\exists l$-Regel mit ϵ-Termen (bei entsprechend angepaßtem Unifikationsbegriff) mächtiger als die δ^{++}-Regel. Es sind also noch kürzere Beweise möglich als mit dieser.

- Die Erzeugung von willkürlichen Bezeichnern für Skolemfunktionen entfällt völlig.

- ϵ-Terme verhalten sich in mancher Hinsicht intuitiver als Skolemfunktionen. In der betroffenen Formel komme beispielsweise eine Metavariable vor, für die der Benutzer eine konkrete Instantiierung vorgegeben hat. Wenn – wie in 4.4 vorgeschlagen – dem Benutzer anstelle der Metavariable der von ihm angegebene Term angezeigt wird, wird dieser eine Formel ohne Metavariablen sehen. Eine Skolemisierungsregel wird trotzdem eine Skolemfunktion mit einem Parameter erzeugen. Bei der Verwendung von ϵ-Termen wird dagegen ein ϵ-Term erzeugt, der die Metavariable enthält. Diese wird bei der Darstellung weiterhin durch den vom Benutzer angegebenen Term ersetzt. Nach außen verhält sich das System so, als stünde an Stelle der Metavariable tatsächlich der angegebene Term.

- Es gibt Hinweise darauf, daß ϵ-Terme für die Transformation formaler Beweise in natürliche Sprache vorteilhaft sind.[3]

An dem Beispiel in Abbildung 5.1 ist schon der wesentliche Nachteil der Verwendung von ϵ-Termen zu erkennen: Bei komplexen Formeln werden die ϵ-Terme sehr groß und unübersichtlich. Dies wird besonders dann zum Problem, wenn eine Formel die quantifizierte Variable mehrfach enthält, was zu entsprechend vielen Kopien des ϵ-Terms führt. Aus der Formel $\exists x.p(x, f(x), g(x))$ wird zum Beispiel

$$p(\epsilon x.p(x, f(x), g(x)), f(\epsilon x.p(x, f(x), g(x))), g(\epsilon x.p(x, f(x), g(x)))).$$

Dieses Verhalten ist aus zweierlei Gründen problematisch. Erstens führt es zu unübersichtlichen Sequenzen, ist also für den interaktiven Gebrauch hinderlich. Zweitens führen

[3] Mündliche Mitteilung von Prof. Dahn, Humboldt Universität Berlin.

große Formeln zu Effizienzproblemen, wenn die Implementierung sie durch ungeschickte Datenstrukturen repräsentiert.

Das Problem der Übersichtlichkeit wäre mittels eines *Abkürzungsmechanismus* zu beheben, der es dem Benutzer zum Beispiel erlaubt, dem System für komplexe Terme eine kompaktere Darstellung vorzugeben. Benutzt man ein Verfahren, das auch parametrisierte Abkürzungen erlaubt, kann man eine ähnliche Darstellung erreichen, wie sie sich mit vom Benutzer benannten Skolemfunktionen ergeben hätte. Es handelt sich jetzt aber nur um eine Darstellung, der Benutzer kann – in einer graphischen Umgebung etwa durch einen Mausklick auf die Abkürzung – die von ihm verlangten Kürzel jederzeit expandieren. Ein solcher Mechanismus ist im übrigen für ein leistungsfähiges interaktives System ohnehin nötig. Es kommt beim interaktiven Beweisen häufig zu umfangreichen Formeln, die zunächst nicht benötigt werden, aber zu unübersichtlichen Sequenzen führen. Es ist für den Benutzer dann sehr angenehm, wenn er die Möglichkeit hat, diese Formeln zwischenzeitlich abzukürzen. Für die Darstellungskomponente eines integrierten automatischen und interaktiven Beweissystems ist also jedenfalls ein Abkürzungsmechanismus nötig. Mit dessen Hilfe können auch ϵ-Terme günstig dargestellt werden.

Auf die Frage nach der effizienten Implementierbarkeit einer Logik mit ϵ-Termen wird in Anhang B näher eingegangen werden. Hier sei nur gesagt, daß Datenstrukturen existieren, die es erlauben, eine Logik mit ϵ-Termen genauso effizient zu implementieren wie einen Kalkül mit δ^{++}-Regel.

Nachdem nun die Vorzüge einer Logik mit ϵ-Termen vorgestellt wurden, soll in den Abschnitten 5.3 bis 5.5 eine formale Definition einer solchen Logik gegeben werden, und gezeigt werden, daß sie eine konservative Erweiterung der Prädikatenlogik erster Stufe ist.

In Abschnitt 5.6 bis wird ein korrekter und vollständiger Kalkül für diese Logik vorgestellt. Der Beweis, daß dieser mächtiger ist als ein Kalkül mit δ^{++}-Regel wird erst in 6.4.3 nachgereicht, da er sich auf einen Kalkül mit Metavariablen bezieht. Bei dem vorgestellten Kalkül wird darauf Wert gelegt, daß er bezüglich der vollen ϵ-Logik vollständig ist, und nicht nur bezüglich des Fragments, das für Beweise von Formeln der Prädikatenlogik erster Stufe nötig ist. Der Benutzer kann also ohne Einschränkungen ϵ-Terme zum Beispiel auch zur Formulierung von Lemmata, für die Eingabe von Fallunterscheidungen mittels der *cut*-Regel, oder zur Angabe von Instantiierungen verwenden. In Abschnitten 5.7 und 5.8 werden einige Varianten von Syntax und Semantik der ϵ-Terme diskutiert.

5.3. Syntax

In der formalen Beschreibung der ϵ-Logik soll auf eine Sortierung verzichtet werden. Ferner sei eine feste Signatur vorausgesetzt. Insbesondere seien folgende disjunkte Mengen gegeben: eine Menge $\mathbf{V}$ von **Objektvariablen**, eine unendliche Menge $\mathbf{M}$ von **Metavariablen**, eine Menge $\mathbf{F}$ von **Funktionssymbolen**, eine Menge $\mathbf{P}$ von **Prädikatensymbolen**, sowie eine Funktion $\rho : (\mathbf{F} \cup \mathbf{P}) \to \mathbb{N}$, die jedem Funktions- und Prädikatensymbol eine **Stelligkeit** zuordnet.

5.1 Definition (Formeln und Terme)
Die Menge **Form** *der* **Formeln** *und die Menge* **Term** *der* **Terme** *werden rekursiv wie folgt definiert:*

- $x \in \mathbf{Term}$ *für jede Objektvariable* $x \in \mathbf{V}$.

- $X \in \mathbf{Term}$ *für jede Metavariable* $X \in \mathbf{M}$.
- $f(t_1, \ldots, t_n) \in \mathbf{Term}$ *für jedes* $f \in \mathbf{F}$ *mit* $\rho(f) = n$ *und Terme* $t_1, \ldots, t_n \in \mathbf{Term}$.
- $\epsilon x.\phi \in \mathbf{Term}$ *für jede Objektvariable* $x \in \mathbf{V}$ *und jede Formel* $\phi \in \mathbf{Form}$.
- $p(t_1, \ldots, t_n) \in \mathbf{Form}$ *für jedes* $p \in \mathbf{P}$ *mit* $\rho(p) = n$ *und Terme* $t_1, \ldots, t_n \in \mathbf{Term}$.
- $s \doteq t \in \mathbf{Form}$ *für alle Terme* $s, t \in \mathbf{Term}$.
- $\phi \wedge \psi \in \mathbf{Form}$ *für alle Formeln* $\phi, \psi \in \mathbf{Form}$.
- $\neg\phi \in \mathbf{Form}$ *für alle Formeln* $\phi \in \mathbf{Form}$.
- $\exists x.\phi \in \mathbf{Form}$ *für alle Objektvariablen* $x \in \mathbf{V}$ *und Formeln* $\phi \in \mathbf{Form}$.

Die Disjunktion und andere boole'sche Konnektive mögen ebenso wie der Allquantor in der üblichen Weise als Abkürzungen angesehen werden. In einer Implementierung würde man natürlich auch für diese Regeln hinzufügen. Diese entsprechen jedoch genau denen für die dualen Operatoren.

Neu gegenüber der klassischen Prädikatenlogik erster Stufe sind die ϵ-Terme, deren syntaktische Besonderheit darin besteht, daß durch sie Formeln syntaktisch in Terme eingebettet werden. Es ist etwa $\epsilon x.(p(x) \wedge \neg q(y))$ ein Term! Neu ist weiterhin die Unterscheidung zwischen den Objektvariablen $\mathbf{V}$ und den Metavariablen $\mathbf{M}$. Man bemerke, daß durch Quantoren und ϵ-Terme nur die Objektvariablen gebunden werden.

Während man für die Prädikatenlogik Grundterme schlicht als Terme definiert, in denen keine Variablen vorkommen, ist hier ein anderes Vorgehen nötig, um zu einem nützlichen Begriff zu kommen: in solchen Termen könnten ja keine ϵ-Terme vorkommen. Unsere Definition verwendet die Menge der freien Variablen von Termen und Formeln. Da in Formeln, die für Sequenzen verwendet werden, nur Metavariablen frei vorkommen sollen, wird der Begriff der *ganzen* Formeln dafür eingeführt.

5.2 Definition (Freie und gebundene Variablen, Grundterme)
Für eine beliebige Menge A *bezeichne* $\mathcal{P}(A) = \{B \mid B \subseteq A\}$ *deren Potenzmenge. Für alle* $\alpha \in \mathbf{Form} \cup \mathbf{Term}$ *sei* $\bar{\mathrm{v}}(\alpha) = (\mathrm{mv}(\alpha), \mathrm{fv}(\alpha), \mathrm{bv}(\alpha)) \in \mathcal{P}(\mathbf{M}) \times \mathcal{P}(\mathbf{V}) \times \mathcal{P}(\mathbf{V})$ *rekursiv definiert durch:*

- $\bar{\mathrm{v}}(x) = (\emptyset, \{x\}, \emptyset)$ *für jede Objektvariable* $x \in \mathbf{V}$.
- $\bar{\mathrm{v}}(X) = (\{X\}, \emptyset, \emptyset)$ *für jede Metavariable* $X \in \mathbf{M}$.
- $\bar{\mathrm{v}}(c) = (\emptyset, \emptyset, \emptyset)$ *für jedes* $c \in \mathbf{F}$ *mit* $\rho(c) = 0$.
- $\bar{\mathrm{v}}(f(t_1, \ldots, t_n)) = \bar{\mathrm{v}}(p(t_1, \ldots, t_n)) = \bar{\mathrm{v}}(t_1 \doteq t_2) = (\bigcup_i \mathrm{mv}(t_i), \bigcup_i \mathrm{fv}(t_i), \bigcup_i \mathrm{bv}(t_i))$
- $\bar{\mathrm{v}}(\phi_1 \wedge \phi_2) = \bar{\mathrm{v}}(\neg\phi_1) = (\bigcup_i \mathrm{mv}(\phi_i), \bigcup_i \mathrm{fv}(\phi_i), \bigcup_i \mathrm{bv}(\phi_i))$
- $\bar{\mathrm{v}}(\epsilon x.\phi) = \bar{\mathrm{v}}(\exists x.\phi) = (\mathrm{mv}(\phi), \mathrm{fv}(\phi) \backslash \{x\}, \mathrm{bv}(\phi) \cup \{x\})$

Von den Komponenten des so definierten Tripels bezeichnet $\mathrm{mv}(\alpha)$ *die Menge der Metavariablen,* $\mathrm{fv}(\alpha)$ *die Menge der* **freien** *Objektvariablen und* $\mathrm{bv}(\alpha)$ *die Menge der* **gebundenen** *Objektvariablen in* α.

*Ein Term $t \in$ **Term** heißt **Grundterm**, falls $\mathrm{fv}(t) = \mathrm{mv}(t) = \emptyset$ und **ganzer Term**, falls $\mathrm{fv}(t) = \emptyset$. Die Menge aller Grundterme wird mit $\mathbf{Term}^0$, und die Menge aller ganzen Terme mit $\mathbf{Term}^*$ bezeichnet. Die Menge aller Terme ohne Metavariablen wird mit $\mathbf{Term}^+ = \{t \in \mathbf{Term} \mid \mathrm{mv}(t) = \emptyset\}$ bezeichnet.*

*Eine Formel $\phi \in$ **Form** heißt **ganze Formel**, falls $\mathrm{fv}(\phi) = \emptyset$. Sie heißt **geschlossen**, falls $\mathrm{fv}(\phi) = \mathrm{mv}(\phi) = \emptyset$. Die Mengen der ganzen, resp. geschlossenen Formeln werden mit $\mathbf{Form}^*$, resp. $\mathbf{Form}^0$ bezeichnet. Für die Menge der Formeln ohne Metavariablen soll $\mathbf{Form}^+$ geschrieben werden.*

Hier ist zu bemerken, daß $\mathrm{fv}(\alpha)$ nur die freien *Objekt*variablen bezeichnet, und nicht die Metavariablen. Die definierten Formel- und Termmengen stehen in folgenden Teilmengen-Verhältnissen:

$$\begin{tikzcd}[column sep=small] & \mathbf{Term} & \\ \mathbf{Term}^+ \arrow[ur, phantom, sloped, "\subseteq"] & & \mathbf{Term}^* \arrow[ul, phantom, sloped, "\subseteq"] \\ & \mathbf{Term}^0 \arrow[ul, phantom, sloped, "\subseteq"] \arrow[ur, phantom, sloped, "\subseteq"] & \end{tikzcd} \quad \text{bzw.} \quad \begin{tikzcd}[column sep=small] & \mathbf{Form} & \\ \mathbf{Form}^+ \arrow[ur, phantom, sloped, "\subseteq"] & & \mathbf{Form}^* \arrow[ul, phantom, sloped, "\subseteq"] \\ & \mathbf{Form}^0 \arrow[ul, phantom, sloped, "\subseteq"] \arrow[ur, phantom, sloped, "\subseteq"] & \end{tikzcd}$$

Aufgrund der Einführung von Metavariablen, und deren gesonderter Behandlung, werden im folgenden Substitutionen, Belegungen und deren Verhältnis zueinander eine wichtige Rolle spielen. Darum soll für die beiden recht ähnlichen Konzepte bewußt eine unterschiedliche Schreibweise verwendet werden.

5.3 Definition (Substitutionen)
*Sei $A \subseteq \mathbf{M} \cup \mathbf{V}$ eine Menge von Variablen. Eine **Substitution** auf A ist eine Funktion $\sigma : A \to \mathbf{Term}$, deren **Wirkungsbereich** $\mathrm{dom}(\sigma) := \{x \in A \mid \sigma(x) \neq x\} \subseteq A$ endlich ist. Die Menge aller Substitutionen auf A wird mit $\mathbf{Sub}(A)$ bezeichnet. Für eine feste Substitution wird gelegentlich die Schreibweise $\sigma = [x_1/t_1, \ldots, x_n/t_n]$ für $\sigma(x_i) = t_i$ verwendet.*

*$\sigma \in \mathbf{Sub}(A)$ heißt **Grundsubstitution**, falls $\sigma(x) \in \mathbf{Term}^0$ für alle $x \in A$. Die Menge aller Grundsubstitutionen auf A wird mit $\mathbf{Sub}^0(A)$ bezeichnet. Für die Menge aller Grundsubstitutionen der Metavariablen wird kurz $\mathbf{Sub}^0 := \mathbf{Sub}^0(M)$ geschrieben.*

*Für eine Variable $v \in A$ und einen Term $t \in$ **Term** wird die **Abänderung** von σ an der Stelle v definiert durch*

$$\sigma_v^t(x) := \begin{cases} t & \textit{falls } x = v \\ \sigma(x) & \textit{sonst} \end{cases}$$

Der Definitionsbereich von Substitutionen wird in der üblichen Weise auf Terme und Formeln erweitert, also $\sigma : \mathbf{Term} \to \mathbf{Term}$ bzw. $\sigma : \mathbf{Form} \to \mathbf{Form}$. Insbesondere ist $\sigma(\epsilon x.\phi) = \epsilon x.\sigma_x^x(\phi)$.

*Die Funktionskomposition $\sigma \circ \tau$ mit $(\sigma \circ \tau)(x) := \sigma(\tau(x))$ zweier Substitutionen ist wieder eine Substitution, die als **Komposition** von σ und τ bezeichnet wird.*

*Gilt für eine Substitution $\sigma \circ \sigma = \sigma$, so heißt σ **idempotent**. Eine **ganze Substitution** ist eine idempotente Substitution $\sigma \in \mathbf{Sub}(M)$ mit $\mathrm{fv}(X) = \emptyset$ für alle $X \in \mathrm{dom}(\sigma)$. Es bezeichne $\mathbf{Sub}^*$ die Menge aller ganzen Substitutionen.*

5.4 Definition (Belegung)
Eine **Belegung** *von* $\mathbf{V}$ *auf eine Menge* $\mathcal{D}$ *ist eine Funktion* $\beta : \mathbf{V} \to \mathcal{D}$.

Für eine Variable $v \in \mathbf{V}$ *und ein Element* $d \in \mathcal{D}$ *wird die* **Abänderung** *von* β *an der Stelle* v *definiert durch*

$$\beta\{v \leftarrow d\}(x) := \begin{cases} d & \text{falls } x = v \\ \beta(x) & \text{sonst} \end{cases}$$

5.4. Semantik

Da die ϵ-Logik eine Erweiterung der Prädikatenlogik erster Stufe sein soll, orientiert sich die Semantik der ϵ-Logik an der Semantik der Prädikatenlogik. Dabei müssen die prädikatenlogischen Strukturen aus Trägermenge und Interpretation um eine Interpretation der ϵ-Terme erweitert werden. Diese soll jedem ϵ-Term ein Element der Trägermenge zuordnen. Da sich für die Semantik der ϵ-Terme mehrere mögliche Varianten ergeben werden, und außerdem spätere Beweise dadurch vereinfacht werden, sollen an diese Interpretation zunächst keinerlei Bedingungen gestellt werden, die die intendierte Semantik beträfen. Daher werden zunächst in Definition 5.5 Prä-Strukturen eingeführt. Durch die Definitionen 5.8, 5.9 und 5.12 werden dann zusätzliche Forderungen eingeführt.[4]

5.5 Definition (Prä-Struktur)
Eine **Prä-Struktur** *ist ein Tripel* $\mathcal{S} = (\mathcal{D}, \mathcal{I}, \mathcal{A})$. *Dabei ist die* **Trägermenge** $\mathcal{D}$ *eine nicht-leere Menge. Die* **Interpretation** $\mathcal{I}$ *ist eine Funktion, die jedem Funktions- und Prädikatensymbol eine Funktion*

$$\begin{aligned} \mathcal{I}(f) : \mathcal{D}^n &\longrightarrow \mathcal{D} && \text{für } f \in \mathbf{F} \text{ mit } \rho(f) = n \\ \mathcal{I}(p) : \mathcal{D}^n &\longrightarrow \{tt, ff\} && \text{für } p \in \mathbf{P} \text{ mit } \rho(p) = n \end{aligned}$$

zuordnet. Die **ϵ-Auswertung** $\mathcal{A}$ *ist eine Funktion, die jedem ϵ-Term* $\epsilon x.\phi \in \mathbf{Term}^+$ *ohne Metavariablen und jeder Belegung* β *von* $\mathbf{V}$ *auf* $\mathcal{D}$ *einen Wert* $\mathcal{A}(\epsilon x.\phi, \beta) \in \mathcal{D}$ *zuordnet.*[5] *Dabei soll der Wert von* $\mathcal{A}$ *nur von der Belegung der in* $\epsilon x.\phi$ *freien Variablen abhängen,*[6] *also falls* $\beta_1|_{\mathrm{fv}(\epsilon x.\phi)} = \beta_2|_{\mathrm{fv}(\epsilon x.\phi)}$, *dann* $\mathcal{A}(\epsilon x.\phi, \beta_1) = \mathcal{A}(\epsilon x.\phi, \beta_2)$. *Für geschlossene ϵ-Terme* $\epsilon x.\phi$ *wird die Belegung weggelassen, und kurz* $\mathcal{A}(\epsilon x.\phi)$ *geschrieben.*

Da Metavariablen als Platzhalter für Grundterme verstanden werden, werden diese auch nicht durch Variablenbelegungen interpretiert. Vielmehr beziehen sich die folgenden semantischen Definitionen auf Terme und Formeln, auf die bereits eine Grundsubstitution der Metavariablen angewandt wurde. Die Termauswertungsfunktion und die

[4] Die Alternative wäre eine simultane Definition von Strukturen und Auswertungsfunktionen durch eine Stufenkonstruktion, wie sie im Beweis des Einbettungssatzes 5.14 durchgeführt wird. Die Definition wäre allerdings sehr unübersichtlich, und auch spätere Beweise würden aufgebläht, da sie sich an dieser Konstruktion orientieren müßten.

[5] In [15] wird die ϵ-Auswertung von Trägermenge und Interpretation getrennt gehalten. Sie tritt in der Auswertungsfunktion val und der Gültigkeitsrelation $\models$ dann – wie die Variablenbelegung – als weiterer Parameter auf. Diese Trennung bietet die Möglichkeit, verschiedene Varianten der Folgerbarkeit zu untersuchen. Da ϵ-Terme hier jedoch hauptsächlich zur Quantorenbehandlung in der Prädikatenlogik erster Stufe eingeführt werden, sind derartige semantische Differenzierungen im Zusammenhang dieser Arbeit nicht interessant.

[6] Dies wird nicht als echte Einschränkung an die ϵ-Auswertung aufgefaßt. Es wäre auch möglich gewesen, $\mathcal{A}$ von vornherein mit einer Belegung *der freien Variablen* des ϵ-Terms zu parametrisieren. Das hätte allerdings zu einem übertriebenen notationellen Aufwand für eine recht unwichtige Bedingung geführt.

Gültigkeitsrelation werden also immer nur auf Terme und Formeln angewandt, die keine Metavariablen enthalten.

5.6 Definition (Auswertung von Termen)
Die **Auswertung** $\mathrm{val}(\mathcal{S}, \beta, t) \in \mathcal{D}$ *eines Terms* $t \in \mathbf{Term}^+$ *in einer Prä-Struktur* $\mathcal{S} = (\mathcal{D}, \mathcal{I}, \mathcal{A})$ *unter einer Belegung* $\beta : \mathbf{V} \to \mathcal{D}$ *wird wie folgt rekursiv definiert.*

- $\mathrm{val}(\mathcal{S}, \beta, x) = \beta(x)$.
- $\mathrm{val}(\mathcal{S}, \beta, f(t_1, \ldots, t_n)) = \mathcal{I}(f)(\mathrm{val}(\mathcal{S}, \beta, t_1), \ldots, \mathrm{val}(\mathcal{S}, \beta, t_n))$.
- $\mathrm{val}(\mathcal{S}, \beta, \epsilon x.\phi) = \mathcal{A}(\epsilon x.\phi, \beta)$.

Für Grundterme $t \in Term^0$ *wird die Belegung weggelassen, und* $\mathrm{val}(\mathcal{S}, t)$ *geschrieben, da diese keinen Einfluß auf die Auswertung von Grundtermen hat.*

Auffällig ist hier, daß bei den ϵ-Termen *keine* Verwendung der Semantik von Formeln stattfindet. Da von $\mathcal{A}$ zunächst keine Auswahleigenschaften verlangt werden, ist hier keine wechselseitig rekursive Definition wie bei der Syntax nötig. Die Auswertung, beziehungsweise Gültigkeit von Formeln wird genauso definiert wie in der Prädikatenlogik ohne ϵ-Terme:

5.7 Definition (Auswertung von Formeln, Gültigkeit)
Die **Auswertung** $\mathrm{val}(\mathcal{S}, \beta, \phi) \in \{tt, ff\}$ *einer Formel* $\phi \in \mathbf{Form}^+$ *in einer Prä-Struktur* $\mathcal{S} = (\mathcal{D}, \mathcal{I}, \mathcal{A})$ *unter einer Belegung* $\beta : \mathbf{V} \to \mathcal{D}$ *wird wie folgt rekursiv definiert:*

- $\mathrm{val}(\mathcal{S}, \beta, p(t_1, \ldots, t_n)) = \mathcal{I}(p)(\mathrm{val}(\mathcal{S}, \beta, t_1), \ldots, \mathrm{val}(\mathcal{S}, \beta, t_n))$.
- $\mathrm{val}(\mathcal{S}, \beta, s \doteq t) = tt$ *gdw.* $\mathrm{val}(\mathcal{S}, \beta, s) = \mathrm{val}(\mathcal{S}, \beta, t)$.
- $\mathrm{val}(\mathcal{S}, \beta, \phi \land \psi) = tt$ *gdw.* $\mathrm{val}(\mathcal{S}, \beta, \phi) = tt$ *und* $\mathrm{val}(\mathcal{S}, \beta, \psi) = tt$.
- $\mathrm{val}(\mathcal{S}, \beta, \neg\phi) = tt$ *gdw.* $\mathrm{val}(\mathcal{S}, \beta, \phi) = ff$.
- $\mathrm{val}(\mathcal{S}, \beta, \exists x.\phi) = tt$ *gdw. ein Trägerelement* $d \in \mathcal{D}$ *existiert mit* $\mathrm{val}(\mathcal{S}, \beta\{x \leftarrow d\}, \phi) = tt$.

Eine Formel ϕ *ohne Metavariablen heißt* **in einer Prä-Struktur** $\mathcal{S} = (\mathcal{D}, \mathcal{I}, \mathcal{A})$ **unter einer Belegung** $\beta : \mathbf{V} \to \mathcal{D}$ **gültig**, *in Zeichen* $\mathcal{S}, \beta \models \phi$, *falls* $\mathrm{val}(\mathcal{S}, \beta, \phi) = tt$.

Wie bei Termen wird die Belegung weggelassen, falls es sich um geschlossene Formeln handelt, da sie dann keinen Einfluß auf die Gültigkeit hat.

Nun sollen zusätzliche Eigenschaften von Prä-Strukturen definiert werden, die die bislang völlig beliebige Interpretation von ϵ-Termen einschränken. Die wichtigste Forderung ist, daß $\epsilon x.$ als Auswahloperator wirkt. Das heißt, falls es überhaupt ein Trägerelement gibt, das ϕ gültig macht, so soll $\epsilon x.\phi$ zu einem solchen ausgewertet werden.

5.8 Definition (Auswahleigenschaft, Struktur)
Eine Prä-Struktur $\mathcal{S} = (\mathcal{D}, \mathcal{I}, \mathcal{A})$ *hat die* **Auswahleigenschaft**, *falls für alle Formeln* $\phi \in \mathbf{Form}^+$ *ohne Metavariablen, alle Variablen* $x \in \mathbf{V}$ *und alle Belegungen* $\beta : \mathbf{V} \to \mathcal{D}$ *gilt:*

$$\text{Falls } \mathcal{S}, \beta \models \exists x.\phi, \text{ dann } \mathcal{S}, \beta\{x \leftarrow \mathcal{A}(\epsilon x.\phi, \beta)\} \models \phi.$$

Eine Prä-Struktur, die die Auswahleigenschaft hat, heißt **Struktur**, *oder* **I-Struktur**. *Gilt eine Formel ϕ in allen Strukturen unter allen Belegungen, so heißt sie* **I-allgemeingültig**,[7] *oder kurz* **I-gültig**, *in Zeichen $\models_I \phi$. Ist eine Formel in mindestens einer Struktur unter mindestens einer Belegung gültig, so heißt sie* **I-erfüllbar**.

Beschränkt man sich darauf, die Auswahleigenschaft zu fordern, erhält man die sogenannte *intensionale* Semantik der ϵ-Terme. Dieser Semantik wird in [15] die meiste Aufmerksamkeit gewidmet. Auch Hilberts Anwendung der ϵ-Terme beruht auf der intensionalen Semantik, obgleich dies wegen des Verzichts auf eine Modellsemantik nicht explizit gemacht wird.

Ein wichtige Eigenschaft der klassischen Prädikatenlogik ist die *Substitutivität*, die im sogenannten Substitutionslemma nachgewiesen wird. Die Substitutivität erlaubt es, die semantische Bedeutung der syntaktischen Operation des Ersetzens von Teiltermen in Formeln zu beschreiben. Sie ist daher ein wichtiges Hilfsmittel um die Korrektheit von Beweisregeln nachzuweisen.

Die ϵ-Logik mit intensionaler Semantik ist *nicht* substitutiv. Etwa folgt aus der Formel $\forall x.q(\epsilon y.p(x,y))$ nicht $q(\epsilon y.p(a,y))$. Ebensowenig kann aus $a \doteq b$ geschlossen werden, daß $\epsilon x.p(x,a) \doteq \epsilon x.p(x,b)$. Man kann jedoch die Substitutivität als zusätzliche Eigenschaft von Strukturen fordern, um zu einer ϵ-Logik mit substitutiver Semantik zu gelangen. In dieser kann dann das Substitutionslemma uneingeschränkt angewandt werden.

5.9 Definition (Substitutivität)
Eine Struktur $\mathcal{S} = (\mathcal{D}, \mathcal{I}, \mathcal{A})$ heißt **substitutiv**, *oder* **S-Struktur**, *falls für alle $x, y \in \mathbf{V}$, $\phi \in \mathbf{Form}^+$, $\beta : \mathbf{V} \to \mathcal{D}$ und $t \in \mathbf{Term}^+$ mit $\mathrm{fv}(t) \cap \mathrm{bv}(\epsilon x.\phi) = \emptyset$ gilt:*

$$\mathcal{A}([y/t](\epsilon x.\phi), \beta) = \mathcal{A}(\epsilon x.\phi, \beta\{y \leftarrow \mathrm{val}(\mathcal{S}, \beta, t)\})$$

Gilt eine Formel ϕ in allen substitutiven Strukturen unter allen Belegungen, so heißt sie **S-allgemeingültig**, *oder kurz* **S-gültig**, *in Zeichen $\models_S \phi$. Ist eine Formel in mindestens einer substitutiven Struktur unter mindestens einer Belegung gültig, so heißt sie* **S-erfüllbar**.

5.10 Lemma (Substitutionslemma)
Sei $\mathcal{S} = (\mathcal{D}, \mathcal{I}, \mathcal{A})$ eine substitutive Struktur, $\beta : \mathbf{V} \to \mathcal{D}$ eine Belegung und $\alpha \in \mathbf{Term}^+ \cup \mathbf{Form}^+$ ein Term oder eine Formel. Seien ferner $t \in \mathbf{Term}^+$ ein Term mit $\mathrm{fv}(t) \cap \mathrm{bv}(\alpha) = \emptyset$ und $y \in \mathbf{V}$ eine Objektvariable. Dann gilt:

$$\mathrm{val}(\mathcal{S}, \beta, [y/t](\alpha)) = \mathrm{val}(\mathcal{S}, \beta\{y \leftarrow \mathrm{val}(\mathcal{S}, \beta, t)\}, \alpha)$$

Beweis. Der Beweis verläuft wie für die klassische Prädikatenlogik. Der zusätzliche Fall für ϵ-Terme folgt direkt aus der Substitutivität. ■

5.11 Beispiel (Anwendung des Substitutionslemmas)
Als Anwendung des Substitutionslemmas soll gezeigt werden, daß in einer substitutiven Struktur, in der $a \doteq b$ gilt auch $\epsilon x.p(x,a) \doteq \epsilon x.p(x,b)$ wahr ist. Gilt $\mathcal{S} \models a \doteq b$ in einer Struktur $\mathcal{S} = (\mathcal{D}, \mathcal{I}, \mathcal{A})$, so ist laut Definition 5.7 $\mathrm{val}(\mathcal{S}, \beta, a) = \mathrm{val}(\mathcal{S}, \beta, b)$. Damit ergibt

[7] Das I soll dabei an die intensionale Semantik erinnern

sich

$$\begin{array}{ll}
\quad \mathrm{val}(\mathcal{S}, \beta, \epsilon x.p(x,a)) & \\
= \mathrm{val}(\mathcal{S}, \beta, [y/a](\epsilon x.p(x,y))) & \text{mit einer neuen Variable } y \in \mathbf{V} \\
= \mathrm{val}(\mathcal{S}, \beta\{y \leftarrow \mathrm{val}(\mathcal{S}, \beta, a)\}, \epsilon x.p(x,y)) & \text{mit dem Substitutionslemma} \\
= \mathrm{val}(\mathcal{S}, \beta\{y \leftarrow \mathrm{val}(\mathcal{S}, \beta, b)\}, \epsilon x.p(x,y)) & \text{da } \mathrm{val}(\mathcal{S}, \beta, a) = \mathrm{val}(\mathcal{S}, \beta, b) \\
= \mathrm{val}(\mathcal{S}, \beta, [y/b](\epsilon x.p(x,y))) & \text{mit dem Substitutionslemma} \\
= \mathrm{val}(\mathcal{S}, \beta, \epsilon x.p(x,b)) &
\end{array}$$

woraus mit Definition 5.7 wiederum $\mathcal{S} \models \epsilon x.p(x,a) \doteq \epsilon x.p(x,b)$ folgt.

Eine genaue Analyse der Fälle, in denen in den folgenden Aussagen über die ϵ-Logik das Substitutionslemma benötigt wird, zeigt, daß auch eine schwächere Form möglich gewesen wäre. Die später vorgestellten Kalkülregeln erzeugen aus geschlossenen Formeln der klassischen Prädikatenlogik immer nur ϵ-Terme ohne freie Objektvariablen. Für die Metavariablen werden Grundterme eingesetzt. Insgesamt werden dann also nur *Grund-ϵ-*Terme betrachtet, die von Substitutionen gar nicht berührt werden. Für solche ϵ-Terme ist die Substitutivität natürlich trivialerweise gegeben, und eine entsprechend eingeschränkte Form des Substitutionslemmas wäre auch ohne zusätzliche Forderungen an die Struktur gültig.

Die im folgenden beschriebenen Kalküle beziehen sich trotzdem auf substitutive Strukturen, und zwar aus drei Gründen. Erstens zeigt das Beispiel, daß gewisse Formeln, deren Gültigkeit der Mensch intuitiv erwartet nur in substitutiven Strukturen auch tatsächlich gültig sind. Zweitens soll dem Benutzer erlaubt werden, mit beliebigen Formeln und Termen zu arbeiten. Die Logik enthält die ϵ-Terme als neues Konstrukt, und jede unintuitive Einschränkung ihres Gebrauchs wird ihre Arbeitsweise schwerer verständlich machen. Drittens wird die formale Behandlung des Kalküls einfacher, da weniger Randbedingungen an das Vorkommen von Variablen oder die Position von zu ersetzenden Teiltermen zu berücksichtigen sind.

In der Motivierung der Substitutivitätseigenschaft wurden Aussagen angegeben, deren Gültigkeit intuitiv erwartet wird, die aber nur unter Hinzunahme der Substitutivität gültig sind. Analysiert man die zugrundeliegende Intuition im Beispiel 5.11, so kommt man aber nicht unbedingt zur Substitutivität. Aus $a \doteq b$ erwartet man, $\epsilon x.p(x,a) \doteq \epsilon x.p(x,b)$ folgern zu können, da die Formeln $p(x,a)$ und $p(x,b)$ *das gleiche bedeuten,* wenn man weiß, daß a und b gleich sind. Die gleiche Intuition würde also zur Annahme führen, daß etwa $(\epsilon x.(p(x) \wedge q(x))) \doteq (\epsilon x.(q(x) \wedge p(x)))$ allgemeingültig sei, da ja der $\wedge$-Operator kommutativ ist. Diese Formel ist jedoch nicht einmal S-allgemeingültig. Es ist jedoch möglich, diese Intuition zu formalisieren, und zu verlangen, daß zwei Terme $\epsilon x.\phi$ und $\epsilon x.\psi$ gleich ausgewertet werden, wenn die beiden Formeln ϕ und ψ semantisch äquivalent sind. Erlaubt man noch verschiedene ϵ-Variablen für die beiden Terme, gelangt man zu folgender Definition:

5.12 Definition (Extensionalität)
Einem ϵ-Term $\epsilon x.\phi$ wird in einer Struktur $\mathcal{S} = (\mathcal{D}, \mathcal{I}, \mathcal{A})$ unter einer Belegung $\beta : \mathbf{V} \to \mathcal{D}$ die **Extension**

$$\mathrm{Ext}(\mathcal{S}, \beta, \epsilon x.\phi) := \{d \in \mathcal{D} \mid \mathcal{S}, \beta\{x \leftarrow d\} \models \phi\}$$

zugeordnet. Eine Struktur $\mathcal{S} = (\mathcal{D}, \mathcal{I}, \mathcal{A})$ heißt **extensional**, *oder* **E-Struktur**, *falls für alle $x, y \in \mathbf{V}$, $\phi, \psi \in \mathbf{Form}^+$, $\beta : \mathbf{V} \to \mathcal{D}$ gilt:*

$$\begin{array}{rl}
\textit{Falls} & \mathrm{Ext}(\mathcal{S}, \beta, \epsilon x.\phi) = \mathrm{Ext}(\mathcal{S}, \beta, \epsilon y.\psi), \\
\textit{dann gilt} & \mathcal{A}(\epsilon x.\phi, \beta) = \mathcal{A}(\epsilon y.\psi, \beta)
\end{array}$$

Gilt eine Formel ϕ in allen extensionalen Strukturen unter allen Belegungen, so heißt sie **E-allgemeingültig***, oder kurz* **E-gültig***, in Zeichen $\models_E \phi$. Ist eine Formel in mindestens einer extensionalen Struktur unter mindestens einer Belegung gültig, so heißt sie* **E-erfüllbar**.

In extensionalen Strukturen sind tatsächlich jene Formeln gültig, von denen man dies aufgrund der oben beschriebenen Intuition annimmt. Eine äquivalente, aber anschaulichere Beschreibung der ϵ-Logik mit extensionaler Semantik nimmt von vornherein in die Strukturen anstelle der ϵ-Auswertung $\mathcal{A}$ eine *Auswahlfunktion* α auf. Diese soll zu jeder Teilmenge $M \subseteq \mathcal{D}$ der Trägermenge ein Element $\alpha(M) \in M$ auswählen. Für die leere Menge soll $\alpha(\emptyset) \in \mathcal{D}$ ein beliebiges Trägerelement liefern. Mit so definierten Strukturen $\mathcal{E} = (\mathcal{D}, \mathcal{I}, \alpha)$ kann man die Auswertung von ϵ-Termen durch

$$\mathrm{val}(\mathcal{E}, \beta, \epsilon x.\phi) = \alpha(\mathrm{Ext}(\mathcal{E}, \beta, \epsilon x.\phi))$$

festlegen.[8] Zu der Formel ϕ wird also die Menge der Werte für die freie Variable x genommen, die ϕ zusammen mit der Belegung für die anderen freien Variablen wahr macht. Aus dieser Menge wird dann mit α ein Element ausgewählt. Diese Definition ist insofern äquivalent zu unserer, als unter den zugehörigen Semantiken jeweils dieselben Formeln allgemeingültig, bzw. erfüllbar sind.

Der Leser mag sich fragen, warum die ϵ-Terme nicht von vornherein mit einer extensionalen Semantik versehen wurden, wenn dies mit weniger technischem Aufwand zu einem intuitiv leichter faßbaren Begriff führt als die substitutive Semantik. Die Antwort wird sich im Rahmen der Betrachtungen zu einem vollständigen Kalkül ergeben. Sowohl für die substitutive als auch für die extensionale Semantik wird ein vollständiger Kalkül angegeben werden. Bei extensionaler Semantik ist hierfür allerdings eine hochgradig indeterministische Regel erforderlich, die für die Verwendung durch einen automatischen Beweiser völlig ungeeignet ist. Beschränkt man sich hingegen auf die substitutive Semantik, kann die volle ϵ-Logik durch einen automatischen Beweiser ähnlich effizient vollständig behandelt werden wie die klassische Prädikatenlogik. Um über solche Differenzierungen diskutieren zu können, wurde den verschiedenen möglichen Semantiken ein gemeinsamer Rahmen zugrundegelegt.

Das folgende Lemma faßt das Verhältnis von intensionalen, substitutiven und extensionalen Strukturen zusammen:

5.13 Lemma (Hierarchielemma)

1. *Extensionale Strukturen sind substitutiv.*

2. *Sei $\phi \in$ **Form**$^+$. Wenn ϕ E-erfüllbar ist, ist ϕ S-erfüllbar. Wenn ϕ S-erfüllbar ist, ist ϕ I-erfüllbar.*

3. *Sei $\phi \in$ **Form**$^+$. Wenn ϕ I-allgemeingültig ist, ist ϕ S-allgemeingültig. Wenn ϕ S-allgemeingültig ist, ist ϕ E-allgemeingültig.*

[8] Diese Definition erfordert streng genommen eine Stufenkonstruktion wie im Beweis zu Satz 5.14, da die Gültigkeitsrelation für die Teilformel ϕ in der Definition von Ext verwendet wird. Da diese alternative Definition jedoch nicht weiter verwendet werden wird, soll auf dieses technische Detail verzichtet werden.

Beweis. Zu 1.: Man zeigt das Substitutionslemma für eine beliebige extensionale Struktur. Dessen Einschränkung auf Terme $\alpha = \epsilon x.\phi$ ergibt die Substitutivität der Struktur. Der Beweis des Substitutionslemmas verläuft wieder genau wie für die klassische Prädikatenlogik, und zwar durch Induktion über die Term- bzw. Formelstruktur von α. Der einzige hier interessante Fall ist der für ϵ-Terme. Sei also $\mathcal{S} = (\mathcal{D}, \mathcal{I}, \mathcal{A})$ eine extensionale Struktur, $\beta : \mathbf{V} \to \mathcal{D}$ eine Belegung und $\alpha = \epsilon x.\phi$ mit einer Variable $x \in \mathbf{V}$ und einer Formel $\phi \in \mathbf{Form}^+$. Seien ferner $t \in \mathbf{Term}^+$ ein Term mit $\mathrm{fv}(t) \cap \mathrm{bv}(\alpha) = \emptyset$ und $y \in \mathbf{V}$ eine Objektvariable. Das Substitutionslemma sei für Teilterme und Teilformeln von α gezeigt, insbesondere für ϕ. Falls $x = y$, gilt trivialerweise

$$\mathrm{val}(\mathcal{S}, \beta, [x/t](\epsilon x.\phi)) = \mathrm{val}(\mathcal{S}, \beta, \epsilon x.\phi) = \mathrm{val}(\mathcal{S}, \beta\{x \leftarrow \mathrm{val}(\mathcal{S}, \beta, t)\}, \epsilon x.\phi)$$

Falls $x \neq y$, ist

$$\begin{aligned}
& \mathrm{Ext}(\mathcal{S}, \beta, \epsilon x.[y/t](\phi)) \\
&= \{d \in \mathcal{D} \mid \mathcal{S}, \beta\{x \leftarrow d\} \models [y/t](\phi)\} \\
&= \{d \in \mathcal{D} \mid \mathcal{S}, \beta\{x \leftarrow d\}\{y \leftarrow \mathrm{val}(\mathcal{S}, \beta\{x \leftarrow d\}, t)\} \models \phi\} \\
&\qquad \text{laut Induktionshypothese} \\
&= \{d \in \mathcal{D} \mid \mathcal{S}, \beta\{y \leftarrow \mathrm{val}(\mathcal{S}, \beta, t)\}\{x \leftarrow d\} \models \phi\} \\
&\qquad \text{da } x \in \mathrm{bv}(\epsilon x.\phi), \text{ also } x \notin \mathrm{fv}(t). \\
&= \mathrm{Ext}(\mathcal{S}, \beta\{y \leftarrow \mathrm{val}(\mathcal{S}, \beta, t)\}, \epsilon x.\phi)
\end{aligned}$$

Da $\mathcal{S}$ extensional ist, muß also gelten:

$$\mathcal{A}(\epsilon x.[y/t](\phi), \beta) = \mathcal{A}(\epsilon x.\phi, \beta\{y \leftarrow \mathrm{val}(\mathcal{S}, \beta, t)\}),$$

und damit:

$$\begin{aligned}
& \mathrm{val}(\mathcal{S}, \beta, [y/t](\epsilon x.\phi)) \\
&= \mathrm{val}(\mathcal{S}, \beta, \epsilon x.[y/t](\phi)) \\
&= \mathcal{A}(\epsilon x.[y/t](\phi), \beta) \\
&= \mathcal{A}(\epsilon x.\phi, \beta\{y \leftarrow \mathrm{val}(\mathcal{S}, \beta, t)\}) \\
&= \mathrm{val}(\mathcal{S}, \beta\{y \leftarrow \mathrm{val}(\mathcal{S}, \beta, t)\}, \epsilon x.\phi)
\end{aligned}$$

Die Aussagen 2. und 3. sind triviale Folgerungen aus 1. und dem Umstand, daß definitionsgemäß alle substitutiven Strukturen auch Strukturen sind. ■

Das Hierarchielemma legt für ein integriertes Beweissystem folgende Sichtweise nahe: Der Benutzer kann von einer extensionalen Semantik ausgehen. Es wird ihm für diese ein vollständiger Kalkül angeboten. Der Automat benutzt nicht den vollen Kalkül, sondern nur einen Teil, der allerdings für die substitutive Semantik vollständig ist. Mit dem System können also alle E-gültigen Aussagen bewiesen werden, automatisch gefunden werden aber nur Beweise für Aussagen, die auch S-gültig sind.

Die eingeführten Eigenschaften von Strukturen, und die damit verbundenen Gültigkeitsbegriffe bilden natürlich in keiner Weise einen vollständigen Satz von möglichen Einschränkungen. Jede beliebige andere Einschränkung an die Form der Strukturen liefert eine weitere Gültigkeitsrelation. Hier wurden nur die im Zusammenhang dieser Arbeit wichtigsten Varianten vorgestellt. In Abschnitt 5.7 wird jedoch noch eine Variante vorgestellt, die die Stabilität gegenüber gebundener Umbenennung beinhaltet.

5.5. Einbettung der Prädikatenlogik in die ϵ-Logiken

Je stärker die an die Strukturen gestellten Forderungen sind, desto mehr Aussagen werden laut Hierarchielemma allgemeingültig. Soll ein Kalkül mit ϵ-Termen verwendet werden, um prädikatenlogische Formeln zu beweisen, muß sichergestellt werden, daß diese in der zugrundegelegten ϵ-Logik genau dann allgemeingültig sind, wenn sie im klassischen prädikatenlogischen Sinn allgemeingültig sind. Die ϵ-Logiken sollen also *konservative Erweiterungen* der klassischen Prädikatenlogik sein. Dies wird im folgenden Einbettungssatz gezeigt.

Nebenbei folgt aus diesem Satz auch die Existenz von Strukturen, die die verschiedenen geforderten Eigenschaften haben, ohne welche die jeweiligen Semantiken nicht sinnvoll wären.

5.14 Satz (Einbettungssatz)
Sei $\phi \in \mathbf{Form}^+$ eine Formel ohne ϵ-Terme.

1. *Die folgenden Aussagen sind äquivalent:*
   (a) *ϕ ist erfüllbar im Sinne der klassischen Prädikatenlogik.*[9]
   (b) *ϕ ist I-erfüllbar.*
   (c) *ϕ ist S-erfüllbar.*
   (d) *ϕ ist E-erfüllbar.*
2. *Die folgenden Aussagen sind äquivalent:*
   (a) *ϕ ist allgemeingültig im Sinne der klassischen Prädikatenlogik.*
   (b) *ϕ ist I-allgemeingültig.*
   (c) *ϕ ist S-allgemeingültig.*
   (d) *ϕ ist E-allgemeingültig.*

Beweis. 1.(a)⇒1.(d): Seien $\mathcal{S}_0 = (\mathcal{D}, \mathcal{I})$ eine prädikatenlogische Struktur und $\beta : \mathbf{V} \to \mathcal{D}$ eine Belegung mit $\mathcal{S}_0, \beta \models \phi$. Wir zeigen die Existenz einer extensionalen Struktur $\mathcal{S} = (\mathcal{D}, \mathcal{I}, \mathcal{A})$ mit $\mathcal{S}, \beta \models \phi$. Da ϕ keine ϵ-Terme enthält, hängt die Gültigkeit nicht von $\mathcal{A}$ ab. Zu zeigen ist also nur, daß überhaupt eine extensionale Struktur mit Trägermenge $\mathcal{D}$ und Interpretation $\mathcal{I}$ existiert.

Unter Verwendung des Auswahlaxioms dürfen wir die Existenz einer Funktion $\alpha : \mathcal{P}(\mathcal{D}) \to \mathcal{D}$ mit der Eigenschaft $\alpha(M) \in M$ für alle nicht-leeren Mengen $M \subseteq \mathcal{D}$ annehmen. Die ϵ-Auswertung $\mathcal{A}$ soll stufenweise konstruiert werden. Dazu seien die Mengen $F_i \subseteq \mathbf{Form}^+$ für $i \in \mathbb{N}$ definiert durch:

$$\begin{aligned} F_0 &:= \{\phi \in \mathbf{Form}^+ \mid \phi \text{ enthält keine } \epsilon\text{-Terme}\} \\ F_{i+1} &:= F_i \cup \{\phi \in \mathbf{Form}^+ \mid \phi \text{ enthält nur } \epsilon\text{-Terme } \epsilon x.\phi' \text{ mit } \phi' \in F_i\} \end{aligned}$$

Offenbar gilt $\mathbf{Form}^+ = \bigcup_{i=0}^{\infty} F_i$. Der Index der kleinsten dieser Mengen, die eine bestimmte Formel enthält, sei gegeben durch $\iota(\phi) := \min\{i \mid \phi \in F_i\}$. Nun wird eine

[9] Verschiedene Autoren verwenden für die Prädikatenlogik leicht unterschiedliche Begriffe. Mit „erfüllbar“ ist hier die Existenz einer Struktur und einer Belegung gemeint, unter der die Formel wahr ist. „Allgemeingültig“ ist eine Formel, wenn sie in allen Strukturen unter allen Belegungen wahr ist.

Familie von ϵ-Auswertungen $\mathcal{A}_i$ für $i \in \mathbb{N}$ definiert durch

$$\begin{aligned}\mathcal{A}_0(\epsilon x.\phi, \beta) &:= d_\perp \\ \mathcal{A}_{i+1}(\epsilon x.\phi, \beta) &:= \begin{cases} \alpha(\mathrm{Ext}_{\iota(\phi)}(\epsilon x.\phi)) & \text{falls } \phi \in F_i. \\ d_\perp & \text{sonst.}\end{cases}\end{aligned}$$

wobei $\mathrm{Ext}_i(\epsilon x.\phi) := \{d \in \mathcal{D} \mid (\mathcal{D}, \mathcal{I}, \mathcal{A}_i), \beta\{x \leftarrow d\} \models \phi\}$, und $d_\perp \in \mathcal{D}$ ein beliebiges Trägerelement ist. Mit Hilfe dieser Funktionen wird nun $\mathcal{A}$ als

$$\mathcal{A}(\epsilon x.\phi, \beta) := \mathcal{A}_{\iota(\phi)+1}(\epsilon x.\phi, \beta)$$

definiert. Damit ist $\mathcal{S} := (\mathcal{D}, \mathcal{I}, \mathcal{A})$ eine extensionale Struktur. Dazu zeigt man zunächst, daß für eine beliebige Formel $\phi \in F_k$ gilt:

$$(\mathcal{D}, \mathcal{I}, \mathcal{A}), \beta \models \phi \quad \text{gdw.} \quad (\mathcal{D}, \mathcal{I}, \mathcal{A}_k), \beta \models \phi. \qquad (\dagger)$$

Für $k = 0$ ist dies offenbar der Fall, da die Gültigkeit von ϕ dann nicht von der ϵ-Auswertung abhängt. Für $k > 0$ enthält ϕ höchstens ϵ-Terme $\epsilon x.\phi'$ mit $\phi' \in F_{k-1}$, also $\iota(\phi) \leq k-1$. Auf die Gültigkeit von ϕ in $(\mathcal{D}, \mathcal{I}, \mathcal{A})$ haben also nur die Werte von $\mathcal{A}(\epsilon x.\phi', \beta)$ für $\iota(\phi') \leq k-1$ einen Einfluß. Für diese gilt:

$$\begin{aligned}\mathcal{A}(\epsilon x.\phi', \beta) &= \mathcal{A}_{\iota(\phi')+1}(\epsilon x.\phi', \beta) \\ &= \alpha(\mathrm{Ext}_{\iota(\phi')}(\epsilon x.\phi')) && \text{da } \phi' \in F_{\iota(\phi')} \\ &= \mathcal{A}_k(\epsilon x.\phi', \beta) && \text{da } \phi' \in F_{k-1}.\end{aligned}$$

Damit folgt ($\dagger$). Damit läßt sich die Auswahleigenschaft wie folgt zeigen: Sei $\phi \in \mathbf{Form}^+$ beliebig.

$$\begin{array}{lll} & (\mathcal{D}, \mathcal{I}, \mathcal{A}), \beta \models \exists x.\phi & \\ \Longrightarrow & (\mathcal{D}, \mathcal{I}, \mathcal{A}_{\iota(\phi)}), \beta \models \exists x.\phi & \text{wegen } (\dagger) \\ \Longrightarrow & (\mathcal{D}, \mathcal{I}, \mathcal{A}_{\iota(\phi)}), \beta\{x \leftarrow d\} \models \phi \quad \text{für ein } d \in \mathcal{D} & \text{(Def. 5.6)} \\ \Longrightarrow & \mathrm{Ext}_{\iota(\phi)}(\epsilon x.\phi) \neq \emptyset & \text{(Def. von } \mathrm{Ext}_{\iota(\phi)}) \\ \Longrightarrow & \alpha(\mathrm{Ext}_{\iota(\phi)}(\epsilon x.\phi)) \in \mathrm{Ext}_{\iota(\phi)}(\epsilon x.\phi) & \text{(Konstruktion von } \alpha) \\ \Longrightarrow & \mathcal{A}_{\iota(\phi)+1}(\epsilon x.\phi, \beta) \in \mathrm{Ext}_{\iota(\phi)}(\epsilon x.\phi) & \text{(Def. von } \mathcal{A}_{\iota(\phi)+1}) \\ \Longrightarrow & (\mathcal{D}, \mathcal{I}, \mathcal{A}_{\iota(\phi)}), \beta\{x \leftarrow \mathcal{A}_{\iota(\phi)+1}(\epsilon x.\phi, \beta)\} \models \phi & \text{(Def. von } \mathrm{Ext}_{\iota(\phi)}.) \\ \Longrightarrow & (\mathcal{D}, \mathcal{I}, \mathcal{A}_{\iota(\phi)}), \beta\{x \leftarrow \mathcal{A}(\epsilon x.\phi, \beta)\} \models \phi & \text{(Def. von } \mathcal{A}) \\ \Longrightarrow & (\mathcal{D}, \mathcal{I}, \mathcal{A}), \beta\{x \leftarrow \mathcal{A}(\epsilon x.\phi, \beta)\} \models \phi & \text{wegen } (\dagger) \end{array}$$

Für die Extensionalität seien $\phi, \psi \in \mathbf{Form}^+$ und $x, y \in \mathbf{V}$ beliebig. Es gilt:

$$\begin{array}{lll} & \mathrm{Ext}(\mathcal{S}, \beta, \epsilon x.\phi) = \mathrm{Ext}(\mathcal{S}, \beta, \epsilon y.\psi) & \\ \Longrightarrow & \quad \{d \in \mathcal{D} \mid (\mathcal{D}, \mathcal{I}, \mathcal{A}), \beta\{x \leftarrow d\} \models \phi\} & \\ & = \{d \in \mathcal{D} \mid (\mathcal{D}, \mathcal{I}, \mathcal{A}), \beta\{y \leftarrow d\} \models \psi\} & \\ \Longrightarrow & \quad \mathrm{Ext}_{\iota(\phi)}(\epsilon x.\phi) & \\ & = \{d \in \mathcal{D} \mid (\mathcal{D}, \mathcal{I}, \mathcal{A}_{\iota(\phi)}), \beta\{x \leftarrow d\} \models \phi\} & \\ & = \{d \in \mathcal{D} \mid (\mathcal{D}, \mathcal{I}, \mathcal{A}_{\iota(\psi)}), \beta\{y \leftarrow d\} \models \psi\} & \text{wegen } (\dagger) \\ & = \mathrm{Ext}_{\iota(\psi)}(\epsilon y.\psi) & \\ \Longrightarrow & \mathcal{A}_{\iota(\phi)+1}(\epsilon x.\phi, \beta) = \mathcal{A}_{\iota(\psi)+1}(\epsilon y.\psi, \beta) & \\ \Longrightarrow & \mathcal{A}(\epsilon x.\phi, \beta) = \mathcal{A}(\epsilon y.\psi, \beta) & \text{wegen } (\dagger) \end{array}$$

1.(d)⇒1.(c) und **1.(c)⇒1.(b)** folgen direkt aus dem Hierarchielemma 5.13.

1.(b)⇒1.(a): Da ϕ keine ϵ-Terme enthält, ist die Gültigkeit von ϕ nicht von der ϵ-Auswertung abhängig. Aus $(\mathcal{D}, \mathcal{I}, \mathcal{A}), \beta \models \phi$ folgt also $(\mathcal{D}, \mathcal{I}), \beta \models \phi$ im Sinne der Prädikatenlogik.

Zu 2.: Die Aussagen für die Allgemeingültigkeit folgen aus 1. und der Tatsache, daß für alle vier Semantiken eine Formel ϕ genau dann allgemeingültig ist, wenn ihre Negation $\neg\phi$ unerfüllbar ist. ■

5.6. Grundkalküle für die ϵ-Logik

Es sollen nun zwei Grundkalküle für die ϵ-Logik vorgestellt werden. Metavariablen werden aus zwei Gründen vorerst nicht benutzt: erstens würden diese das Vollständigkeitsargument komplizieren, und zweitens ist für die Gleichheitsbehandlung mit Metavariablen das Konzept der Constraints nötig, welches erst im folgenden Kapitel eingeführt wird. Hier soll also mit Sequenzen gearbeitet werden, in denen alle Formeln geschlossen sind, und die außerdem keine Metavariablen enthalten. Die für die Grundkalküle gezeigten Eigenschaften sollen dann später durch *lifting* auf die Kalküle mit Metavariablen übertragen werden.

In gewisser Weise stellen die Grundkalküle die Benutzersicht der Kalküle mit Metavariablen dar: Wenn der Benutzer in den Kalkülen mit Metavariablen für diese sofort bei ihrer Einführung Belegungen angibt, und das System, wie in 4.5 vorgeschlagen, statt der Metavariablen diese Terme anzeigt, so wird es für ihn so aussehen, als arbeite er mit dem entsprechenden Grundkalkül.

Zuerst wird der Kalkül $\mathcal{G}_S$ vorgestellt, der für die substitutive Semantik ohne Gleichheit vollständig ist, und der für die Verwendung durch einen automatischen Beweiser geeignet ist, wenn man zusätzlich Metavariablen einführt. Es ist möglich, zu diesem Kalkül eine Gleichheitsregel hinzuzunehmen, um dann einen auch für die substitutive Semantik mit Gleichheit vollständigen Kalkül zu erhalten. Davon wurde jedoch abgesehen, da der Vollständigkeitsbeweis ohnehin mit erheblichem technischem Aufwand verbunden ist.

Danach werden zu diesem Kalkül Gleichheitsregeln, sowie eine Extensionalitätsregel hinzugenommen, und gezeigt, daß der erweiterte Kalkül $\mathcal{G}_E$ vollständig für die extensionale Semantik ist. Es wird sich jedoch zeigen, daß die Extensionalitätsregel für die automatische Benutzung ungeeignet ist.

5.6.1. Der Kalkül $\mathcal{G}_S$

Die vorgestellten Kalküle arbeiten auf *Grundsequenzen*, die hier definiert werden sollen.

5.15 Definition (Grundsequenzen)
Eine **Grundsequenz** *ist ein als* $\Gamma \vdash \Delta$ *geschriebenes geordnetes Paar, wobei* $\Gamma, \Delta \subseteq$ $\mathbf{Form}^0$. Γ *heißt* **Antezedent** *und* Δ **Sukzedent** *der Sequenz. Die Mengenklammern für* Γ *und* Δ *werden weggelassen, wenn Sequenzen explizit angegeben werden. Ferner ist mit der Schreibweise* $\phi, \Gamma \vdash \Delta$ *eine Sequenz* $\Gamma' \vdash \Delta$ *gemeint mit* $\Gamma' = \{\phi\} \cup \Gamma$*, und für den Sukzedenten entsprechend.*

Um dem Benutzer einen möglichst freien Umgang mit dem System zu ermöglichen, sollte der Kalkül auch vollständig sein, wenn der Benutzer selber – etwa in Lemmata oder *cut*-Formeln – beliebige ϵ-Terme eingeben kann, also ϵ-Terme vorkommen, die nicht durch eine der Beweisregeln eingeführt werden. Etwa soll die S-gültige Formel $\exists x.(p(x) \wedge q(x)) \rightarrow q(\epsilon x.q(x))$ beweisbar sein. Unter Verwendung der *cut*-Regel des Sequenzenkalküls für die Formel $\exists x.q(x)$ wäre dafür folgender Beweis möglich:

$$
\dfrac{
\dfrac{
\dfrac{
\dfrac{p(\mathbf{c}), q(\mathbf{c}) \vdash q(\mathbf{c})}{p(\mathbf{c}), q(\mathbf{c}) \vdash \exists x.q(x)}\ \exists r
}{p(\mathbf{c}) \wedge q(\mathbf{c}) \vdash \exists x.q(x)}\ \wedge l
}{\exists x.(p(x) \wedge q(x)) \vdash \exists x.q(x)}\ \exists l
\quad
\dfrac{q(\epsilon x.q(x)) \vdash q(\epsilon x.q(x))}{\exists x.q(x) \vdash q(\epsilon x.q(x))}\ \exists l
}{\exists x.(p(x) \wedge q(x)) \vdash q(\epsilon x.q(x))}\ cut
$$

wobei **c** eine Abkürzung für $\epsilon x.(p(x) \wedge q(x))$ sein soll. Die *cut*-Regel ist jedoch zu indeterministisch, um für ein automatisches Beweissystem geeignet zu sein. Man kann aber für diese spezielle Verwendung der *cut*-Regel eine neue Regel formulieren, die den Namen ϵ-*cut* erhalten soll:

$$\frac{\Gamma \vdash \exists x.\phi, \Delta \quad [x/\epsilon x.\phi](\phi), \Gamma \vdash \Delta}{\Gamma \vdash \Delta} \ \epsilon\text{-}cut$$

Dabei ist $\epsilon x.\phi \in \mathbf{Term}^0$ ein Grund-ϵ-Term, $\exists x.\phi$ also eine geschlossene Formel. Diese Regel entspricht der Auswahleigenschaft: im linken Zweig muß gezeigt werden, daß ein x existiert, für welches ϕ wahr wird. Im rechten Zweig kann dann verwendet werden, daß ϕ insbesondere für $\epsilon x.\phi$ wahr wird. Nun soll diese Regel noch so eingeschränkt werden, daß $\epsilon x.\phi$ tatsächlich in der unteren Sequenz vorkommen muß. Was „vorkommen" in diesem Zusammenhang bedeutet, soll durch die folgende Definition verdeutlicht werden.

5.16 Definition (relevante Grundterme, GT**)**
Für $t \in \mathbf{Term}^+$ sei $\mathrm{GT}'(t) \subseteq \mathbf{Term}^0$ die Menge der Teilterme von t, die Grundterme sind:

$$\mathrm{GT}'(t) := \{t' \in \mathbf{Term}^0 \mid t' \textit{ ist Teilterm von } t\}.$$

Für eine Grundsequenz $\Gamma \vdash \Delta$ sei

$$\mathrm{GT}'(\Gamma \vdash \Delta) := \bigcup_{p(t_1,\ldots,t_n)\in\Gamma\cup\Delta} GT'(t_i)$$

$$\mathrm{GT}(\Gamma \vdash \Delta) := \begin{cases} \mathrm{GT}'(\Gamma \vdash \Delta) & \textit{falls } \mathrm{GT}'(\Gamma \vdash \Delta) \neq \emptyset, \\ \{c\} & \textit{sonst,} \end{cases}$$

wobei c ein beliebiges Konstantensymbol ist.

Es ist zu bemerken, daß aufgrund der ϵ-Terme nicht jeder Teilterm eines Grundterms ebenfalls Grundterm sein muß. $\mathrm{GT}(\Sigma)$ ist die Menge der Grundterme, die in *atomaren* Formeln aus Σ vorkommen.[10] Sei zum Beispiel eine Sequenz $\Gamma \vdash \Delta = p(a,a) \wedge q(a) \vdash p(f(b,b), \epsilon x.q(f(x,c)))$ gegeben. Dann ist $\mathrm{GT}(\Sigma) = \{f(b,b), b, \epsilon x.q(f(x,c)), c\}$. Die Definition stellt sicher, daß $\mathrm{GT}(\Gamma \vdash \Delta)$ nie leer ist, was in Sonderfällen für die Vollständigkeit wichtig sein kann.

Wie gezeigt werden soll, reicht es für die Vollständigkeit, die ϵ-*cut*-Regel für Terme $\epsilon x.\phi \in \mathrm{GT}(\Gamma \vdash \Delta)$ anzuwenden. Auch für die Quantor-Instantiierung brauchen nur Terme aus $\mathrm{GT}(\Gamma \vdash \Delta)$ herangezogen werden. Diese Einschränkungen führen zum Kalkül $\mathcal{G}_S$, dessen Regeln in Abbildung 5.2 vorgestellt werden. Wie man leicht sieht, kann diese Einschränkung sogar noch weiter verschärft werden: Stammt ein ϵ-Term aus einer Anwendung der $\exists l$-Regel, braucht die ϵ-*cut*-Regel für diesen Term nicht angewandt werden. Ihre Anwendung ist also nur für ϵ-Terme erforderlich, die im Problem selber vorkommen. Zur Vereinfachung der Darstellung wurde auf die formale Berücksichtigung dieser Einschränkung verzichtet.

Die Definition des Beweisbegriffs unterscheidet sich geringfügig von dem für den Sequenzenkalkül üblichen: Axiome werden nicht *angewandt*, da dies später beim Kalkül mit Metavariablen auch nicht geschehen soll.

[10] Gleichungen brauchen hier nicht berücksichtigt zu werden.

Abbildung 5.2.: Beweisregeln des Kalküls $\mathcal{G}_S$

$$\frac{\phi, \psi, \Gamma \vdash \Delta}{\phi \wedge \psi, \Gamma \vdash \Delta} \wedge l \qquad \frac{\Gamma \vdash \phi, \Delta \quad \Gamma \vdash \psi, \Delta}{\Gamma \vdash \phi \wedge \psi, \Delta} \wedge r$$

$$\frac{\Gamma \vdash \phi, \Delta}{\neg\phi, \Gamma \vdash \Delta} \neg l \qquad \frac{\phi, \Gamma \vdash \Delta}{\Gamma \vdash \neg\phi, \Delta} \neg r$$

$$\frac{[x/\epsilon x.\phi](\phi), \Gamma \vdash \Delta}{\exists x.\phi, \Gamma \vdash \Delta} \exists l \qquad \frac{\Gamma \vdash [x/t](\phi), \exists x.\phi, \Delta}{\Gamma \vdash \exists x.\phi, \Delta} \exists r$$

wobei $t \in \mathrm{GT}(\Gamma \vdash \Delta)$.

$$\frac{\Gamma \vdash \exists x.\phi, \Delta \quad [x/\epsilon x.\phi](\phi), \Gamma \vdash \Delta}{\Gamma \vdash \Delta} \epsilon\text{-}cut$$

wobei $\epsilon x.\phi \in \mathrm{GT}(\Gamma \vdash \Delta)$.

5.17 Definition ($\mathcal{G}_S$-Beweis)
Ein $\mathcal{G}_S$-**Beweis** *für eine geschlossene Formel* $\phi \in \mathbf{Form}^0$ *ist ein Baum, dessen Knoten mit Sequenzen markiert sind, so daß*

- *der Wurzelknoten mit der Sequenz* $\vdash \phi$ *markiert ist, und*
- *für jeden mit* Σ *markierten inneren Knoten, dessen Nachkommen mit* $\Sigma_1, \ldots, \Sigma_k$ *markiert sind,*

$$\frac{\Sigma_1 \quad \ldots \quad \Sigma_k}{\Sigma}$$

  Instanz einer der Regeln von $\mathcal{G}_S$ *ist.*[11]

Ein $\mathcal{G}_S$-**Axiom** *ist eine Sequenz* $\Gamma \vdash \Delta$ *mit* $p(t_1, \ldots, t_n) \in \Gamma \cap \Delta$ *für eine atomare Formel* $p(t_1, \ldots, t_n)$.

Ein $\mathcal{G}_S$*-Beweis heißt* **geschlossen**, *falls er endlich ist, und jedes Blatt mit einem* $\mathcal{G}_S$*-Axiom markiert ist.*

Der Kalkül G_S ist korrekt und vollständig für Formeln ohne Gleichheit und substitutive Semantik.

5.18 Satz (Korrektheit von $\mathcal{G}_S$)
Falls für eine Formel $\phi \in \mathbf{Form}^0$ *ein geschlossener* $\mathcal{G}_S$*-Beweis existiert, ist* ϕ *S-allgemeingültig.*

Die Korrektheit soll hier nicht bewiesen werden. Erstens ist sie nicht schwer zu zeigen. Zweitens wird dieser Kalkül nur als Hilfsmittel eingeführt, um später die Vollständigkeit des erweiterten Kalküls mit Metavariablen zeigen zu können. Dessen Korrektheit wird in Satz 6.8 gezeigt.

5.19 Satz (Vollständigkeit von $\mathcal{G}_S$)
Sei $\phi \in \mathbf{Form}^0$ *eine geschlossene, S-allgemeingültige Formel, in der die Gleichheit* $\doteq$ *nicht vorkommt. Dann existiert ein geschlossener* $\mathcal{G}_S$*-Beweis für* ϕ.

[11] Der Begriff *Instanz* soll hier nicht näher erläutert werden, da dies derselbe Begriff wie in vielen anderen Kalkülen ist.

Der Beweis dieses Satzes ist Inhalt des folgenden Abschnitts.

5.6.2. Beweis der Vollständigkeit von $\mathcal{G}_S$

Der Beweis basiert auf dem Beweis der Vollständigkeit des Tableaukalküls mit Hintikka-Mengen. Dabei treten jedoch einige technische Schwierigkeiten auf, weshalb der Beweis hier sehr detailliert dargelegt wird. Diese Detailliertheit ist vor allem deswegen wichtig, weil durch die ϵ-Terme syntaktische Eigenschaften, wie etwa die Substitutivität, die sonst als selbstverständlich hingenommen werden, nicht mehr unbedingt gelten müssen. Bei der Suche nach dem Beweis scheiterten mehrere Versuche an solchen technischen Details. Es ist jedenfalls empfehlenswert, sich vor der Lektüre den Beweis für die klassische Prädikatenlogik ins Gedächtnis zu rufen.

Die Grundidee eines Vollständigkeitsbeweises mit Hintikka-Mengen besteht darin, ausgehend von der Annahme, es gäbe *keinen* Beweis für eine Formel ϕ, eine Struktur zu konstruieren, in welcher ϕ nicht gilt, und damit einen Widerspruch zur Allgemeingültigkeit von ϕ herzustellen.

Für die klassische Prädikatenlogik ohne Gleichheit wird meist die Menge aller Grundterme als Trägermenge gewählt, und die Interpretation so festgelegt, daß jeder Grundterm zu sich selbst ausgewertet wird. Verzichtet man auf die Einschränkungen der ϵ-*cut*-Regel und der $\exists r$-Regel auf relevante Grundterme, so läßt sich diese Methode recht problemlos auf die ϵ-Logik mit substitutiver Semantik übertragen. Die weiter unten benutzte Bedingung (H5) gilt dann nämlich für *jeden* ϵ-Term, und nicht nur solche aus $\mathrm{GT}(H)$. Das macht es sehr leicht, die Auswahleigenschaft zu zeigen. Die Substitutivität ist trivial, da die Terme durch Substitution ausgewertet werden.

Die Einschränkung, Quantoren nur mit bereits in der Sequenz vorkommenden Grundtermen zu instantiieren, läßt sich in der klassischen Prädikatenlogik in den Griff bekommen, indem man als Trägermenge einfach alle relevanten Grundterme nimmt. Diese werden dann immer noch zu sich selbst ausgewertet, und es stellt sich heraus, daß es auf die Auswertung von anderen Termen nicht ankommt.

Leider ist diese Änderung für die ϵ-Logik nicht so einfach: Es soll ja eine S-Struktur erzeugt werden, die die Substitutivitäts- und Auswahleigenschaft für *alle* ϵ-Terme hat, und nicht nur für diejenigen, die in der Hintikka-Menge vorkommen. Es ist für das Beweisprinzip wesentlich, daß die relevanten Grundterme, die im Beweis vorkommen, zu sich selbst ausgewertet werden. Angenommen, t ist ein beliebiger Term, der in der konstruierten Struktur zu einem Grundterm a ausgewertet wird. In einer substitutiven Struktur müssen dann die Terme $f(t)$ und $f(a)$ gleich ausgewertet werden, obwohl möglicherweise beide nicht in der Hintikka-Menge vorkommen. Um dies zu ermöglichen, werden die folgenden syntaktischen Begriffe eingeführt.

5.20 Definition (generalisierter Term)
Ein Term oder eine Formel $\alpha \in \mathbf{Form}^+ \cup \mathbf{Term}^+$ *heißt* **generalisiert** *bzgl. einer Menge von Variablen* $M \subseteq \mathbf{V}$*, falls*

- $\alpha = p(t_1, \ldots, t_n)$*, und* $t_1, \ldots, t_n$ *sind generalisiert bzgl.* M*, oder*
- $\alpha = \neg\phi$*, und* ϕ *ist generalisiert bzgl.* M*, oder*
- $\alpha = \phi_1 \wedge \phi_2$*, und* ϕ_1, ϕ_2 *sind generalisiert bzgl.* M*, oder*
- $\alpha = \exists x.\phi$*, und* ϕ *ist generalisiert bzgl.* $M \cup \{x\}$*, oder*

- $\alpha = x \in \mathbf{V}$, *oder*
- $\alpha = f(t_1, \ldots, t_n)$, *und* $\mathrm{fv}(\alpha) \cap M \neq \emptyset$, *und* $t_1, \ldots, t_n$ *sind generalisiert bzgl.* M, *oder*
- $\alpha = \epsilon x.\phi$, *und* $\mathrm{fv}(\alpha) \cap M \neq \emptyset$, *und* ϕ *ist generalisiert bzgl.* $M \cup \{x\}$.

α *heißt generalisiert bzgl.* x, *falls* α *generalisiert bzgl.* $\{x\}$ *ist.*

Es ist zum Beispiel $p(x, \epsilon y.p(f(y), x))$ generalisiert bzgl. x, da die Variable x in dem Term $\epsilon y.p(f(y), x)$ frei vorkommt. Dagegen ist die Formel $\phi = p(x, \epsilon y.p(f(y), z))$ ist *nicht* generalisiert bzgl. x, da $t = \epsilon y.p(f(y), z)$ als Teilterm vorkommt, das freie Vorkommen der Variable z in t auch in ϕ ein freies Vorkommen ist, und x in t nicht frei vorkommt.

Für den Vollständigkeitsbeweis wird folgende Aussage gebraucht:

> Für jede Formel und jeden Term $\alpha \in \mathbf{Term}^+ \cup \mathbf{Form}^+$ und jede Menge $M \subseteq \mathbf{V}$ existiert ein $\bar{\alpha} \in \mathbf{Term}^+ \cup \mathbf{Form}^+$ und eine Substitution $[v_i/g_i]$, so daß $\alpha = [v_i/g_i]\bar{\alpha}$, wobei $\bar{\alpha}$ generalisiert ist bzgl. M, und die Variablen v_i nicht in α vorkommen. $\bar{\alpha}$ und $[v_i/g_i]$ sind bis auf Umbenennung der Variablen v_i eindeutig bestimmt. $\bar{\alpha}$ heißt **Generalisierung** von α bzgl. M.

Es ist zum Beispiel $p(x, v_1)$ die Generalisierung bzgl. x von $p(x, \epsilon y.p(f(y), z))$, wobei $g_1 = \epsilon y.p(f(y), z)$. Die Generalisierung von $p(f(x, a), \epsilon y.q(f(a, b), x, y))$ bzgl. x lautet $p(f(x, v_1), \epsilon y.q(v_2, x, y))$ mit $g_1 = a$ und $g_2 = f(a, b)$. Eine wichtige Eigenschaft der Generalisierung ist die folgende: Seien $\alpha \in \mathbf{Term}^+ \cup \mathbf{Form}^+$, $t \in \mathbf{Term}^+$, $y \in \mathbf{V}$ mit $\mathrm{fv}(t) \cap \mathrm{bv}(\alpha) = \emptyset$. Ist $\bar{\alpha}$ Generalisierung von α bzgl. x, und $\alpha = [v_i/g_i](\bar{\alpha})$, so ist $\bar{\alpha}$ auch Generalisierung von $[y/t](\alpha)$ bzgl. x, da $\bar{\alpha}$ bzgl. x generalisiert ist,

$$[y/t](\alpha) = [y/t]([v_i/g_i]\bar{\alpha}) = [y/t, v_i/[y/t](g_i)](\bar{\alpha})$$

und die Generalisierung bis auf Variablenumbenennung eindeutig ist. Nun soll der eigentliche Beweis geführt werden.

Angenommen ϕ sei S-gültig, aber kein $\mathcal{G}_S$-Beweis für ϕ sei geschlossen. Durch (faire) Anwendung der Grund-Beweisregeln in Abb. 5.2 kann eine Kette[12] von Beweisbäumen für ϕ erzeugt werden. Dabei soll jede auf eine Formel in einer Sequenz anwendbare Regel irgendwann angewandt werden. Insbesondere soll jede $\exists$-Formel rechts auf jedem Ast mit jedem relevanten Grundterm instantiiert werden. Die ϵ-cut Regel soll auf alle relevanten Grundterme irgendwann angewandt werden. Dies ist möglich, da jede atomare Formel und jede $\exists$-Formel rechts, die einmal in einer Sequenz vorkommt, auch in allen durch Regelanwendungen aus dieser generierten Sequenzen auftritt.

Wenn es keinen Beweis für die Formel gibt, so wird keiner der Beweise in dieser Kette geschlossen sein. Insbesondere wird ein (möglicherweise unendlicher) Beweisbaum approximiert, der mindestens einen Ast $(\Gamma_i \vdash \Delta_i)$ hat, so daß keine der Sequenzen $\Gamma_i \vdash \Delta_i$ ein Axiom ist. Man definiert die Menge $H \subseteq \mathbf{Form}^0$ durch:

$$H := \{\phi \mid \phi \in \bigcup_i \Gamma_i\} \cup \{\neg\psi \mid \psi \in \bigcup_i \Delta_i\}.$$

[12] Die Bäume bilden mit der Teil-Baum-Ordnung einen vollständige Partialordnung. Unter einer Kette ist eine bezüglich dieser Ordnung total geordnete Menge von Bäumen zu verstehen. Der von einer Kette approximierte Beweisbaum ist das Supremum dieser Menge.

Entsprechend Definition 5.16 seien

$$\mathrm{GT}'(H) := \bigcup_{p(t_1,\ldots,t_n)\in H} GT'(t_i) \cup \bigcup_{\neg p(t_1,\ldots,t_n)\in H} GT'(t_i)$$

$$\mathrm{GT}(H) := \begin{cases} \mathrm{GT}'(H) & \text{falls } \mathrm{GT}'(H) \neq \emptyset, \\ \{c\} & \text{sonst.} \end{cases}$$

Es ist dann offenbar $\mathrm{GT}(H) = \bigcup_i \mathrm{GT}(\Gamma_i \vdash \Delta_i)$. H erfüllt die folgenden Bedingungen:

(H0) Für kein Atom $\phi = p(t_1, \ldots, t_n)$ ist $\phi \in H$ und $\neg\phi \in H$, da sonst der Ast abgeschlossen wäre.

(H1) Für jede Formel $\phi = \phi_1 \wedge \phi_2 \in H$ ist $\phi_1 \in H$ und $\phi_2 \in H$, wegen der Regel $\wedge l$.

(H2) Für jede Formel $\phi = \neg(\phi_1 \wedge \phi_2) \in H$ ist $\neg\phi_1 \in H$ oder $\neg\phi_2 \in H$, wegen der Regeln $\wedge l$, $\neg l$, $\neg r$.

(H3) Für jede Formel $\phi = \exists x.\phi \in H$ ist $[x/\epsilon x.\phi](\phi) \in H$, wegen der Regel $\exists l$.

(H4) Für jede Formel $\phi = \neg\exists x.\phi \in H$ und jeden Term $t \in \mathrm{GT}(H)$ ist $[x/t](\neg\phi) \in H$, wegen der Regeln $\exists r$, $\neg l$, $\neg r$.

(H5) Für jeden ϵ-Term $\epsilon x.\phi \in \mathrm{GT}(H)$ ist $[x/\epsilon x.\phi](\phi) \in H$ oder $\neg\exists x.\phi \in H$, wegen der Regel ϵ-*cut*.

(H6) Für jeden Term $\neg\neg\phi \in H$ ist $\phi \in H$, wegen der Regeln $\neg l$ und $\neg r$.

Es soll nun eine substitutive Struktur $\mathcal{S} = (\mathcal{D}, \mathcal{I}, \mathcal{A})$ definiert werden, die alle Formeln aus H wahr macht. Da $\Delta_0 = \{\phi\}$, also $\neg\phi \in H$, gilt dann insbesondere $\mathcal{S} \not\models \phi$, was im Widerspruch zur Allgemeingültigkeit von ϕ steht. Der Beweis benutzt eine Stufenkonstruktion.

Zunächst wird die Menge $E(t)$ der in einem Term t vorkommenden ϵ-Terme definiert durch

$$\begin{aligned} E(x) &:= \emptyset, \\ E(f(t_1, \ldots, t_n)) &:= \textstyle\bigcup_{i=1}^n E(t_i), \\ E(\epsilon x.\phi) &:= \{\epsilon x.\phi\}. \end{aligned}$$

ϵ-Terme, die in anderen ϵ-Termen geschachtelt vorkommen, werden also nicht betrachtet. Die Menge $T(\phi)$ der in einer Formel ϕ vorkommenden Terme wird definiert durch:

$$\begin{aligned} T(p(t_1, \ldots, t_n)) &:= \{t_1, \ldots, t_n\}, \\ T(\neg\phi) &:= T(\phi), \\ T(\phi_1 \wedge \phi_2) &:= T(\phi_1) \cup T(\phi_2), \\ T(\exists x.\phi) &:= T(\phi). \end{aligned}$$

Auch hier sind also nur die Terme auf der „obersten Ebene“ gemeint, also nicht Terme in Formeln in ϵ-Termen, oder dergleichen. Damit werden nun Mengen $T_i \subseteq \mathbf{Term}^+$ und $F_i \subseteq \mathbf{Form}^+$ definiert:

$$\begin{aligned} T_0 &:= \{t \in \mathbf{Term}^+ \mid E(t) \subseteq \mathrm{GT}(H)\}, \\ T_{i+1} &:= T_i \cup \{t \in \mathbf{Term}^+ \mid E(t) \subseteq \{\epsilon x.\phi \mid \phi \in F_i\}\} && \text{für } i \geq 0, \\ F_i &:= \{\phi \in \mathbf{Form}^+ \mid T(\phi) \subseteq T_i\} && \text{für } i \geq 0. \end{aligned}$$

Offenbar gilt $\mathrm{GT}(H) \subseteq T_0$, sowie $\mathbf{Term}^+ = \bigcup_i T_i$ und $\mathbf{Form}^+ = \bigcup_i F_i$. Seien $\iota(t) := \min\{i \mid t \in T_i\}$ und $\iota(\phi) := \min\{i \mid \phi \in F_i\}$ für beliebige Terme t, bzw. Formeln ϕ.

Sei $\alpha : \mathcal{P}(\mathrm{GT}(H)) \to \mathrm{GT}(H)$ eine beliebige Auswahlfunktion mit $\alpha(A) \in A$ für alle nicht-leeren Mengen $A \subseteq \mathrm{GT}(H)$.Man definiert induktiv:

$$\mathcal{D} := \mathrm{GT}(H)$$

$$\mathcal{I}(f)(t_1,\dots,t_n) := \begin{cases} f(t_1,\dots,t_n) & \text{falls } f(t_1,\dots,t_n) \in \mathrm{GT}(H), \\ t_\perp & \text{sonst.} \end{cases}$$

$$\mathcal{I}(p)(t_1,\dots,t_n) := \begin{cases} \mathit{tt} & \text{falls } p(t_1,\dots,t_n) \in H, \\ \mathit{ff} & \text{sonst.} \end{cases}$$

$$\mathcal{A}_0(\epsilon x.\phi,\beta) := \begin{cases} \epsilon x.\phi & \text{falls } \epsilon x.\phi \in \mathrm{GT}(H), \\ t_\perp & \text{sonst.} \end{cases}$$

$$\mathcal{A}_{i+1}(\epsilon x.\phi,\beta) := \begin{cases} \mathcal{A}'_{i+1}(\epsilon x.\bar\phi,\bar\beta) & \text{falls } \epsilon x.\phi \in T_{i+1}, \\ t_\perp & \text{sonst,} \end{cases}$$

wobei $\bar\phi$ die Generalisierung von ϕ bzgl. x ist, $\phi = [v_i/g_i](\bar\phi)$, und $\bar\beta = \beta\{v_i \leftarrow \mathrm{val}(\mathcal{S}_{\iota(g_i)},\beta,g_i)\}$. (†)

$$\mathcal{A}'_{i+1}(\epsilon x.\bar\phi,\bar\beta) := \begin{cases} [\bar\beta](\epsilon x.\bar\phi) & \text{falls } [\bar\beta](\epsilon x.\bar\phi) \in \mathrm{GT}(H), \\ \alpha(\mathrm{Ext}(\mathcal{S}_{\iota(\bar\phi)},\bar\beta,\epsilon x.\bar\phi)) & \text{sonst. (‡)} \end{cases}$$

$$\mathcal{S}_i := (\mathcal{D},\mathcal{I},\mathcal{A}_i)$$

Dabei ist $t_\perp \in \mathrm{GT}(H)$ ein beliebiges Trägerelement, und die Substitution $[\bar\beta]$ wird definiert durch $[\bar\beta](x) := \bar\beta(x)$ für $x \in \mathrm{fv}(\bar\phi)$, was möglich ist, da es sich bei den Trägerelementen um Terme handelt. Man beachte, daß dies eine zulässige induktive Definition ist. Die Induktion tritt an den Stellen (†) und (‡) auf. Wegen der Bedingung $\epsilon x.\phi \in T_{i+1}$, und da es sich um Teilterme, bzw. -Formeln des ϵ-Terms handelt, sind $\iota(g_i), \iota(\bar\phi) \le i$. Diese Definitionen sind so gewählt, daß $\mathcal{S}_i$ gerade zur Auswertung von Termen in T_i, bzw. Formeln in F_i genügt, und diese Auswertung in den höheren Stufen erhalten bleibt. Der Term $[\bar\beta](\epsilon x.\bar\phi)$ stellt eine Art Normalform dar, die stabil unter Substitutionen ist. Dies ist wichtig für den Nachweis der Substitutivität. Daß diese Konstruktion die Auswahlfunktion α auf die Extension anwendet, bedeutet übrigens nicht, daß hier eine extensionale Struktur erzeugt wird: Gerade für Terme mit $[\bar\beta](\epsilon x.\bar\phi) \in \mathrm{GT}(H)$ ist die Extensionalität nicht notwendigerweise gegeben.

Nun wird $\mathcal{S}$ definiert durch

$$\mathcal{A}(\epsilon x.\phi,\beta) := \mathcal{A}_{\iota(\epsilon x.\phi)}(\epsilon x.\phi,\beta)$$
$$\mathcal{S} := (\mathcal{D},\mathcal{I},\mathcal{A})$$

Offenbar sind $\mathcal{S}$ und alle $\mathcal{S}_i$ Prä-Strukturen. Insbesondere stellt die Definition von $\mathrm{GT}(H)$ sicher, daß $\mathcal{D} \neq \emptyset$.

Mit den gegebenen Definitionen wird nun der Reihe nach gezeigt:

1. $\mathrm{val}(\mathcal{S}_0,\beta,t) = t$ für $t \in \mathrm{GT}(H)$ und β beliebig.
2. Für beliebige $t \in \mathbf{Term}^+$, $\phi \in \mathbf{Form}^+$ und Belegungen β gilt:
   a. $\mathrm{val}(\mathcal{S},\beta,t) = \mathrm{val}(\mathcal{S}_{\iota(t)},\beta,t)$.
   b. $\mathrm{val}(\mathcal{S},\beta,t) = t$, falls $t \in \mathrm{GT}(H)$.

c. $\mathcal{S}, \beta \models \phi$ gdw. $\mathcal{S}_{\iota(\phi)}, \beta \models \phi$.

3. Es gilt

$$\mathcal{A}(\epsilon x.\phi, \beta) = \begin{cases} [\bar\beta](\epsilon x.\bar\phi) & \text{falls } [\bar\beta](\epsilon x.\bar\phi) \in \mathrm{GT}(H), \\ \alpha(\mathrm{Ext}(\mathcal{S}, \bar\beta, \epsilon x.\bar\phi)) & \text{sonst,} \end{cases}$$

wobei $\bar\phi$ die Generalisierung von ϕ bzgl. x ist, $\phi = [v_i/g_i](\bar\phi)$, und $\bar\beta = \beta\{v_i \leftarrow \mathrm{val}(\mathcal{S}, \beta, g_i)\}$.

4. $\mathcal{S}$ ist substitutiv.[13]

5. $\mathcal{S} \models \phi$ für alle $\phi \in H$.

6. $\mathcal{S}$ hat die Auswahleigenschaft.

Zu 1. Induktion über die Struktur von $t \in \mathrm{GT}(H)$.

- $t = f(t_1, \ldots, t_n) \implies t_i \in GT(H) \overset{\text{Ind.}}{\implies} \mathrm{val}(\mathcal{S}_0, \beta, t_i) = t_i$
$\implies \mathrm{val}(\mathcal{S}_0, \beta, t) = \mathcal{I}(f)(t_1, \ldots, t_n) = f(t_1, \ldots, t_n) = t.$

- $t = \epsilon x.\phi \implies \mathrm{val}(\mathcal{S}_0, \beta, t) = \mathcal{A}_0(\epsilon x.\phi, \beta) = \epsilon x.\phi = t.$

t kann keine Variable sein, da $\mathrm{GT}(H)$ nur Grundterme enthält.

Zu 2a. t enthält nur ϵ-Terme $\epsilon x.\phi$ mit $\iota(\epsilon x.\phi) \leq \iota(t)$. Zu zeigen ist, daß $\mathcal{A}$ und $\mathcal{A}_{\iota(t)}$ auf diesen ϵ-Termen gleich sind. Für $\iota(t) = 0$ ist die Aussage trivial. Sonst sind zwei Fälle zu betrachten.

- Falls $\iota(\epsilon x.\phi) = 0$, ist $\epsilon x.\phi \in \mathrm{GT}(H)$, $\implies \mathcal{A}(\epsilon x.\phi, \beta) = \mathcal{A}_0(\epsilon x.\phi, \beta) = \epsilon x.\phi$. Sei $\epsilon x.\phi = [v_i/g_i](\epsilon x.\bar\phi)$, $\bar\phi$ die Generalisierung von ϕ bzgl. x und $\bar\beta$ die zugehörige Belegung $\bar\beta = \beta\{v_i \leftarrow \mathrm{val}(\mathcal{S}_{\iota(g_i)}, \beta, g_i)\}$. Da $\epsilon x.\phi \in \mathrm{GT}(H)$, die g_i Teilterme von $\epsilon x.\phi$ sind, und außerdem Grundterme sein müssen, sind auch die $g_i \in \mathrm{GT}(H)$, also $\iota(g_i) = 0$. Mit 1. folgt $\bar\beta = \beta\{v_i \leftarrow g_i\}$, und damit $\mathcal{A}'_{\iota(t)}(\epsilon x.\bar\phi, \bar\beta) = [\bar\beta](\epsilon x.\bar\phi) = [\beta]([v_i/g_i](\epsilon x.\bar\phi)) = [\beta](\epsilon x.\phi) = \epsilon x.\phi \in \mathrm{GT}(H) \implies \mathcal{A}_{\iota(t)}(\epsilon x.\phi, \beta) = \epsilon x.\phi$.

- Falls $\iota(\epsilon x.\phi) > 0$, ist

$$\begin{aligned}
&\mathcal{A}(\epsilon x.\phi, \beta) \\
&= \mathcal{A}_{\iota(\epsilon x.\phi)}(\epsilon x.\phi, \beta) \\
&= \mathcal{A}'_{\iota(\epsilon x.\phi)}(\epsilon x.\bar\phi, \bar\beta), \qquad \text{da } \epsilon x.\phi \in T_{\iota(\epsilon x.\phi)}. \\
&= \begin{cases} [\bar\beta](\epsilon x.\bar\phi) & \text{falls } [\bar\beta](\epsilon x.\bar\phi) \in \mathrm{GT}(H), \\ \alpha(\mathrm{Ext}(\mathcal{S}_{\iota(\bar\phi)}, \bar\beta, \epsilon x.\bar\phi)) & \text{sonst,} \end{cases} \\
&\quad \text{wobei } \bar\phi \text{ die Generaliesierung von } \phi \text{ bzgl. } x \text{ ist, } \phi = [v_i/g_i](\bar\phi), \\
&\quad \text{und } \bar\beta = \beta\{v_i \leftarrow \mathrm{val}(\mathcal{S}_{\iota(g_i)}, \beta, g_i)\}. \\
&= \mathcal{A}'_{\iota(t)}(\epsilon x.\bar\phi, \bar\beta), \qquad \text{da } \epsilon x.\phi \in T_{\iota(t)}. \\
&= \mathcal{A}_{\iota(t)}(\epsilon x.\phi, \beta)
\end{aligned}$$

Wesentlich ist hier, daß $\bar\beta$ und $\bar\phi$ in den Definitionen für $\mathcal{A}_{\iota(t)}$ und $\mathcal{A}_{\iota(\epsilon x.\phi)}$ gleich sind.

[13] In Definition 5.9 wird die Substitutivität für Strukturen definiert, während wir an dieser Stelle nur wissen, daß $\mathcal{S}$ Prä-Struktur ist. Die Definition läßt sich allerdings problemlos auf Prä-Strukturen anwenden, und dies ist hier gemeint.

Zu 2b. Für $t \in \mathrm{GT}(H)$ ist $\mathrm{val}(\mathcal{S},\beta,t) \overset{2a.}{=} \mathrm{val}(\mathcal{S}_0,\beta,t) \overset{1.}{=} t$.

Zu 2c. ϕ enthält nur Terme $t \in T_{\iota(\phi)}$. Die Aussage folgt somit aus 2a.

Zu 3. Da in der Definition von $\mathcal{A}$ und den Induktionen (†) und (‡) die Interpretationen jeweils mit ι ausgewählt werden, ist hier keine Induktion, sondern nur eine Fallunterscheidung notwendig.

- $\iota(\epsilon x.\phi) = 0 \implies \mathcal{A}(\epsilon x.\phi,\beta) = \mathcal{A}_0(\epsilon x.\phi,\beta) = \epsilon x.\phi$. Da $\epsilon x.\phi \in \mathrm{GT}(H)$, sind alle $g_i \in \mathrm{GT}(H) \overset{2b.}{\implies} \mathrm{val}(\mathcal{S},\beta,g_i) = g_i \implies \bar{\beta} = \beta\{v_i \leftarrow g_i\} \implies [\bar{\beta}](\epsilon x.\bar{\phi}) = [\beta]([v_i/g_i](\epsilon x.\bar{\phi})) = [\beta](\epsilon x.\phi) = \epsilon x.\phi \in \mathrm{GT}(H)$.[14]
- $\iota(\epsilon x.\phi) > 0$. Es ist

$$\begin{aligned} &\mathcal{A}(\epsilon x.\phi,\beta)\\ &= \mathcal{A}'_{\iota(\epsilon x.\phi)}(\epsilon x.\bar{\phi},\bar{\beta}_1)\\ &= \begin{cases} [\bar{\beta}_1](\epsilon x.\bar{\phi}) & \text{falls } [\bar{\beta}_1](\epsilon x.\bar{\phi}) \in \mathrm{GT}(H),\\ \alpha(\mathrm{Ext}(\mathcal{S}_{\iota(\bar{\phi})},\bar{\beta}_1,\epsilon x.\bar{\phi})) & \text{sonst,} \end{cases} \end{aligned}$$

  wobei $\bar{\beta}_1 = \beta\{v_i \leftarrow \mathrm{val}(\mathcal{S}_{\iota(g_i)},\beta,g_i)\}$. Es gilt $\bar{\beta}_1 = \bar{\beta}$, wegen 2a., und außerdem, wegen 2.c, $\mathrm{Ext}(\mathcal{S}_{\iota(\bar{\phi})},\bar{\beta},\epsilon x.\bar{\phi}) = \mathrm{Ext}(\mathcal{S},\bar{\beta},\epsilon x.\bar{\phi})$, womit die Aussage gezeigt ist.

Zu 4. Man nimmt eine Induktionsverstärkung vor, und zeigt das Substitutionslemma für beliebige Terme und Formeln. Interessant ist nur der Fall für ϵ-Terme. Zu zeigen ist also

$$\mathcal{A}([y/t](\epsilon x.\phi),\beta) = \mathcal{A}(\epsilon x.\phi,\beta')$$

mit $\beta' := \beta\{y \leftarrow \mathrm{val}(\mathcal{S},\beta,t)\}$, wobei $\mathrm{fv}(t) \cap \mathrm{bv}(\epsilon x.\phi) = \emptyset$. Das Substitutionslemma für Teilterme von ϕ kann als Induktionsvoraussetzung verwendet werden. Ferner sei vorausgesetzt, daß $y \in \mathrm{fv}(\epsilon x.\phi)$ und $y \neq x$, da die Aussage sonst trivial ist. Es sind zwei Fälle zu betrachten:

- $\iota(\epsilon x.\phi) = 0 \implies \epsilon x.\phi \in \mathbf{Term}^0 \implies [y/t](\epsilon x.\phi) = \epsilon x.\phi$, woraus die Aussage mit 2b. folgt.
- $\iota(\epsilon x.\phi) > 0$. Für die Anwendung von 3. auf die linke Seite wird benutzt, daß die Generalisierungen von ϕ und $[y/t]\phi$ gleich sind. Weiter ist $[y/t](\phi) = [y/t]([v_i/g_i]\bar{\phi}) = [y/t, v_i/[y/t](g_i)](\bar{\phi})$. Also

$$\mathcal{A}([y/t](\epsilon x.\phi),\beta) \overset{3.}{=} \begin{cases} [\bar{\beta}_1](\epsilon x.\bar{\phi}) & \text{falls } [\bar{\beta}_1](\epsilon x.\bar{\phi}) \in \mathrm{GT}(H),\\ \alpha(\mathrm{Ext}(\mathcal{S},\bar{\beta}_1,\epsilon x.\bar{\phi})) & \text{sonst,} \end{cases}$$

  mit $\bar{\beta}_1 = \beta\{y \leftarrow \mathrm{val}(\mathcal{S},\beta,t), v_i \leftarrow \mathrm{val}(\mathcal{S},\beta,[y/t](g_i))\}$. Man kann $\bar{\beta}_1$ wie folgt vereinfachen:

$$\begin{aligned} \bar{\beta}_1 &= \beta\{y \leftarrow \mathrm{val}(\mathcal{S},\beta,t), v_i \leftarrow \mathrm{val}(\mathcal{S},\beta,[y/t](g_i))\}\\ &= \beta'\{v_i \leftarrow \mathrm{val}(\mathcal{S},\beta,[y/t](g_i))\}\\ &\overset{\text{Ind.}}{=} \beta'\{v_i \leftarrow \mathrm{val}(\mathcal{S},\beta',g_i)\} \end{aligned}$$

[14]Die Argumentation ist hier fast dieselbe wie für 2a., allerdings werden die g_i jetzt in $\mathcal{S}$ statt $\mathcal{S}_{\iota(g_i)}$ ausgewertet. Dies ist möglich wegen 2b.

da die g_i Teilterme von $\epsilon x.\phi$ sind. Für die rechte Seite ist

$$\mathcal{A}(\epsilon x.\phi, \beta') \overset{3.}{=} \begin{cases} [\bar{\beta}_2](\epsilon x.\bar{\phi}) & \text{falls } [\bar{\beta}_2](\epsilon x.\bar{\phi}) \in \mathrm{GT}(H), \\ \alpha(\mathrm{Ext}(\mathcal{S}, \bar{\beta}_2, \epsilon x.\bar{\phi})) & \text{sonst,} \end{cases}$$

mit $\bar{\beta}_2 = \beta'\{v_i \leftarrow \mathrm{val}(\mathcal{S}, \beta', g_i)\} = \bar{\beta}_1$. Damit ist die Identität gezeigt.

Zu 5. Der Beweis erfolgt durch Induktion über die Formelkomplexität von $\phi \in H$.

- $\phi = p(t_1, \ldots, t_n) \implies$ alle $t_i \in \mathrm{GT}(H) \overset{2b.}{\implies}$ alle $t_i = \mathrm{val}(\mathcal{S}, \beta, t_i) \implies \mathrm{val}(\mathcal{S}, \beta, \phi) = \mathcal{I}(p)(t_1, \ldots, t_n) = tt \implies \mathcal{S}, \beta \models \phi$.

- $\phi = \neg p(t_1, \ldots, t_n) \overset{(H0)}{\implies} p(t_1, \ldots, t_n) \notin H \implies \mathrm{val}(\mathcal{S}, \beta, p(t_1, \ldots, t_n)) = ff$ also analog zum ersten Fall $\mathcal{S}, \beta \models \phi$.

- $\phi = \neg\neg\phi_1 \overset{(H6)}{\implies} \phi_1 \in H \overset{Ind.}{\implies} \mathcal{S}, \beta \models \phi_1 \implies \mathcal{S}, \beta \models \phi$.

- $\phi = \phi_1 \wedge \phi_2 \overset{(H1)}{\implies} \phi_1, \phi_2 \in H \overset{Ind.}{\implies} \mathcal{S}, \beta \models \phi_1$ und $\mathcal{S}, \beta \models \phi_2 \implies \mathcal{S}, \beta \models \phi$.

- $\phi = \neg(\phi_1 \wedge \phi_2) \overset{(H2)}{\implies} \neg\phi_1 \in H$ oder $\neg\phi_2 \in H. \overset{Ind.}{\implies} \mathcal{S}, \beta \not\models \phi_1$ oder $\mathcal{S}, \beta \not\models \phi_2 \implies \mathcal{S}, \beta \not\models \phi_1 \wedge \phi_2 \implies \mathcal{S}, \beta \models \phi$.

- $\phi = \exists x.\phi_1 \overset{(H3)}{\implies} [x/\epsilon x.\phi_1](\phi_1) \in H \overset{Ind.}{\implies} \mathcal{S}, \beta \models [x/\epsilon x.\phi_1](\phi_1) \overset{4.}{\implies} \mathcal{S}, \beta\{x \leftarrow \mathrm{val}(\mathcal{S}, \beta, \epsilon x.\phi_1)\} \models \phi_1 \implies \mathcal{S}, \beta \models \phi$.

- $\phi = \neg\exists x.\phi_1 \overset{(H4)}{\implies} [x/t](\neg\phi_1) \in H$ für alle $t \in \mathrm{GT}(H) \overset{Ind.}{\implies} \mathcal{S}, \beta \models [x/t]\neg\phi_1$ für alle $t \in \mathrm{GT}(H) \overset{4.,2b.}{\implies} \mathcal{S}, \beta\{x \leftarrow t\} \not\models \phi_1$ für alle $t \in \mathrm{GT}(H) = \mathcal{D} \implies \mathcal{S}, \beta \not\models \exists x.\phi_1$. $\implies \mathcal{S}, \beta \models \phi$.

Zu 6. Zu zeigen ist noch, daß $\mathcal{S}$ die Auswahleigenschaft hat. Sei $\mathcal{S}, \beta \models \exists x.\phi$. Laut 3. gilt

$$\mathcal{A}(\epsilon x.\phi, \beta) = \begin{cases} [\bar{\beta}](\epsilon x.\bar{\phi}) & \text{falls } [\bar{\beta}](\epsilon x.\bar{\phi}) \in \mathrm{GT}(H), \\ \alpha(\mathrm{Ext}(\mathcal{S}, \bar{\beta}, \epsilon x.\bar{\phi})) & \text{sonst,} \end{cases}$$

mit den üblichen Definitionen für $\bar{\phi}$ und $\bar{\beta}$. Gemäß den Zweigen der Fallunterscheidung können zwei Fälle auftreten:

- Fall 1: $[\bar{\beta}](\epsilon x.\bar{\phi}) \in \mathrm{GT}(H)$. Es ist $[\bar{\beta}](\epsilon x.\bar{\phi}) = \epsilon x.[\bar{\beta}]_x^x(\bar{\phi})$. Laut (H5) gilt entweder $\neg\exists x.[\bar{\beta}]_x^x(\bar{\phi}) \in H$ oder $[x/\epsilon x.[\bar{\beta}]_x^x(\bar{\phi})]([\bar{\beta}]_x^x(\bar{\phi})) \in H$.
    - Fall 1a: $\neg\exists x.[\bar{\beta}]_x^x(\bar{\phi}) \in H \overset{5.}{\implies} \mathcal{S} \models \neg\exists x.[\bar{\beta}]_x^x(\bar{\phi}) \implies \mathcal{S} \models [\bar{\beta}]\neg\exists x.\bar{\phi} \overset{4.}{\implies} \mathcal{S}, \beta\{v_i \leftarrow \mathrm{val}(\mathcal{S}, \beta, g_i)\} \models \neg\exists x.\bar{\phi} \overset{4.}{\implies} \mathcal{S}, \beta \models [v_i/g_i]\neg\exists x.\bar{\phi}$ $\implies \mathcal{S}, \beta \models \neg\exists x.\phi \implies \mathcal{S}, \beta \not\models \exists x.\phi$ im Widerspruch zur Voraussetzung $\mathcal{S}, \beta \models \exists x.\phi$. Dieser Fall kann also nicht eintreten.
    - Fall 1b: $[x/\epsilon x.[\bar{\beta}]_x^x(\bar{\phi})]([\bar{\beta}]_x^x(\bar{\phi})) \in H$. Es ist $[x/\epsilon x.[\bar{\beta}]_x^x(\bar{\phi})] \circ [\bar{\beta}]_x^x = [\bar{\beta}] \circ [x/\epsilon x.\bar{\phi}]$ $\overset{5.}{\implies} \mathcal{S} \models [\bar{\beta}]([x/\epsilon x.\bar{\phi}](\bar{\phi})) \overset{4.}{\implies} \mathcal{S}, \beta\{v_i \leftarrow \mathrm{val}(\mathcal{S}, \beta, g_i)\} \models [x/\epsilon x.\bar{\phi}](\bar{\phi}) \overset{4.}{\implies} \mathcal{S}, \beta \models [v_i/g_i]([x/\epsilon x.\bar{\phi}](\bar{\phi}))$. Durch Hineinziehen der Substitution $[v_i/g_i]$ folgt $\mathcal{S}, \beta \models [x/[v_i/g_i](\epsilon x.\bar{\phi})]([v_i/g_i](\bar{\phi})) \implies \mathcal{S}, \beta \models [x/\epsilon x.\phi](\phi) \overset{4.}{\implies} \mathcal{S}, \beta\{x \leftarrow \mathcal{A}(\epsilon x.\phi, \beta)\} \models \phi$.

- Fall 2: $[\bar{\beta}](\epsilon x.\bar{\phi}) \notin \mathrm{GT}(H)$. Es gilt

$$\begin{aligned}
&\mathcal{S}, \beta \models \exists x.\phi \\
\Longrightarrow\ &\mathcal{S}, \beta\{x \leftarrow d\} \models \phi \quad \text{für ein } d \in \mathrm{GT}(H) \\
\Longrightarrow\ &\mathcal{S}, \beta\{x \leftarrow d\} \models [v_i/g_i](\bar{\phi}) \\
\overset{4.}{\Longrightarrow}\ &\mathcal{S}, \beta\{x \leftarrow d, v_i \leftarrow \mathrm{val}(\mathcal{S}, \beta\{x \leftarrow d\}, g_i)\} \models \bar{\phi} \\
\Longrightarrow\ &\mathcal{S}, \bar{\beta}\{x \leftarrow d\} \models \bar{\phi}, \quad \text{da } x \notin \mathrm{fv}(g_i) \\
\Longrightarrow\ &\mathrm{Ext}(\mathcal{S}, \bar{\beta}, \epsilon x.\bar{\phi}) \neq \emptyset \\
\Longrightarrow\ &\alpha(\mathrm{Ext}(\mathcal{S}, \bar{\beta}, \epsilon x.\bar{\phi})) \in \mathrm{Ext}(\mathcal{S}, \bar{\beta}, \epsilon x.\bar{\phi}) \\
\Longrightarrow\ &\mathcal{S}, \bar{\beta}\{x \leftarrow \alpha(\mathrm{Ext}(\mathcal{S}, \bar{\beta}, \epsilon x.\bar{\phi}))\} \models \bar{\phi} \\
\Longrightarrow\ &\mathcal{S}, \bar{\beta}\{x \leftarrow \mathcal{A}(\epsilon x.\phi, \beta)\} \models \bar{\phi} \\
\overset{4.}{\Longrightarrow}\ &\mathcal{S}, \beta\{x \leftarrow \mathcal{A}(\epsilon x.\phi, \beta)\} \models \phi
\end{aligned}$$

Damit ist die Vollständigkeit gezeigt!

5.6.3. Der Kalkül $\mathcal{G}_E$

Während die substitutive Semantik den Vorteil hat, daß für sie ein vollständiger Kalkül existiert, mit dem ein automatisches Beweissystem arbeiten kann, ist die extensionale Semantik der Intuition leichter zugänglich. Der Kalkül $\mathcal{G}_S$ soll daher zu einem Kalkül $\mathcal{G}_E$ erweitert werden, welcher auch für die extensionale Semantik vollständig ist. Dabei bedeutet „erweitert“, daß jeder (geschlossene) $\mathcal{G}_S$-Beweis auch ein (geschlossener) $\mathcal{G}_E$-Beweis ist. *Es ist dann möglich, dem Benutzer den Kalkül $\mathcal{G}_E$ zur Verfügung zu stellen, während der automatische Beweiser nur $\mathcal{G}_S$-Beweise, bzw. -Teilbeweise erzeugt.* Der Kalkül $\mathcal{G}_E$ ist also primär auf die Bedürfnisse eines menschlichen Beweisers zugeschnitten, während sich $\mathcal{G}_S$ an den Anforderungen des Automaten orientiert.[15]

Die Einschränkungen $t \in \mathrm{GT}(\Gamma \vdash \Delta)$, bzw. $\epsilon x.\phi \in \mathrm{GT}(\Gamma \vdash \Delta)$ der Regeln $\exists r$ und *ϵ-cut* können für den Kalkül $\mathcal{G}_E$ weggelassen werden. Es soll dem Benutzer erlaubt sein, diese Regeln auf beliebige Terme anzuwenden, wenn ihm dies sinnvoll erscheint. Hinzukommen muß eine Regel, die es ermöglicht, die Extensionalität auszunutzen. Sie muß es ermöglichen, aus der Gleichheit der Extensionen zweier Terme die Gleichheit der zugehörigen ϵ-Terme zu schließen. Dazu bietet sich die folgende Regel an:

$$\frac{\Gamma \vdash \forall z.\Big([x/z](\phi) \leftrightarrow [y/z](\psi)\Big), \Delta \quad \epsilon x.\phi \doteq \epsilon y.\psi, \Gamma \vdash \Delta}{\Gamma \vdash \Delta}$$

wobei $\epsilon x.\phi$, $\epsilon y.\psi \in \mathbf{Term}^0$ Grundterme sind, und $z \in \mathbf{V}$ eine Variable ist, die weder in ϕ, noch in ψ gebunden vorkommt. In dieser Regel kommen ein Allquantor und eine Äquivalenz vor, die wir als Abkürzungen auffassen. Außerdem sind auf den linken Ast sofort noch einige Regeln anwendbar. Unter Benutzung der Definition $\forall x.\phi := \neg\exists x.\neg\phi$

[15] Natürlich müssen beide Kalküle noch mit Metavariablen versehen werden, um automatisches Beweisen zu ermöglichen.

erhält man folgende Ableitung:

$$\dfrac{\dfrac{\dfrac{\dfrac{\Gamma \vdash [x/\mathbf{c}](\phi) \leftrightarrow [y/\mathbf{c}](\psi), \Delta}{\neg\Big([x/\mathbf{c}](\phi) \leftrightarrow [y/\mathbf{c}](\psi)\Big), \Gamma \vdash \Delta}\,\neg l}{\exists z.\neg\Big([x/z](\phi) \leftrightarrow [y/z](\psi)\Big), \Gamma \vdash \Delta}\,\exists l}{\Gamma \vdash \forall z.\Big([x/z](\phi) \leftrightarrow [y/z](\psi)\Big), \Delta}\,\neg r \qquad \epsilon x.\phi \doteq \epsilon y.\psi, \Gamma \vdash \Delta}{\Gamma \vdash \Delta}$$

wobei $\mathbf{c} := \epsilon z.\neg\Big([x/z](\phi) \leftrightarrow [y/z](\psi)\Big)$. Diese Ableitung führt zu folgender Regel:

$$\dfrac{\Gamma \vdash [x/\mathbf{c}](\phi) \leftrightarrow [y/\mathbf{c}](\psi), \Delta \quad \epsilon x.\phi \doteq \epsilon y.\psi, \Gamma \vdash \Delta}{\Gamma \vdash \Delta}\,ext$$

Die Abkürzung[16] $\leftrightarrow$ wird hier nicht weiter aufgelöst, da sonst eine unübersichtliche Regel mit drei Prämissen entstehen würde. Eine Implementierung wird natürlich die Äquivalenz als normalen Junktor zur Verfügung stellen. Dann bereitet die Regel in der dargestellten Form keine Probleme. Für unsere Zwecke genügt die Feststellung, daß für alle Prä-Strukturen $\mathcal{S}$ und Belegungen β gilt:

$$\mathrm{val}(\mathcal{S}, \beta, \phi \leftrightarrow \psi) = \begin{cases} tt & \text{falls } \mathrm{val}(\mathcal{S}, \beta, \phi) = \mathrm{val}(\mathcal{S}, \beta, \psi), \\ ff & \text{sonst.} \end{cases} \qquad (*)$$

An dieser Stelle ist bereits ersichtlich, warum die *ext* Regel für ein automatisches Beweissystem nicht geeignet ist: Selbst wenn man die Regel so einschränken kann, daß sie nur auf ϵ-Terme angewandt wird, die auch in der Sequenz vorkommen – es wurde nicht versucht, zu zeigen, daß man damit noch vollständig ist – so müßte die Regel doch für jedes *Paar* von ϵ-Termen angewandt werden. Die Anzahl von Kombinationen ist bekanntlich quadratisch in der Zahl vorkommender ϵ-Terme. Dies wäre noch zu vertreten, aber jede Regelanwendung erzeugt einen neuen ϵ-Term, nämlich $\mathbf{c}$, auf welchen die *ext*-Regel wiederum angewandt werden müßte.

Möglicherweise lassen sich Einschränkungen, oder sogar eine andere Regel angeben, mit denen auch die extensionale Semantik durch einen automatischen Beweiser vollständig behandelt werden kann. Da aber nicht klar ist, wie solche Einschränkungen oder Regeln aussehen könnten, und der zugehörige Vollständigkeitsbeweis sicherlich mindestens so kompliziert wie der des letzten Abschnitts sein würde, wurde an dieser Stelle nicht weiter geforscht.

Um den Kalkül $\mathcal{G}_E$ zu komplettieren, fehlen nun noch Regeln für die Gleichheit. Hier können die üblichen Einsetzungsregeln eines klassischen Grundkalküls übernommen werden. Diese werden normalerweise wie folgt notiert:

$$\dfrac{s \doteq t, \phi(t), \phi(s), \Gamma \vdash \Delta}{s \doteq t, \phi(s), \Gamma \vdash \Delta}\,\overset{\rightarrow}{=} l$$

und entsprechend $\overset{\leftarrow}{=} l$, falls die Gleichung andersherum angewandt wird, und $\overset{\rightarrow}{=} r$ und $\overset{\leftarrow}{=} r$, falls ϕ im Sukzedenten der Sequenz steht. Dabei wird mit $\phi(s)$ ein bestimmtes Vorkommen des Terms s in ϕ bezeichnet. Um an dieser Stelle keine Positionen einführen

[16] $\phi \leftrightarrow \psi := \neg(\phi \wedge \neg\psi) \wedge \neg(\neg\phi \wedge \psi)$

Abbildung 5.3.: Beweisregeln des Kalküls $\mathcal{G}_E$

$$\frac{\phi, \psi, \Gamma \vdash \Delta}{\phi \wedge \psi, \Gamma \vdash \Delta} \wedge l \qquad \frac{\Gamma \vdash \phi, \Delta \quad \Gamma \vdash \psi, \Delta}{\Gamma \vdash \phi \wedge \psi, \Delta} \wedge r$$

$$\frac{\Gamma \vdash \phi, \Delta}{\neg\phi, \Gamma \vdash \Delta} \neg l \qquad \frac{\phi, \Gamma \vdash \Delta}{\Gamma \vdash \neg\phi, \Delta} \neg r$$

$$\frac{[x/\epsilon x.\phi](\phi), \Gamma \vdash \Delta}{\exists x.\phi, \Gamma \vdash \Delta} \exists l \qquad \frac{\Gamma \vdash [x/t](\phi), \exists x.\phi, \Delta}{\Gamma \vdash \exists x.\phi, \Delta} \exists r$$

wobei $t \in \mathbf{Term}^0$.

$$\frac{s \doteq t, [y/t]\phi, [y/s]\phi, \Gamma \vdash \Delta}{s \doteq t, [y/s]\phi, \Gamma \vdash \Delta} \stackrel{\rightarrow}{=} l \qquad \frac{s \doteq t, [y/s]\phi, [y/t]\phi, \Gamma \vdash \Delta}{s \doteq t, [y/t]\phi, \Gamma \vdash \Delta} \stackrel{\leftarrow}{=} l$$

$$\frac{s \doteq t, \Gamma \vdash [y/t]\phi, [y/s]\phi, \Delta}{s \doteq t, \Gamma \vdash [y/s]\phi, \Delta} \stackrel{\rightarrow}{=} r \qquad \frac{s \doteq t, \Gamma \vdash [y/s]\phi, [y/t]\phi, \Delta}{s \doteq t, \Gamma \vdash [y/t]\phi, \Delta} \stackrel{\leftarrow}{=} r$$

$$\frac{\Gamma \vdash \exists x.\phi, \Delta \quad [x/\epsilon x.\phi](\phi), \Gamma \vdash \Delta}{\Gamma \vdash \Delta} \epsilon\text{-}cut$$

wobei $\epsilon x.\phi \in \mathbf{Term}^0$.

$$\frac{\Gamma \vdash [x/\mathbf{c}](\phi) \leftrightarrow [y/\mathbf{c}](\psi), \Delta \quad \epsilon x.\phi \doteq \epsilon y.\psi, \Gamma \vdash \Delta}{\Gamma \vdash \Delta} ext$$

wobei $\epsilon x.\phi$, $\epsilon y.\psi \in \mathbf{Term}^0$, $\mathbf{c} = \epsilon z.\neg\Big([x/z](\phi) \leftrightarrow [y/z](\psi)\Big)$, $z \in \mathbf{V} \setminus (\mathrm{bv}(\phi) \cup \mathrm{bv}(\psi))$.

zu müssen, und trotzdem technisch präzise zu sein, schreiben wir $\phi(s)$ als $[y/s]\phi'$, und gehen damit von einer Formel ϕ' aus, in der die kritische Position durch die Variable y gekennzeichnet ist. So kommt man zu der Regel

$$\frac{s \doteq t, [y/t]\phi, [y/s]\phi, \Gamma \vdash \Delta}{s \doteq t, [y/s]\phi, \Gamma \vdash \Delta} \stackrel{\rightarrow}{=} l$$

Diese Regel ist korrekt, da s und t Grundterme sind. Es kann also zu keinen Konflikten mit in ϕ gebundenen Variablen kommen. Auch die Möglichkeit, daß y in ϕ nicht, oder mehrmals frei vorkommt, beeinträchtigt nicht die Korrektheit. Bei der Beweisführung hat diese Notation den Vorteil, daß man mit der Substitutivität argumentieren kann. Sämtliche Regeln des Kalküls $\mathcal{G}_E$ sind in Abbildung 5.3 zusammengefaßt. Mit diesen Regeln werden $\mathcal{G}_E$-Beweise definiert, wobei für die Gleichheit neue Axiome hinzukommen.

5.21 Definition ($\mathcal{G}_E$-Beweis)
Ein $\mathcal{G}_E$-**Axiom** *ist eine Sequenz* $\Gamma \vdash \Delta$ *mit* $p(t_1, \ldots, t_n) \in \Gamma \cap \Delta$ *für eine atomare Formel* $p(t_1, \ldots, t_n)$, *oder* $s \doteq s \in \Delta$ *für einen Term* $s \in \mathbf{Term}^0$. $\mathcal{G}_E$-**Beweise** *und* **geschlossene** $\mathcal{G}_E$*-Beweise sind wie für* $\mathcal{G}_S$ *definiert, siehe Def. 5.17.*

5.22 Satz (Korrektheit von $\mathcal{G}_E$)
Falls für eine Formel $\phi \in \mathbf{Form}^0$ *ein geschlossener* $\mathcal{G}_E$*-Beweis existiert, ist* ϕ *E-allgemeingültig.*

Wie für den Kalkül $\mathcal{G}_S$ wird hier auf den Nachweis der Korrektheit verzichtet, und nur die Vollständigkeit bewiesen.

5.23 Satz (Vollständigkeit von $\mathcal{G}_E$)
Sei $\phi \in \mathbf{Form}^0$ eine geschlossene, E-allgemeingültige Formel. Dann existiert ein geschlossener $\mathcal{G}_E$-Beweis für ϕ.

Beweis. Auch dieser Beweis benutzt Hintikkamengen. Da nun die Gleichheit hinzukommt, ist jedoch wie im klassischen Fall eine Äquivalenzklassenbildung nötig. Zunächst wird eine Menge H genauso konstruiert wie auf Seite 49. Deren Eigenschaften sind jedoch wegen des anderen Regelsatzes nicht dieselben wie dort. Die Eigenschaften (H0), (H4) und (H5) werden ersetzt durch:

(H0) Für keine atomare Formel $\phi = p(t_1, \dots, t_n)$ oder $\phi = s \doteq t$ ist $\phi \in H$ und $\neg\phi \in H$, da sonst der Ast abgeschlossen wäre.

(H4) Für jede Formel $\phi = \neg\exists x.\phi \in H$ und jeden Term $t \in \mathbf{Term}^0$ ist $[x/t](\neg\phi) \in H$ wegen der Regeln $\exists r$, $\neg l$, $\neg r$.

(H5) Für jeden ϵ-Term $\epsilon x.\phi \in \mathbf{Term}^0$ ist $[x/\epsilon x.\phi](\phi) \in H$ oder $\neg\exists x.\phi \in H$ wegen der Regel *ϵ-cut*.

Hinzu kommen die Eigenschaften:

(H7) Für alle ϵ-Terme $\epsilon x.\phi$, $\epsilon y.\psi \in \mathbf{Term}^0$ ist $\neg\Big([x/\mathbf{c}](\phi) \leftrightarrow [y/\mathbf{c}](\psi)\Big) \in H$, oder $\epsilon x.\phi \doteq \epsilon y.\psi \in H$, wegen der Regel *ext*.

(H8) Für jede Gleichung $s \doteq t \in H$ oder $t \doteq s \in H$, und jede *atomare*[17] Formel der Form $[y/s](\phi) \in H$ ist $[y/t](\phi) \in H$, wegen der vier $\doteq$-Regeln.

Nun wird aus H durch folgenden Abschluß eine Menge $\bar{H}$ konstruiert:

- $H \subseteq \bar{H}$.
- $s \doteq s \in \bar{H}$ für alle $s \in \mathbf{Term}^0$.
- Für jede Gleichung $s \doteq t \in \bar{H}$ oder $t \doteq s \in \bar{H}$, und jede Gleichung der Form $[y/s](l \doteq r) \in \bar{H} \setminus H$ ist $[y/t](l \doteq r) \in \bar{H}$.

Offenbar hat auch $\bar{H}$ die Eigenschaften (H0) bis (H8), wenn dort überall H durch $\bar{H}$ ersetzt wird. Lediglich (H8) erfordert etwas Überlegung: es kommen ja neue Gleichungen hinzu! Allerdings läßt sich jede Anwendung einer Gleichung $s \doteq t \in \bar{H} \setminus H$ auf eine atomare Formel aus H simulieren durch Anwendung von Gleichungen in H, und zwar durch Anwendung derjenigen Gleichungen, durch die $s \doteq t$ aus einer Gleichung $s_0 \doteq s_0 \in \bar{H}$ entstanden ist. Zusätzlich zu (H0) bis (H8) hat $\bar{H}$ noch die folgende Eigenschaft:

(H10) $s \doteq s \in \bar{H}$ für alle $s \in \mathbf{Term}^0$.

[17] Atomare Formeln sind hier Formeln der Form $p(t_1, \dots, t_n)$ oder $s \doteq t$.

Wie im Beweis für die klassische Prädikatenlogik mit Gleichheit wird nun anhand der Gleichungen in $\bar{H}$ eine Kongruenzrelation auf $\mathbf{Term}^0$ definiert durch

$$s \sim t \quad \text{gdw.} \quad s \doteq t \in \bar{H}.$$

Die Relation $\sim$ ist reflexiv wegen (H10), transitiv wegen (H8) und symmetrisch, da für $s \doteq t \in \bar{H}$ wegen (H10) $t \doteq t \in \bar{H}$, und somit laut (H8) auch $t \doteq s \in \bar{H}$. Weiter gilt wegen (H8) und (H10):

$$\text{Falls } t \sim t', \text{ dann } [y/t]s \sim [y/t']s \tag{i}$$

für beliebige Grundterme t, t' und s. Daraus folgt mit $t_i \sim t'_i$:

$$f(t_1, \ldots, t_n) \sim f(t'_1, \ldots, t'_n) \qquad \text{(ii)},$$

und schließlich

$$p(t_1, \ldots, t_n) \in \bar{H} \quad \text{gdw.} \quad p(t'_1, \ldots, t'_n) \in \bar{H}. \tag{iii}$$

Die Äquivalenzklasse eines Terms t bezüglich dieser Äquivalenzrelation soll mit $\bar{t}$ bezeichnet werden. Ferner soll $v(\bar{t}) \in \bar{t}$ für jede Klasse einen festen Vertreter auswählen.

Mit diesen Festlegungen kann nun eine Prä-Struktur $\mathcal{S} = (\mathcal{D}, \mathcal{I}, \mathcal{A})$ definiert werden, wobei

$$\begin{aligned}
\mathcal{D} &:= \mathbf{Term}^0/\!\sim\; = \{\bar{t} \mid t \in \mathbf{Term}^0\} \\
\mathcal{I}(f)(\overline{t_1}, \ldots, \overline{t_n}) &:= \overline{f(t_1, \ldots, t_n)} \\
\mathcal{I}(p)(\overline{t_1}, \ldots, \overline{t_n}) &:= \begin{cases} tt & \text{falls } p(t_1, \ldots, t_n) \in \bar{H}, \\ f\!f & \text{sonst.} \end{cases} \\
\mathcal{A}(\epsilon x.\phi, \beta) &:= \overline{[\beta]\epsilon x.\phi}
\end{aligned}$$

wobei $[\beta]$ die Substitution mit $[\beta](x) = v(\beta(x))$ bezeichnet. Dabei ist $\mathcal{I}(f)$ wohldefiniert wegen (ii), $\mathcal{I}(p)$ wegen (iii) und $\mathcal{A}$ wegen der festen Wahl der Vertreter für $[\beta]$. Es wird nun in dieser Reihenfolge gezeigt:

1. $\mathrm{val}(\mathcal{S}, \beta, t) = \overline{[\beta](t)}$ für alle Terme $t \in \mathbf{Term}^0$.
2. $\mathcal{S}$ ist substitutiv.[18]
3. $\mathcal{S}, \beta \models \phi$ für alle $\phi \in \bar{H}$ und alle Belegungen β.
4. $\mathcal{S}$ hat die Auswahleigenschaft.
5. $\mathcal{S}$ ist extensional.

Wie im Beweis für $\mathcal{G}_S$ ist dann ein Widerspruch zur Annahme, ϕ sei E-allgemeingültig, hergestellt, da $\neg\phi \in H \subseteq \bar{H}$, also wegen 3.: $\mathcal{S} \not\models \phi$.

Zu 1. Induktion über die Struktur von t ergibt:

- Falls $t = x \in \mathbf{V}$, ist $\mathrm{val}(\mathcal{S}, \beta, t) = \beta(x) = \overline{v(\beta(x))} = \overline{[\beta](x)}$.

[18]Siehe Fußnote auf Seite 52.

- Falls $t = f(t_1, \ldots, t_n)$, ist

$$\begin{aligned} \mathrm{val}(\mathcal{S}, \beta, t) &= \mathcal{I}(f)(\mathrm{val}(\mathcal{S}, \beta, t_1), \ldots, \mathrm{val}(\mathcal{S}, \beta, t_n)) \\ &\overset{\text{Ind.}}{=} \mathcal{I}(f)(\overline{[\beta]t_1}, \ldots, \overline{[\beta]t_n}) \\ &= \overline{f([\beta]t_1, \ldots, [\beta]t_n)} \\ &= \overline{[\beta](f(t_1, \ldots, t_n))}. \end{aligned}$$

- Falls $t = \epsilon x.\phi$, folgt die Aussage direkt aus der Definition von $\mathcal{A}$.

Zu 2. Seien $\beta' := \beta\{y \leftarrow \mathrm{val}(\mathcal{S}, \beta, t)\}$ und $t' := v(\mathrm{val}(\mathcal{S}, \beta, t))$. Wegen 1. ist $t' \sim [\beta](t)$. Ferner ist

$$[\beta'] = [\beta]_y^{v(\mathrm{val}(\mathcal{S}, \beta, t))} = [\beta]_y^{t'}.$$

Damit gilt:

$$\begin{aligned} &\mathcal{A}([y/t](\epsilon x.\phi), \beta) \\ =\ &\overline{[\beta]([y/t](\epsilon x.\phi))} \\ =\ &\overline{[y/[\beta](t)]([\beta]_y^y(\epsilon x.\phi))} \\ \overset{\text{(i)}}{=}\ &\overline{[y/t']([\beta]_y^y(\epsilon x.\phi))} \\ =\ &\overline{[\beta]_y^{t'}(\epsilon x.\phi)} \\ =\ &\mathcal{A}(\epsilon x.\phi, \beta'). \end{aligned}$$

Zu 3. Der Beweis verläuft fast wörtlich wie für die Aussage 5. im Vollständigkeitsbeweis für $\mathcal{G}_S$, also durch Induktion über die Formelkomplexität. Es werden also nur diejenigen Fälle aufgeführt, die sich hier unterscheiden, bzw. durch die Gleichheit dazukommen.

- $\phi = p(t_1, \ldots, t_n) \implies$ alle $t_i \in \mathbf{Term}^0 \overset{1.}{\implies}$ alle $\mathrm{val}(\mathcal{S}, \beta, t_i) = \overline{t_i} \implies \mathrm{val}(\mathcal{S}, \beta, \phi) = \mathcal{I}(p)(\overline{t_1}, \ldots, \overline{t_n}) = tt \implies \mathcal{S}, \beta \models \phi$

- $\phi = \neg\exists x.\phi_1 \overset{\text{(H4)}}{\implies} [x/t](\neg\phi_1) \in \bar{H}$ für alle $t \in \mathbf{Term}^0 \overset{\text{Ind.}}{\implies} \mathcal{S}, \beta \models [x/t]\neg\phi_1$ für alle $t \in \mathbf{Term}^0 \overset{1.,2.}{\implies} \mathcal{S}, \beta\{x \leftarrow \overline{t}\} \not\models \phi_1$ für alle $t \in \mathbf{Term}^0$, also für alle $\overline{t} \in \mathcal{D}$ $\implies \mathcal{S}, \beta \not\models \exists x.\phi_1 \implies \mathcal{S}, \beta \models \phi$.

- $\phi = s \doteq t \implies s \sim t \implies \mathrm{val}(\mathcal{S}, \beta, s) \overset{1.}{=} \overline{s} = \overline{t} \overset{1.}{=} \mathrm{val}(\mathcal{S}, \beta, t) \implies \mathcal{S}, \beta \models \phi$.

- $\phi = \neg(s \doteq t) \overset{\text{H0}}{\implies} s \doteq t \notin \bar{H} \implies s \not\sim t$, und weiter wie im vorigen Fall.

Zu 4. Sei $\mathcal{S}, \beta \models \exists x.\phi$. Anwendung von (H5) auf den ϵ-Term $\epsilon x.\phi' := [\beta](\epsilon x.\phi)$ ergibt, daß entweder $\neg\exists x.\phi' \in \bar{H}$, oder $[x/\epsilon x.\phi'](\phi') \in \bar{H}$.

- Falls $\neg\exists x.\phi' \in \bar{H}$, folgt mit 3.: $\mathcal{S} \models \neg\exists x.\phi' \implies \mathcal{S} \models [\beta](\neg\exists x.\phi') \overset{2.}{\implies} \mathcal{S}, \beta \not\models \exists x.\phi$. Dieser Fall kann also nicht eintreten.

- Falls $[x/\epsilon x.\phi'](\phi') \in \bar{H}$, folgt durch Umschreiben der Formel mit der Definition von ϕ': $[\beta]_x^{[\beta](\epsilon x.\phi)}(\phi) \in \bar{H} \overset{3.}{\implies} \mathcal{S} \models [\beta]_x^{[\beta](\epsilon x.\phi)}(\phi) \overset{2.}{\implies} \mathcal{S}, \beta\{x \leftarrow \mathrm{val}(\mathcal{S}, \beta, [\beta](\epsilon x.\phi))\} \models \phi \overset{2.}{\implies} \mathcal{S}, \beta\{x \leftarrow \mathrm{val}(\mathcal{S}, \beta, \epsilon x.\phi)\} \models \phi \implies \mathcal{S}, \beta\{x \leftarrow \mathcal{A}(\epsilon x.\phi, \beta)\} \models \phi$.

Zu 5. Seien $\epsilon x.\phi$, $\epsilon y.\psi \in \mathbf{Term}^+$ mit $\mathrm{Ext}(\mathcal{S},\beta,\epsilon x.\phi) = \mathrm{Ext}(\mathcal{S},\beta,\epsilon y.\psi)$. Seien $\epsilon x.\phi' := [\beta](\epsilon x.\phi)$ und $\epsilon y.\psi' := [\beta](\epsilon y.\psi)$. Anwendung von (H7) auf diese beiden Grundterme ergibt folgende zwei Fälle:

- Fall 1: $\neg([x/\mathbf{c}](\phi') \leftrightarrow [y/\mathbf{c}](\psi')) \in \bar{H}$ mit $\mathbf{c} = \epsilon z.\neg(([x/z](\phi') \leftrightarrow [y/z](\psi'))$. Mit 3. und der auf Seite 56 angegebenen Eigenschaft $(*)$ folgt:

$$\begin{aligned} & \mathrm{val}(\mathcal{S},\beta,[x/\mathbf{c}](\phi')) \neq \mathrm{val}(\mathcal{S},\beta,[y/\mathbf{c}](\psi')) \\ \stackrel{1.,2.}{\Longrightarrow}\; & \mathrm{val}(\mathcal{S},\beta\{x \leftarrow \overline{\mathbf{c}}\},\phi) \neq \mathrm{val}(\mathcal{S},\beta\{y \leftarrow \overline{\mathbf{c}}\},\psi) \\ \Longrightarrow\; & \mathrm{Ext}(\mathcal{S},\beta,\epsilon x.\phi) \neq \mathrm{Ext}(\mathcal{S},\beta,\epsilon y.\psi), \end{aligned}$$

  da $\overline{\mathbf{c}}$ in genau einer der beiden Extensionen liegt. Damit ist dieser Fall ausgeschlossen.

- Fall 2: $\epsilon x.\phi \doteq \epsilon y.\psi \in \bar{H}$. Dann gilt $\epsilon x.\phi' \sim \epsilon y.\psi' \Longrightarrow \overline{[\beta](\epsilon x.\phi)} = \overline{[\beta](\epsilon y.\psi)}$ $\Longrightarrow \mathcal{A}(\epsilon x.\phi,\beta) = \mathcal{A}(\epsilon y.\psi,\beta)$.

■

5.7. Umbenennung von gebundenen Variablen

Eine mit der Substitutivität verwandte Eigenschaft der klassischen Prädikatenlogik ist deren Unempfindlichkeit gegenüber Umbenennung von gebundenen Variablen: Man kann die in einer Formel gebundenen Variablen umbenennen, ohne dadurch die Gültigkeit der Formel zu beeinflussen, falls keine Namenskonflikte zwischen Variablen auftauchen. Diese Eigenschaft ist aus zwei Gründen interessant: Erstens entspricht sie der Intuition, daß die Semantik einer Formel nicht von den *Namen* der darin gebundenen Variablen abhängig sein soll. In einem interaktiv verwendeten Kalkül hätte eine Semantik, die gegenüber gebundener Umbenennung unempfindlich ist, also Vorteile. Zweitens wird gebundene Umbenennung oft als technisches Werkzeug in Beweisen *über* Kalküle verwendet, um „Zusammenstöße“ von Variablen beim Ersetzen von Teiltermen vermeiden zu können.

Wie die Substitutivität ist auch Stabilität gegenüber gebundener Umbenennung nicht für alle Strukturen der ϵ-Logik gegeben. Man kann sie jedoch als zusätzliche Eigenschaft von Strukturen fordern:

5.24 Definition (Stabilität gegenüber gebundener Umbenennung)
Eine Struktur $\mathcal{S} = (\mathcal{D},\mathcal{I},\mathcal{A})$ heißt **stabil gegenüber gebundener Umbenennung**, *oder* **U-Struktur**, *falls für alle ϵ-Terme $\epsilon x.\phi$ mit einem (nicht notwendigerweise echten) Teilterm $\epsilon y.\psi$ oder einer Teilformel $\exists y.\psi$, und allen Variablen $z \in \mathbf{V} \setminus \mathrm{fv}(\psi)$ unter allen Belegungen β gilt:*

$$\mathcal{A}(\epsilon x.\phi,\beta) = \mathcal{A}(\overline{\epsilon x.\phi},\beta),$$

wobei $\overline{\epsilon x.\phi}$ aus $\epsilon x.\phi$ entsteht, indem der Teilterm $\epsilon y.\psi$ durch $\epsilon z.[y/z](\psi)$, bzw. die Teilformel $\exists y.\psi$ durch $\exists z.[y/z](\psi)$ ersetzt wird.

Eine **SU-Struktur** *ist eine Struktur, die sowohl* **S-Struktur**, *als auch* **U-Struktur** *ist.*

Ähnlich wie dies für die Substitutivität im Hierarchielemma 5.13 geschehen ist, kann gezeigt werden, daß alle E-Strukturen auch U-Strukturen, und damit SU-Strukturen sind. Es ergibt sich also die folgende erweiterte Hierarchie:

$$\begin{array}{ccccc}
 & & \text{E-Strukturen} & & \\
 & & \cup & & \\
 & & \text{SU-Strukturen} & & \\
 & \rotatebox{45}{$\subset$} & & \rotatebox{135}{$\subset$} & \\
\text{S-Strukturen} & & & & \text{U-Strukturen} \\
 & \rotatebox{135}{$\subset$} & & \rotatebox{45}{$\subset$} & \\
 & & \text{I-Strukturen} & &
\end{array}$$

Eine entsprechende Hierarchie ergibt sich für die jeweils erfüllbaren Formeln, und durch Umkehrung der Inklusionen für die allgemeingültigen Formeln.

Da für ein interaktives Beweissystem ein für die extensionale Semantik vollständiger Kalkül verwendet werden soll, können mit diesem auch U-allgemeingültige Formeln gezeigt werden. Dazu ist allerdings die *ext*-Regel nötig. Für die U-gültige Gleichheit $\epsilon x.p(x) = \epsilon y.p(y)$ findet man zum Beispiel folgenden $\mathcal{G}_E$-Beweis:

$$\dfrac{\dfrac{\dfrac{p(\mathbf{c}) \vdash p(\mathbf{c}) \quad p(\mathbf{c}) \vdash p(\mathbf{c})}{\vdots}}{\vdash p(\mathbf{c}) \leftrightarrow p(\mathbf{c})} \qquad \epsilon x.p(x) = \epsilon y.p(y) \vdash \epsilon x.p(x) = \epsilon y.p(y)}{\vdash \epsilon x.p(x) = \epsilon y.p(y)}\; ext$$

Das ist für eine so einfache Aussage ein recht komplizierter Beweis. Der Beweis der vergleichbaren Formel $\exists x.p(x) \leftrightarrow \exists y.p(y)$ bereitet im klassischen Sequenzenkalkül allerdings die gleichen Probleme, und dies stellt für das interaktive Beweisen kaum ein Problem dar, da solche Äquivalenzen in der Praxis nicht besonders oft auftreten. Es ist also zu erwarten, daß auch die Behandlung von U-gültigen Formeln mit der *ext*-Regel in der Praxis nur selten nötig ist.

Man würde sich jedoch wünschen, daß auch der automatische Beweiser mit U-gültigen Formeln umgehen kann. Dazu wäre ein vollständiger Kalkül für die SU-Semantik nötig, der ohne *ext* oder ähnliche indeterministische Regeln auskommt. Es ist möglich, die Menge der Axiome des Kalküls $\mathcal{G}_S$ so zu erweitern, daß jede Sequenz $\Gamma \vdash \Delta$ mit $\phi \in \Gamma$ und $\psi \in \Delta$ ein Axiom ist, falls ϕ und ψ bis auf gebundene Umbenennung gleich sind. Der resultierende Kalkül (mit den gleichen Regeln wie $\mathcal{G}_S$) ist vollständig für die SU-Semantik ohne Gleichheit, wie man zeigen kann, indem man im Vollständigkeitsbeweis für $\mathcal{G}_S$ von Grundtermen zu Äquivalenzklassen bezüglich gebundener Umbenennung übergeht. Nimmt man die Gleichheit dazu, so muß in den entsprechenden Regeln darauf geachtet werden, daß statt der Gleichheit von zu ersetzenden Teiltermen nur die Gleichheit bis auf gebundene Umbenennung verlangt wird. Eine geeignete Regel für einen Grundkalkül wäre also

$$\frac{s \doteq t, [y/t](\phi), [y/s'](\phi), \Gamma \vdash \Delta}{s \doteq t, [y/s'](\phi), \Gamma \vdash \Delta}$$

falls s und s' bis auf gebundene Umbenennung gleich sind. Im Kalkül mit Metavariablen muß diese Erweiterung natürlich berücksichtigt werden.

Ob gebundene Umbenennung in ein integriertes Beweissystem aufgenommen werden sollte, kann in diesem Rahmen nicht endgültig entschieden werden. Der Vorteil wäre die Möglichkeit, eine größere Menge allgemeingültiger Formeln *automatisch* beweisen zu können. Der Nachteil ist eine kompliziertere Implementierung, die – gemäß der oben erwähnten Erfahrungen beim interaktiven Beweisen – wahrscheinlich nur selten tatsächlich einen Vorteil bringt. Da diese Aspekte schwer gegeneinander abzuwägen sind, wurde in dieser Arbeit keine Entscheidung über die Verwendung gebundener Umbenennung getroffen. Sie wird im Rest dieser Arbeit nicht weiter betrachtet, da ihre Berücksichtigung technisch aufwendig ist, jedoch keine wesentlichen neuen Aspekte aufwirft.

Am Anfang dieses Abschnitts wurde erwähnt, daß die gebundene Umbenennung ein häufig benutztes technisches Hilfsmittel in Metabeweisen ist. In den bisherigen Beweisen wurde sie jedoch nicht verwendet, und auch im folgenden wird dies nicht nötig sein. Die Umbenennung gebundener Variablen ist nämlich in eingeschränkter Form auch in S-Strukturen möglich: Quantifizierte Variablen können umbenannt werden, sofern der betroffene Quantor nicht innerhalb eines ϵ-Terms liegt. Die Umbenennung von Variablen *innerhalb* von ϵ-Termen wird für die Beweise nicht gebraucht.

5.8. Erweiterungen der ϵ-Logik

In diesem Abschnitt sollen zwei syntaktische Erweiterungen der ϵ-Logik vorgestellt werden, die bislang nicht erwähnt wurden, da sie die Diskussion kompliziert hätten, ohne wirklich neue Aspekte zu beinhalten.

5.8.1. α-Terme

Will man ein integriertes Beweissystem unter Verwendung von ϵ-Termen implementieren, so wird man sicherlich den Allquantor und Junktoren wie $\vee$, $\to$ und $\leftrightarrow$ in den Kalkül aufnehmen und entsprechende Regeln für sie einführen, damit der Benutzer direkt mit ihnen umgehen kann. Welche Regeln nötig sind, ergibt sich unmittelbar aus der Expansion der Abkürzungen. Hier soll die Behandlung von Allquantoren im Sukzedent von Sequenzen betrachtet werden. Mit der Abkürzung $\forall x.\phi := \neg\exists x.\neg\phi$ ergibt sich folgende Ableitung:

$$\dfrac{\dfrac{\dfrac{\Gamma \vdash [x/\epsilon x.\neg\phi](\phi), \Delta}{[x/\epsilon x.\neg\phi](\neg\phi), \Gamma \vdash \Delta}\,\neg l}{\exists x.\neg\phi, \Gamma \vdash \Delta}\,\exists l}{\Gamma \vdash \forall x.\phi, \Delta}\,\neg r$$

Die Regel zur Behandlung von Allquantoren rechts muß also lauten:

$$\frac{\Gamma \vdash [x/\epsilon x.\neg\phi](\phi), \Delta}{\Gamma \vdash \forall x.\phi, \Delta}\,\forall r$$

Das Bilden eines ϵ-Terms für die *negierte* Formel in der oberen Sequenz ist intuitiv nicht ganz einfach zu verstehen. Es ist zu zeigen, daß ϕ für alle Werte von x gilt. Angenommen, es existiere ein Wert für x, für den ϕ *nicht* gilt. Dann stellt die Auswahleigenschaft sicher, daß $\epsilon x.\neg\phi$ einen solchen Wert liefert. Kann gezeigt werden, daß ϕ sogar für diesen Wert gilt, muß die Annahme verworfen werden, also gilt ϕ für alle x.

Es ist zu bemerken, daß diese Problematik der ϵ-Terme nicht neu ist: Die Skolemfunktionen der verschiedenen δ-Regeln des Tableaukalküls mit freien Variablen bringen genau dieselbe Schwierigkeit mit sich.

Der übliche informelle Ansatz, um eine allquantifizierte Aussage zu beweisen, besteht darin, die Aussage für ein „beliebiges“ x zu beweisen. Dieser Intuition könnte entgegengekommen werden, indem man zu den ϵ-Termen duale α-Terme einführt.[19] Dabei kann für die theoretische Betrachtung $\alpha x.\phi$ als Abkürzung für $\epsilon x.\neg\phi$ angesehen werden. Es ergibt sich dann die Regel:

$$\frac{\Gamma \vdash [x/\alpha x.\phi](\phi), \Delta}{\Gamma \vdash \forall x.\phi, \Delta}\ \forall r$$

Während $\epsilon x.\phi$ umgangssprachlich „ein x, für das ϕ gilt, vorausgesetzt ein solches existiert“ bezeichnet, kann $\alpha x.\phi$ gelesen werden als „ein beliebiges x, für das ϕ zu zeigen ist“. Diese Umschreibung entspricht der Einführung von $[x/\alpha x.\phi](\phi)$ auf der rechten Seite der Sequenz. Für einen Beweis durch Widerspruch kann man annehmen, ϕ gelte für irgendein x nicht, und die Formel $\neg([x/\alpha x.\phi](\phi))$ auf der linken Seite verwenden. Hier bietet sich die Umschreibung „ein x für das ϕ nicht gilt, sofern ein solches existiert“ an.

Oft beschreiben in einer Beweisverpflichtung die Formeln im Antezedenten ein *intendiertes Modell*, während von den Formeln im Sukzedenten zu zeigen ist, daß diese dort auch gelten. In dem intendierten Modell gelten also zunächst alle Formeln der Sequenz. Dies ändert sich erst bei der Anwendung der Negationsregeln. Eine Formel $\exists x.\phi$ im Antezedenten weist also auf die Existenz eines Elements im intendierten Modell hin, das ϕ erfüllt. Dieses Element wird durch $\epsilon x.\phi$ bezeichnet. Im intendierten „Normalfall“ bezeichnet also $\epsilon x.\phi$ ein Element, für das ϕ gilt. Modelle, in denen $[x/\epsilon x.\phi](\phi)$ nicht gilt, bilden die „Ausnahme“. Wenn nun $\forall x.\phi$ zu zeigen ist, und im intendierten Modell auch tatsächlich gilt, dann gilt eben *nicht* $[x/\epsilon x.\neg\phi](\neg\phi)$. Die Verwendung des α-Terms $\alpha x.\phi$ verdeutlicht den Bezug zum intendierten Modell.

Da im Sequenzenkalkül nicht eindeutig zwischen direkten Beweisen und Beweisen durch Widerspruch unterschieden werden kann, und durch die Negationsregeln Formeln auf die andere Seite von Sequenzen wandern können, muß der Benutzer jedenfalls zwischen den verschiedenen Intuitionen wechseln. Er muß mit dem Verhältnis zwischen ϵ- und α-Termen vertraut sein.

Für die Implementierung muß man α-Terme beim Astabschluß und bei der Gleichheitseinsetzung berücksichtigen, damit zum Beispiel $p(\epsilon x.\neg\phi) \vdash p(\alpha x.\phi)$ als Axiom erkannt wird. Alternativ könnte auch eine Regel zur Verfügung gestellt werden, die α-Terme in ϵ-Terme konvertiert.

Die Verwendung von α-Termen hätte folgende Vor- und Nachteile:

+ Die Logik und der Kalkül werden symmetrischer.

+ Die $\forall r$-Regel wird intuitiv leichter faßbar.

+ α-Terme bieten die Möglichkeit, das Verhalten des bezeichneten Objekts in einem intendierten Modell zu reflektieren.

– Der Benutzer verwendet meistens Abkürzungen für ϵ-Terme. Diese wird er von vornherein so wählen, daß sie seiner Intuition entsprechen. Wenn er unabgekürzte ϵ-Terme verwendet, wird er wahrscheinlich auch mit der bisherigen $\forall r$-Regel zurecht kommen.

[19] Der Name wurde gewählt, um anzudeuten, daß ϵ-Terme von $\exists$-Quantoren, und α-Terme von $\forall$-Quantoren stammen. α mag auch an engl. *arbitrary* erinnern. In der Tat sind die α-Terme mit den *arbitrary objects* von [9] eng verwandt.

- α-Terme bedeuten zusätzlichen Implementierungsaufwand, und daher neue Fehlerquellen bei der Implementierung.

- Für den automatischen Teil des Beweissystems bringen α-Terme keine Vorteile, sondern sogar Effizienzeinbußen, da die Überprüfung von Astabschlüssen komplizierter wird.

Ob die Verwendung von α-Termen echte Vorteile für die Interaktive Benutzung bringt, wird wohl nur durch die praktische Arbeit mit einer Implementierung geklärt werden können. Sie sollen im Rest dieser Arbeit jedenfalls nicht weiter verwendet werden.

5.8.2. Mehrstellige ϵ-Terme

Eine andere syntaktische Erweiterung von ϵ-Logiken betrifft Reihungen gleicher Quantoren. Man betrachte etwa die Sequenz $\exists x.\exists y.\exists z.p(x,y,z) \vdash$. In einem klassischen Sequenzenkalkül ergibt sich dafür die Ableitung:

$$\dfrac{\dfrac{\dfrac{p(x',y',z') \vdash}{\exists z.p(x',y',z) \vdash}\ \exists r}{\exists y.\exists z.p(x',y,z) \vdash}\ \exists r}{\exists x.\exists y.\exists z.p(x,y,z) \vdash}\ \exists r$$

mit neu eingeführten Variablen x', y', z'.[20] Faßt man derartige Folgen von Anwendungen der $\exists l$-Regel zu einer Regel zusammen, erhält man

$$\frac{[x_i/x'_i](\phi), \Gamma \vdash \Delta}{\exists x_1 \ldots \exists x_k.\phi, \Gamma \vdash \Delta}$$

wobei die x'_i neue Variablen sind. Diese Form der Regel reflektiert sehr gut die Symmetrie zwischen den Variablen bei derartigen Reihungen von Existenzquantoren.

Benutzt man nun ϵ-Terme, so erhält man statt dessen die Ableitung

$$\dfrac{\dfrac{\dfrac{p(\mathbf{c}_x,\mathbf{c}_y,\mathbf{c}_z) \vdash}{\exists z.p(\mathbf{c}_x,\mathbf{c}_y,z) \vdash}\ \exists r}{\exists y.\exists z.p(\mathbf{c}_x,y,z) \vdash}\ \exists r}{\exists x.\exists y.\exists z.p(x,y,z) \vdash}\ \exists r$$

wobei

$$\begin{aligned}
\mathbf{c}_x &= \epsilon x.\exists y.\exists z.p(x,y,z),\\
\mathbf{c}_y &= \epsilon y.\exists z.p(\mathbf{c}_x,y,z),\\
\mathbf{c}_z &= \epsilon z.p(\mathbf{c}_x,\mathbf{c}_y,z).
\end{aligned}$$

Die Symmetrie zwischen den Variablen ist hier also nicht mehr erkenntlich.[21] Es ist aber auch eine symmetrischere Notation möglich: Wir führen sogenannte „mehrstellige

[20]Dies sind freie Variablen mit universeller Semantik im Sinne des klassischen Sequenzenkalküls, siehe Abschnitt 3.

[21]Die δ^{++}-Regel zeigt ein ähnliches Verhalten: Es werden sukzessive neue Skolemsymbole erzeugt für Formeln, die zuvor eingeführten Symbole enthalten. Betrachtet man nur die Skolemsymbole ohne die zugehörigen Formeln, bzw. Formelklassen, so ergibt sich natürlich ein symmetrischeres Bild. Dieses erhält man aber auch, wenn man die ϵ-Terme jeweils durch Abkürzungen ersetzt, die zugehörigen Formeln also nicht betrachtet. Die vorgestellte Problematik ist also keine Eigenheit der ϵ-Logiken, sondern ein generelles Problem von mächtigen Skolemisierungsverfahren.

ϵ-Terme“ ein, und schreiben $\epsilon_{x_i}(x_1,\ldots,x_k).\phi$, um die Auswahl eines Tupels von Trägerelementen anzudeuten, die ϕ erfüllen, und von diesem Tupel das der Variable x_i zugeordnete Element zu extrahieren.

Bevor die Semantik dieser Notation diskutiert wird, soll demonstriert werden, wie sie in der $\exists l$-Regel verwendet werden soll. Entsprechend zur obigen mehrfachen $\exists l$-Regel erhält man für mehrstellige ϵ-Terme:

$$\frac{[x_i/\epsilon_{x_i}(x_1,\ldots,x_k).\phi](\phi),\Gamma\vdash\Delta}{\exists x_1\ldots\exists x_k.\phi,\Gamma\vdash\Delta}$$

Die obige Ableitung würde nun lauten:

$$\frac{p(\mathbf{c}_x,\mathbf{c}_y,\mathbf{c}_z)\vdash}{\exists x.\exists y.\exists z.p(x,y,z)\vdash}$$

wobei

$$\begin{aligned}\mathbf{c}_x &= \epsilon_x(x,y,z).p(x,y,z),\\ \mathbf{c}_y &= \epsilon_y(x,y,z).p(x,y,z),\\ \mathbf{c}_z &= \epsilon_z(x,y,z).p(x,y,z).\end{aligned}$$

Mit mehrstelligen ϵ-Termen wird die Symmetrie auf notationeller Ebene also sehr gut wiedergespiegelt. Es fragt sich nun, was genau diese neuen Terme bedeuten sollen. Es wird sich zeigen, daß die Auswahl einer Semantik gewisse Probleme mit sich bringt.

Die Semantik für mehrstellige ϵ-Terme sollte sich jedenfalls daran orientieren, welche Eigenschaften man sich von diesen erhofft. Die Schreibweise $t_1\equiv t_2$ möge bedeuten, daß zwei Terme in einer gewissen Struktur und unter einer Belegung, die sich aus dem Zusammenhang ergeben, gleich ausgewertet werden. Erwünschte Eigenschaften sind dann etwa:

$$\epsilon_{x_i}(x_1,\ldots,x_k).\phi\equiv\epsilon_{x_i}.(x_{\pi(1)},\ldots,x_{\pi(k)}).\phi\quad\text{f. alle Perm. }\pi\text{ von }\{1,\ldots,k\},\qquad(C)$$

$$\epsilon_{x_1}(x_1,\ldots,x_k).\phi\equiv\epsilon x_1.\exists x_2\ldots\exists x_k.\phi,\qquad(A)$$

in allen Strukturen unter allen Belegungen. Dabei drückt (C) gerade die gewollte Symmetrie zwischen den Variablen aus. (A) soll sicherstellen, daß man bei Verwendung der bisherigen $\exists l$-Regel äquivalente ϵ-Terme erhält, die neue Regel also tatsächlich einer mehrmaligen Anwendung der alten Regel entspricht.

Wie das folgende Beispiel demonstriert, gibt es keine sinnvolle Semantik, unter der (C) und (A) für alle ϵ-Terme gelten, und die außerdem für die einstelligen ϵ-Terme extensional ist. Es wird nämlich eine Trägermenge $\mathcal{D}$ und eine Interpretation $\mathcal{I}$ angegeben, zu denen keine ϵ-Auswertung gefunden werden kann, die allen drei Anforderungen genügt. Damit würde der Einbettungssatz ungültig, die Logik also keine konservative Erweiterung der Prädikatenlogik.

5.25 Beispiel (Eigenschaften mehrstelliger ϵ-Terme)
Gegeben sei eine extensionale Struktur $\mathcal{S}=(\mathcal{D},\mathcal{I},\mathcal{A})$ mit $\mathcal{D}:=\{0,1,2\}$. $\mathcal{I}$ werte das Prädikat p gemäß folgender Tabelle aus:

$p(x,y)$		y		
		0	1	2
x	0	*ff*	*tt*	*ff*
	1	*ff*	*ff*	*tt*
	2	*tt*	*ff*	*ff*

Angenommen, die Struktur erfülle (C) und (A) für alle ϵ-Terme. Es gilt

$$\mathrm{Ext}(\mathcal{S}, \beta, \epsilon x.\exists y.p(x,y)) = \mathrm{Ext}(\mathcal{S}, \beta, \epsilon y.\exists x.p(x,y)) = \mathcal{D}$$

Also muß gelten:

$$\begin{aligned} \epsilon_x(x,y).p(x,y) &\equiv \epsilon x.\exists y.p(x,y) && \text{wegen } (A), \\ &\equiv \epsilon y.\exists x.p(x,y) && \text{wegen der Extensionalität,} \\ &\equiv \epsilon_y(y,x).p(x,y) && \text{wegen } (A), \\ &\equiv \epsilon_y(x,y).p(x,y) && \text{wegen } (C). \end{aligned}$$

Da es Paare (x, y) gibt, für die p wahr wird, muß

$$p(\epsilon_x(x,y).p(x,y), \epsilon_y(x,y).p(x,y))$$

in $\mathcal{S}$ gelten. Aber aus der Tabelle ist aber zu ersehen, daß p ungültig ist, wenn die Argumente gleich sind, womit ein Widerspruch hergestellt ist.

Es gibt also keine Semantik für mehrstellige ϵ-Terme, die zu allen Eigenschaften führt, die man von diesen intuitiv erwartet. Verzichtet man auf eine der beiden Eigenschaften (C) und (A), so kann man zwar eine Semantik angeben. Aber mehrstellige ϵ-Terme sollten die Logik klarer und verständlicher machen. So würden sie syntaktisch eine Symmetrie vorspiegeln, die nicht tatsächlich vorhanden ist, den Benutzer also eher in die Irre führen.

Der Verzicht auf mehrstellige ϵ-Terme richtet hingegen keinen großen Schaden an: Da der Benutzer meistens ohnehin Abkürzungen für ϵ-Terme verwendet, würde er sie nicht oft zu sehen bekommen. Und je weniger verschiedene Konzepte die Logik beinhaltet, desto einfacher ist sie zu handhaben.

5.9. Zusammenfassung

Es wurde die Notwendigkeit eines Skolemisierungs-Mechanismus in einer Logik mit Metavariablen identifiziert, und vorgeschlagen, dazu ϵ-Terme zu verwenden. Dies wurde unter anderem daran motiviert, daß ϵ-Terme intuitiv leichter zugänglich sind als Skolemfunktionen. Ferner wurde darauf verwiesen, daß ϵ-Skolemisierung stärker als δ^{++}-Skolemisierung ist, was in 6.4.3 bewiesen werden wird.

Für Logiken mit ϵ-Termen wurde eine Modell-Semantik vorgestellt, und verschiedene Varianten der Semantik von ϵ-Termen vorgestellt. Für die substitutive und die extensionale Semantik wurden vollständige Grundkalküle angegeben. Abschließend wurden einige semantische und syntaktische Varianten der ϵ-Logik diskutiert.

6. Gleichheitsbehandlung mit Constraints

Das Beweisen mit Gleichungen ist ein wichtiger Aspekt eines integrierten Beweissystems. Viele Problemstellungen lassen sich am natürlichsten unter Verwendung der Gleichheit formulieren, und die Umformung von Termen und Formeln durch Einsetzen von Gleichungen ist eine der wichtigsten Beweistechniken der Mathematik. Leider gehört die Gleichheitsbehandlung zu den großen Problemen automatischer Beweissysteme. Dies liegt im wesentlichen an der Symmetrie der Gleichheit, die bei unbeschränktem Einsetzen von Gleichungen zu nicht mehr handhabbaren Suchräumen führt. Für automatische Beweissysteme wurden spezielle Verfahren entwickelt, um diese Suchräume zu beschränken.

Für rein interaktive Beweissysteme stellt die Gleichheit kein so großes Problem dar: Da der Benutzer einen bestimmten Beweisplan verfolgt, weiß er auch, wann und wie er welche Gleichung anwenden muß. Die benötigten Regeln haben etwa die Form der Gleichheitsregeln des Kalküls $\mathcal{G}_E$, siehe Abb. 5.3. Es wird sich jedoch zeigen, daß diese Regeln bei der Einführung von Metavariablen Probleme bereiten. Man muß also einen komplizierteren Weg einschlagen, und sollte dabei doch zu einem Ergebnis gelangen, das sich vom Benutzer genauso intuitiv anwenden läßt wie die Regeln des Grundkalküls.

Angenehmerweise ermöglicht die Verwendung von sog. „Constraints“ die Lösung beider Probleme durch *ein* Konzept. Einerseits haben sich constraintbasierte Verfahren zur automatischen Gleichheitsbehandlung bewährt. Andererseits erlauben Constraints die Formulierung von Gleichheitsregeln, die sich für den Benutzer so verhalten wie Regeln des Grundkalküls, sofern er konkrete Instantiierungen für Metavariablen angibt.

Constraints haben sogar noch weitere Vorteile für ein integriertes System: wie bereits in Abschnitt 4.5 angedeutet wurde, können sie vom Benutzer dazu verwendet werden, den Suchraum des automatischen Beweisers einzuschränken, und sie erlauben es ihm, Metavariablen unterschiedlich konkret zu instantiieren.

Es bietet sich daher an, in diesem Rahmen die Gleichheit mit Constraints zu behandeln. Stellvertretend für bestehende constraintbasierte Verfahren zur automatischen Gleichheitsbehandlung soll hier gezeigt werden, wie sich das Verfahren von Beckert [4] in den hier vorgestellten Rahmen einbetten läßt. Da dieses Verfahren recht komplex ist, mag es als praxisrelevantes Beispiel gelten, wenn es darum geht, andere Verfahren zu akkomodieren.

Zunächst soll jedoch die Idee von Constraints anhand der beiden erwähnten Blickrichtungen erläutert werden, und zwar zunächst im Hinblick auf korrekte Gleichheitsregeln in einem Kalkül mit Metavariablen, und dann im Zusammenhang des automatischen Beweisens.

6.1. Gleichheitsbehandlung mit Metavariablen

Die Probleme der Gleichheitsbehandlung in einem Kalkül mit Metavariablen sollen zunächst an einem einfachen Beweis demonstriert werden. Zu beweisen sei folgende Formel:

$$\exists x.((f(a) \doteq a \to f(x) \doteq a) \wedge x \doteq f(a))$$

Wenn man zum Kalkül $\mathcal{G}_E$ die leicht abzuleitende Regel

$$\frac{\phi, \Gamma \vdash \psi, \Delta}{\Gamma \vdash \phi \to \psi, \Delta} \to r$$

hinzunimmt, und den Quantor mit $f(a)$ instantiiert, findet man folgenden Beweis, wobei nicht mehr benötigte Formeln wie üblich weggelassen werden:

$$\cfrac{\cfrac{\cfrac{\cfrac{\cfrac{f(a) \doteq a \vdash a \doteq a}{f(a) \doteq a \vdash f(a) \doteq a}\stackrel{\rightarrow}{=}r}{f(a) \doteq a \vdash f(f(a)) \doteq a}\stackrel{\rightarrow}{=}r}{\vdash f(a) \doteq a \to f(f(a)) \doteq a}\to r \qquad \vdash f(a) \doteq f(a)}{\vdash (f(a) \doteq a \to f(f(a)) \doteq a) \wedge f(a) \doteq f(a)}\wedge r}{\vdash \exists x.((f(a) \doteq a \to f(x) \doteq a) \wedge x \doteq f(a))}\exists r$$

Der Beweis ist geschlossen, da beide Blätter Formeln der Form $t \doteq t$ im Sukzedenten haben. Was geschieht, wenn man x statt dessen mit einer Metavariablen X instantiiert? Man gelangt zu folgendem Baum:

$$\cfrac{\cfrac{\cfrac{f(a) \doteq a \vdash f(X) \doteq a}{\vdash f(a) \doteq a \to f(X) \doteq a}\to r \qquad \vdash X \doteq f(a)}{\vdash (f(a) \doteq a \to f(X) \doteq a) \wedge X \doteq f(a)}\wedge r}{\vdash \exists x.((f(a) \doteq a \to f(x) \doteq a) \wedge x \doteq f(a))}\exists r$$

Der rechte Ast ist natürlich genau dann geschlossen, wenn X mit dem Term $f(a)$ belegt wird. Auf dem linken Ast können aber die Gleichheitsregeln nicht angewandt werden, ohne eine bestimmte Belegung von X vorauszusetzen.

Eine mögliche Verfahrensweise bestünde darin, X auf einen bestimmten Term „festzulegen“, und damit die vorhandenen Terme zu konkretisieren. Dies entspräche den im Tableaukalkül von Fitting [10] verwendeten Gleichheitsregeln. Dort muß vor Anwendung einer Gleichung durch eine Substitution dafür gesorgt werden, daß die entsprechenden Terme gleich sind. Es wird also ein *Unifikator* ermittelt und *auf den gesamten Beweis* angewandt, um alle anderen Vorkommen der freien Variablen ebenfalls zu konkretisieren. Danach wird die Gleichung angewandt. Diese Substitutionsanwendung ist natürlich eine destruktive Operation: Wird für die Variable X der Term a eingesetzt, so wird die Gleichung $f(X) \doteq c$ durch $f(a) \doteq c$ ersetzt. Wenn zum Schließen des Beweises aber statt dessen $f(b) = c$ benötigt wird, muß die Substitution durch *backtracking* rückgängig gemacht werden, und b für X eingesetzt werden.

Derartige destruktive Operationen sind für ein System, das auch interaktiv benutzt werden soll, von Nachteil: in Abschnitt 2.2 wurde darauf hingewiesen, daß es für den menschlichen Beweiser wichtig ist, die Beweisgeschichte nachvollziehen zu können. Auch die in 4.4 vorgeschlagene Trennung von Beweisbaum und Substitutionseinschränkung wäre dann nicht mehr gegeben.

Eine Variante dieser Idee bestünde darin, zu jedem Ast des Beweisbaums mitzuführen, unter welchen Substitutionen für die Metavariablen die Regelanwendungen auf dem Ast korrekt sind. Statt bei einer Gleichheitsanwendung die unifizierende Substitution auf den Beweis anzuwenden, markiert man den Ast also mit der Bedingung, daß die Instantiierung der Metavariablen die Terme unifiziert. Da diese Bedingungen mit jeder Anwendung einer Gleichheitsregel wachsen, und die Beweisgeschichte nachvollziehbar bleiben soll, markiert man statt der Äste lieber die Sequenzen selber, und bezieht sich für den Abschluß auf die Bedingungen in den Blättern. Dekoriert man etwa den Sequenzenoperator $\vdash$ mit (Mengen von) syntaktischen Gleichungen,[1] erhält man folgenden Beweis für das Beispiel:

$$
\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{f(a) \doteq a \overset{X=f(a)}{\vdash} a \doteq a}{f(a) \doteq a \overset{X=f(a)}{\vdash} f(a) \doteq a}\,\overset{\rightarrow}{\doteq}r}{f(a) \doteq a \vdash f(X) \doteq a}\,\overset{\rightarrow}{\doteq}r}{\vdash f(a) \doteq a \rightarrow f(X) \doteq a}\,\rightarrow r \qquad \vdash X \doteq f(a)}{\vdash (f(a) \doteq a \rightarrow f(X) \doteq a) \wedge X \doteq f(a)}\,\wedge r}{\vdash \exists x.((f(a) \doteq a \rightarrow f(x) \doteq a) \wedge x \doteq f(a))}\,\exists r
$$

Dieser Beweis ist mit der Grundsubstitution $\sigma := [X/f(a)]$ geschlossen, weil

1. jedes Blatt unter σ ein Axiom ist, und
2. die Bedingungen in jedem Blatt unter σ erfüllt sind.

Mit diesem Verfahren wäre die separate Rücksetzbarkeit gesichert, da unter einer anderen Instantiierung der Metavariablen diejenigen Äste als offen erkannt werden, deren Bedingungen nicht mehr erfüllt sind. Leider wird dieses Verfahren aber in der Praxis nicht sinnvoll sein: Da die Einschränkungen nur stärker werden können, wenn der Baum wächst, kann ein Ast überhaupt nur dann jemals geschlossen werden, wenn die Instantiierung der Metavariablen seine Bedingung erfüllen. Wenn die Anwendung einer Gleichheit eine Fehlentscheidung war, wird zum Schließen des Beweises nötige Instantiierung in der Regel die Bedingung für diese Gleichheit nicht erfüllen. Damit bleibt einem nichts anderes übrig, als den Beweisbaum an der Stelle der fraglichen Gleichheitsanwendung abzuschneiden. Damit verliert man aber auch alle Beweisschritte, die danach auf dem Ast angewandt wurden. Dies ist unangenehm, da eventuell viele dieser Schritte überhaupt nicht von der falschen Regelanwendung abhängen, sondern für den korrekten Beweis durchaus sinnvoll sind.[2]

Dieses Problem wird jedoch durch eine kleine Änderung aus der Welt geschafft: Statt ganze Äste, bzw. Sequenzen mit Bedingungen an die Instantiierung der Metavariablen zu versehen, heftet man solche, als *Constraint* bezeichnete Bedingungen an die einzelnen Formeln einer Sequenz. Die abschließende Substitution muß dann nicht mehr für jeden *Ast* einer bestimmten Bedingung genügen, sondern sie muß die Constraints der Formeln erfüllen, die für den Astabschluß verwendet werden. Schreibt man eine Formel ϕ mit

[1] In diesen Bedingungen bezeichnet „$=$“ die syntaktische Gleichheit von Termen, deren Gültigkeit von einer Instantiierung der Metavariablen abhängt. Diese ist von der Objekt-Gleichheit „$\doteq$“, deren Gültigkeit von einem Modell abhängt, verschieden!

[2] In einem Kalkül, der nicht auf einer strengen Baumstruktur beruht, kann die Situation anders sein, siehe Kapitel 8.

Constraint C als $\phi \ll C$, so erhält man als Beweis:

$$
\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{f(a) \doteq a \vdash a \doteq a \ll X = f(a)}{f(a) \doteq a \vdash f(a) \doteq a \ll X = f(a)}\,\overset{\rightarrow}{\doteq} r}{f(a) \doteq a \vdash f(X) \doteq a}\,\overset{\rightarrow}{\doteq} r}{\vdash f(a) \doteq a \rightarrow f(X) \doteq a}\,\rightarrow r \qquad \vdash X \doteq f(a)}{\vdash (f(a) \doteq a \rightarrow f(X) \doteq a) \wedge X \doteq f(a)}\,\wedge r}{\vdash \exists x.((f(a) \doteq a \rightarrow f(x) \doteq a) \wedge x \doteq f(a))}\,\exists r
$$

Dieser Beweis ist mit der Grundsubstitution $[X/f(a)]$ geschlossen, weil *jedes Blatt unter der Substitution ein Axiom ist, wenn man Formeln wegläßt, deren Constraint von der Substitution nicht erfüllt wird.* Die meisten automatischen Beweiser hätten im linken Ast zunächst versucht, den Beweis mit $[X/a]$ zu schließen. Sehen wir was geschieht:[3]

$$
\dfrac{\dfrac{\dfrac{\dfrac{f(a) \doteq a \vdash a \doteq a \ll X = a, f(X) \doteq a}{f(a) \doteq a \vdash f(X) \doteq a}\,\overset{\rightarrow}{\doteq} r}{\vdash f(a) \doteq a \rightarrow f(X) \doteq a}\,\rightarrow r \qquad \vdash X \doteq f(a)}{\vdash (f(a) \doteq a \rightarrow f(X) \doteq a) \wedge X \doteq f(a)}\,\wedge r}{\vdash \exists x.((f(a) \doteq a \rightarrow f(x) \doteq a) \wedge x \doteq f(a))}\,\exists r
$$

Der linke Ast wäre unter einer Instantiierung σ mit $\sigma(X) = a$ geschlossen, da dann der Constraint der Formel $a \doteq a$ erfüllt wäre. Der rechte Ast benötigt hingegen $\sigma(X) = f(a)$. Diese Bedingungen sind unvereinbar, es existiert also keine Instantiierung, unter der der Beweis geschlossen wäre. Aber die Gleichung $f(X) \doteq a$ ohne Constraint steht noch auf dem Ast, da die Gleichheitsregeln die ursprünglichen Formeln nicht entfernen. Es können also ausgehend von *diesem* Baum die beiden richtigen Anwendungen der Gleichheitsregeln durchgeführt werden, um den Baum zu schließen. Ungeschickte Anwendung von Gleichheitsregeln „verbaut“ also nicht den Weg zum richtigen Beweis, der Kalkül ist *beweiskonfluent.*

Es ist vielleicht nicht einzusehen, warum der Benutzer mit einem solchen Kalkül genauso einfach arbeiten kann wie mit einem Grundkalkül. Dies ist eine Frage der Beweispräsentation, deren Beantwortung einige technische Hilfsmittel erfordert. Es wird in Abschnitt 6.6 darauf eingegangen.

6.2. Constraintbasierte Verfahren zur automatischen Gleichheitsbehandlung

Das Hauptproblem der automatischen Gleichheitsbehandlung sind die häufig auftretenden großen Suchräume. Man erinnere sich, daß im Vollständigkeitsbeweis für den Kalkül $\mathcal{G}_E$ bei der Konstruktion der Hintikka-Menge verlangt wurde, daß *alle* Gleichungen an jeder Stelle angewandt werden, an der dies möglich ist. Hat man etwa eine Gleichung $a \doteq f(a,a)$ und eine Formel $p(a)$, so müssen sukzessive alle Formeln $p(f(a,a))$, $p(f(f(a,a),a))$, $p(f(a,f(a,a)))$, usw. erzeugt werden.

Es gibt aber Verfahren, um diesen Suchraum zu beschränken. Die erfolgreichsten dieser Verfahren beruhen auf folgender Idee: Gleichungen werden nur in einer Richtung

[3] Bei den bisherigen Beispielen wurde die Formel $f(X) \doteq a$ im linken Ast nach der ersten Gleichheitsanwendung weggelassen, da sie nicht mehr gebraucht wurde, und auch keine neuen Abschlüsse ermöglicht hätte. Hier wird sie dennoch dazugeschrieben, da sich der folgende Absatz auf sie bezieht.

angewandt, und zwar so, daß die von der Anwendung betroffenen Formeln (oder Terme) bezüglich einer bestimmten Ordnung kleiner werden. Wenn diese Ordnung noethersch ist, werden die Formeln irgendwann minimal bezüglich dieser Ordnung, es sind dann also keine Gleichungen mehr anwendbar. In einem Grundkalkül ist diese Einschränkung relativ leicht einzuführen, wie folgendes Beispiel demonstriert.

Zu beweisen sei die Formel

$$\exists x.\exists y.((g(x) \doteq f(b) \wedge g(y) \doteq a \wedge p(f(b))) \rightarrow p(a)).$$

Man wählt nun eine totale, noethersche Ordnung auf den Grundtermen, etwa die lexikographische Pfadordnung $\prec$ zur Vorordnung $a < b < f < g$.[4] Man beschränkt sich bei der Anwendung von Gleichungen auf die Richtung, die einen Term durch einen kleineren ersetzt. In diesem Beispiel ist die rechte Seite jeder Gleichung kleiner als die linke. Wenn beide Quantoren mit b instantiiert werden, erhält man, nach Erledigung der Aussagenlogik, folgende Ableitung:

$$\cfrac{\cfrac{p(a), f(b) \doteq a, g(b) \doteq f(b), g(b) \doteq a, p(f(b)) \vdash p(a)}{f(b) \doteq a, g(b) \doteq f(b), g(b) \doteq a, p(f(b)) \vdash p(a)} \;\stackrel{\rightarrow}{=} l}{g(b) \doteq f(b), g(b) \doteq a, p(f(b)) \vdash p(a)} \;\stackrel{\rightarrow}{=} l$$

Damit ist der Beweis offenbar geschlossen. Man kann zeigen, daß ein Grundkalkül mit dieser Einschränkung vollständig ist. Leider sind Grundkalküle für das automatische Beweisen aus anderen Gründen unbrauchbar. Die folgende Untersuchung bezieht sich auf einen Sequenzenkalkül mit Metavariablen. Die an diesem beobachteten Phänomene sind natürlich dieselben wie bei einem Tableaukalkül mit freien Variablen, aber sogar beim Resolutionskalkül kommt man auf ähnliche Ergebnisse. Dies hängt damit zusammen, daß alle diese Kalküle sich in der Behandlung der Prädikatenlogik ohne Gleichheit voneinander unterscheiden. Das Einsetzen von Gleichungen tritt aber – mit den damit verbundenen Problemen – überall auf.

Versuchen wir also wieder, die gegebene Ableitung in einen Kalkül mit Metavariablen zu übertragen. Die Quantoren werden nun mit Metavariablen X, Y instantiiert, und man erhält die Sequenz

$$g(X) \doteq f(b), g(Y) \doteq a, p(f(b)) \vdash p(a)$$

Die erste dieser Gleichungen kann nun nur dann auf die zweite angewandt werden, wenn die abschließende Substitution X und Y den gleichen Grundterm zuordnet. Im letzten Abschnitt wurde vorgeschlagen, diese Bedingung durch einen an die Formel gehefteten Constraint auszudrücken. Damit erhielte man die Sequenz

$$f(b) \doteq a \ll X = Y, g(X) \doteq f(b), g(Y) \doteq a, p(f(b)) \vdash p(a).$$

Damit ist aber noch nicht gewährleistet, daß die linke Seite der Gleichung $g(X) \doteq f(b)$ größer ist als die rechte.[5] Hierbei handelt es sich also um eine weitere Einschränkung. Wenn solche Ordnungseinschränkungen auch in die Constraints aufgenommen werden,

[4]Die Ordnung muß noch gewisse syntaktische Verträglichkeitseigenschaften aufweisen. Darauf braucht an dieser Stelle jedoch nicht eingegangen zu werden. Zur Definition der lexikographischen Pfadordnung, siehe etwa [2]. Reduktionsordnungen für ϵ-Terme werden in Abschnitt 6.5 besprochen.

[5]Bei diesem einfachen Beispiel ist dies tatsächlich für alle Instantiierungen von X der Fall. Im allgemeinen braucht das aber nicht so zu sein.

und die konjuktive Verknüpfung von Constraints mit „$\sqcup$“ notiert wird, erhält man insgesamt die folgende Ableitung:

$$\cfrac{p(a) \ll (X = Y \sqcup f(b) \prec g(X)),\ f(b) \doteq a \ll (\ldots),\ g(X) \doteq f(b), g(Y) \doteq a, p(f(b)) \vdash p(a)}{\cfrac{f(b) \doteq a \ll (X = Y \sqcup f(b) \prec g(X)),\ g(X) \doteq f(b), g(Y) \doteq a, p(f(b)) \vdash p(a)}{g(X) \doteq f(b), g(Y) \doteq a, p(f(b)) \vdash p(a)} \ \vec{\doteq} l} \ \vec{\doteq} l$$

Der Constraint $X = Y \sqcup f(b) \prec g(X)$ wird etwa von der Grundsubstitution $[X/b, Y/b]$ erfüllt, also ist der Beweis mit dieser Grundsubstitution geschlossen. Man bemerke, wie der Constraint bei der zweiten Anwendung der Gleichheitsregel „vererbt“ wird: da er eine Bedingung für die Gültigkeit der ersten Regelanwendung darstellt, darf die Formel $f(b) \doteq a$ auch nur weiter verwendet werden, wenn diese Bedingung erfüllt ist.

Die präzise Formulierung eines solchen Verfahrens für automatische Beweiser ist nicht einfach. Man möchte ja möglichst viele Gleichheitsanwendungen verbieten, und dabei trotzdem die Vollständigkeit wahren. Insbesondere müssen etwa Vorkehrungen getroffen werden, die eine Anwendung wie die folgende verhindern, in der eine Gleichung „unterhalb einer Variablenposition“ angewandt wird:

$$\frac{f(a) \doteq b \vdash p(f(b)) \ll X = f(f(a)), p(X)}{f(a) \doteq b \vdash p(X)} \ \vec{\doteq} r$$

Müßte der Automat derartige Gleichungsanwendungen berücksichtigen, würde der Suchraum wieder explodieren. Die genaue Einschränkung derjenigen Instanzen der Gleichheitsregeln, die benötigt werden, und der Nachweis der Vollständigkeit des resultierenden Systems sind die schwierigsten theoretischen Aspekte der automatischen Gleichheitsbehandlung. Es wurden jedoch mehrere Verfahren entwickelt, die in ähnlicher Weise Ordnungsconstraints verwenden, wie dies hier (stark vereinfacht) demonstriert wurde.

In dieser Arbeit soll kein eigenes Verfahren hergeleitet werden, sondern gezeigt werden, wie ein existierendes Verfahren an die Gegebenheiten eines Kalküls mit Metavariablen und ϵ-Termen angepaßt werden kann. Es wurde das Verfahren aus [4] verwendet, welches (unter anderem) mit Ordnungsconstraints arbeitet. Der Rest dieses Abschnitts orientiert sich schon deshalb stillschweigend an diesem Verfahren, weil die Gleichheitsbehandlung in Hinblick auf dessen Bedürfnisse entwickelt wurde. Es wird aber darauf geachtet, daß die Ergebnisse für sich selbst verständlich sind, und so wenig wie möglich Eigenheiten dieses Verfahrens wiederspiegeln. Dadurch sollte es möglich sein, auch andere constraintbasierte Verfahren in diesen Kontext einzubetten.

Da das Verfahren von Beckert universelle Variablen verwendet, welche erst im nächsten Kapitel eingeführt werden, wird auf die Entsprechung zwischen jenem Verfahren und dem hier vorgestellten Kalkül erst in 7.5 eingegangen.

6.3. Constraintsprachen

Welche Form Constraints konkret haben, unterscheidet sich natürlich von einem Verfahren zum anderen. In den letzten beiden Abschnitten wurden die zwei wichtigsten Formen vorgestellt:

- Gleichheitsconstraints: Es wird die syntaktische Gleichheit von zwei Termen verlangt, z.B. $X = Y$, $f(X, a) = f(b, Y)$ oder $f(X) = g(Y)$. Das letzte Beispiel ist ein *unerfüllbarer* Constraint, da die beiden Terme unabhängig von der Instantiierung von X und Y verschieden sind.

- Ordnungsconstraints: Es wird verlangt, daß zwei Terme in einer bestimmten Ordnungsbeziehung stehen, etwa $X \prec f(a)$ oder $f(X) \prec g(Y)$. Die Ordnungsrelation ist dabei im allgemeinen eine Reduktionsordnung, die festgelegt wird, bevor das Verfahren gestartet wird.

Diese *atomaren* Constraints müssen für die meisten Verfahren verknüpft werden. Oft genügt die Konjunktion, die im Beispiel des letzten Abschnitts schon verwendet wurde.

Es sind jedoch auch andere Formen von atomaren Constraints denkbar: In [20] und [16] werden z.B. AC-Unifikations-Constraints der Form $s =_{AC} t$ verwendet. Ein solcher Constraint ist erfüllt, wenn die Terme bis auf Assoziativität und Kommutativität bestimmter Operatoren gleich sind. Ebenso sind natürlich andere als nur konjunktive Verknüpfungen zwischen Constraints denkbar.

Im allgemeinen beschreibt ein Constraint eine Menge von Grundsubstitutionen. Diese Grundsubstitutionen *erfüllen* den Constraint. Jeder Formalismus, der es erlaubt, Mengen von Grundsubstitutionen zu beschreiben, kann im Prinzip als Constraintsprache angesehen werden. Um möglichst offen für Erweiterungen zu bleiben, soll die Constraintsprache hier möglichst abstrakt gehalten werden. Es werden also nur bestimmte Forderungen an die Ausdrucksmächtigkeit gestellt. In den Beispielen wird eine minimale Constraintsprache verwendet, die diesen Anforderungen genügt.

An dieser Stelle sollte eine Unterscheidung zwischen zwei Sorten von Constraints eingeführt werden:

- **korrektheits-relevante (k.r.) Constraints**. Dies sind Constraints, die von der abschließenden Instantiierung der Metavariablen erfüllt werden müssen, um einen korrekten Beweis zu gewährleisten. Von dieser Art sind etwa die Gleichheitsconstraints, die in Abschnitt 6.1 eingeführt wurden.

- **korrektheits-irrelevante (k.i.) Constraints**. Dies sind Constraints, die bei der Suche einer abschließenden Substitution nicht berücksichtigt werden *müssen*, um einen korrekten Beweis zu finden. Diese Constraints dienen im allgemeinen der Suchraumbeschränkung für den automatischen Teil des Systems; es werden keine Regeln angewandt, die zu Formeln mit unerfüllbaren Constraints führen, da diese nie für einen Abschluß nötig sind. Die Ordnungsconstraints im Zusammenhang des letzten Abschnitts sind korrektheits-irrelevant.

Diese Aufteilung ist nicht ohne weiteres formal zu fassen: Korrektheit ist eine Eigenschaft von Kalkülen, und die Formulierung des Kalküls verwendet die Constraints in irgendeiner Weise. Man zeigt die Korrektheit des Kalküls inklusive dessen Umgang mit den Constraints. Es gibt keinen kanonischen Weg, um von einem Kalkül zu einem geänderten Kalkül zu kommen, der bestimmte Constraints „nicht berücksichtigt“. Um ein etwas praxisfernes Beispiel zu nennen, wäre ein Kalkül denkbar, der alle Constraints doppelt verwaltet. Dieser bliebe natürlich korrekt, wenn er nur jeweils eine der beiden Kopien betrachtet. Welche der beiden Kopien ist nun k.r.?

Trotzdem ist die Unterscheidung in praktisch auftretenden Fällen meist klar. In unserem Kalkül sollen nur k.r. Constraints explizit betrachtet werden, und zwar aus folgendem Grund: K.i. Constraints dienen der Suchraumbeschränkung. Sie sind also Teil jener Informationen, die der Automat verwendet, um zu entscheiden, wann welche Regel angewandt wird. K.i. Constraints sind *nicht* bindend für den Benutzer: Wenn dieser eine Gleichung in einer anderen Richtung anwenden will, als die Maschine es täte, soll er nicht daran gehindert werden. Wenn der Benutzer dem Automaten vorschreibt, bei der Beweissuche Quantoren nur zweimal zu instantiieren, soll er selber ja auch nicht daran gehindert werden, dies öfter zu tun. Auch soll der Umstand, daß der Automat die *ext*-Regel nie verwenden wird, den Benutzer nicht an deren Verwendung hindern.

Andererseits stellen solche Suchraum-Einschränkungen natürlich relevante Informationen dar, wenn man die Arbeitsweise des Automaten nachvollziehen möchte. Dies spräche dafür, in einem integrierten Kalkül doch auch k.i. Constraints explizit zu verwalten. In diesem Fall müßte der Beweis dem Benutzer unter normalen Umständen so präsentiert werden, daß er die k.i. Constraints nicht zu sehen bekommt, sondern nur, wenn er dies explizit wünscht.

Es ist jedenfalls nicht schwer, die Darstellung in dieser Arbeit so zu erweitern, daß auch k.i. Constraints verwaltet werden. Diese können entweder mit den k.r. Constraints zu einem Constraint zusammengefaßt werden, oder sie werden von den k.r. Constraints separat gehalten, und wie diese an die Formeln einer Sequenz geheftet. Bei der ersteren Wahl könnte allerdings, je nach den Möglichkeiten der Constraintsprache, die Abtrennung der k.i.-Information Probleme bereiten, während bei der letzteren verwandte Informationen getrennt verwaltet werden. Welche Wahl günstiger ist, muß wohl von Verfahren zu Verfahren entschieden werden.

Für unsere Zwecke soll es genügen, die benötigten k.r. Constraints in den Kalkül aufzunehmen, und davon auszugehen, daß benötigte k.i. Constraints davon unabhängig den Formeln einer Sequenz zugeordnet werden können.

Wir wollen nun Constraints formal definieren:

6.1 Definition (Constraintsprache)
Eine **Constraintsprache** *ist eine Menge* $\mathbf{C}$ *von* **Constraints** *mit einer Funktion* $\mathrm{Sat} : \mathbf{C} \to \mathcal{P}(\mathbf{Sub}^0)$. *Eine Grundsubstitution für die Metavariablen* $\sigma \in \mathbf{Sub}^0$ **erfüllt** *einen Constraint* $C \in \mathbf{C}$, *falls* $\sigma \in \mathrm{Sat}(C)$. *Ein Constraint* C *ist* **erfüllbar**, *falls es eine Substitution gibt, die ihn erfüllt, d.h.* $\mathrm{Sat}(C) \neq \emptyset$. *Ein Constraint* C_1 *wird von einem Constraint* C_2 **subsumiert**, *in Zeichen* $C_1 \sqsubseteq C_2$ *falls* $\mathrm{Sat}(C_1) \supseteq \mathrm{Sat}(C_2)$. *Zwei Constraints* C_1, C_2 *heißen* **äquivalent**, *in Zeichen* $C_1 \equiv C_2$, *falls* $\mathrm{Sat}(C_1) = \mathrm{Sat}(C_2)$.

Eine Constraintsprache heißt **zulässig**, *falls sie folgende Eigenschaften hat:*

- *Es existiert ein Constraint* $\bot \in \mathbf{C}$ *mit* $\mathrm{Sat}(\bot) = \mathbf{Sub}^0$.
- *Für alle ganzen Terme* $t, s \in \mathbf{Term}^*$ *existiert uniform ein Constraint* $s = t \in \mathbf{C}$ *mit*
$$\mathrm{Sat}(s = t) = \{\sigma \in \mathbf{Sub}^0 \mid \sigma(s) = \sigma(t)\}.$$
- *Für alle Constraints* $C_1, C_2 \in \mathbf{C}$ *existiert uniform ein Constraint* $C_1 \sqcup C_2 \in \mathbf{C}$ *mit*
$$\mathrm{Sat}(C_1 \sqcup C_2) = \mathrm{Sat}(C_1) \cap \mathrm{Sat}(C_2).$$
- *Die Erfüllbarkeit von Constraints aus* $\mathbf{C}$ *ist entscheidbar.*

Die ersten drei Bedingungen an zulässige Constraints sind Mindestanforderungen an die Ausdrucksmächtigkeit der Constraintsprache. Sie entsprechen gerade den Operationen, die in dieser Arbeit auf Constraints benötigt werden. Man bemerke, daß $\sqcup$ die Rolle einer Konjunktion spielt, und $\bot$ „wahr", also trivial erfüllt ist. Die letzte Bedingung *beschränkt* die Mächtigkeit der Sprache. Sie ergibt sich daraus, daß der Beweisabschluß durch die Erfüllbarkeit eines Constraints ausgedrückt werden wird. Würde man ein rein automatisches Beweissystem anstreben, würde ein rekursiv aufzählbares Erfüllbarkeitsprädikat genügen, da die Logik ohnehin unentscheidbar ist. Für ein interaktives System ist es jedoch wichtig, dem Benutzer anzugeben, ob der derzeitige Beweis geschlossen ist, denn nur so kann dieser eventuelle Fehler in seiner Beweisführung entdecken: Der Automat durchsucht – etwa mittels inkrementeller Tiefensuche – einen Raum von möglichen Beweisen, und kann den Erfüllbarkeitstest nach einer gewissen Zeit abbrechen. Die Vollständigkeit bleibt gewahrt, wenn bei erfolgloser Suche die Suchtiefe hochgesetzt wird, und auch dem Erfüllbarkeitstest mehr Zeit zugestanden wird. Der Mensch verfolgt hingegen einen bestimmten Beweisplan. Wenn ihm nun in der Beweisführung ein Fehler unterläuft, wird er trotzdem auf die Bestätigung des Beweisabschlusses warten. Wenn er weder diese Bestätigung, noch eine negative Antwort erhält, kann er nicht sicher sein, ob er einen Fehler gemacht hat, oder lediglich die Berechnung lange braucht.

In Abschnitt 6.6 wird gezeigt werden, wie Beweise mit Constraints für den Benutzer günstig dargestellt werden können. Dort wird verlangt, daß sogar die *Subsumtionsrelation*, bzw. gleichwertig die *Äquivalenz* entscheidbar ist.[6] Darauf sollte bei der Wahl der Constraintsprache also geachtet werden.

Im folgenden sei $(\mathbf{C}, \mathrm{Sat})$ eine fest gewählte, zulässige Constraintsprache. In den Beispielen werden wir als Constraints nur $\perp$ und (Konjunktionen von) Gleichungen $s = t$ verwenden. Die Entscheidbarkeit der Erfüllbarkeit dieser Sprache folgt aus der Entscheidbarkeit simultaner syntaktischer Unifizierbarkeit.

6.4. Ein Kalkül mit Metavariablen

Da nun ein Kalkül mit Metavariablen definiert werden soll, muß die Definition 5.15 entsprechend von Grundsequenzen auf Sequenzen mit Metavariablen erweitert werden. Außerdem wird jede Formel einer Sequenz mit einem Constraint versehen.

6.2 Definition (Sequenzen)
Eine **Formel mit Constraint** *ist ein als $\phi \ll C$ geschriebenes geordnetes Paar, wobei $\phi \in \mathbf{Form}^*$ und $C \in \mathbf{C}$. Für $\phi \ll \perp$ wird einfach ϕ geschrieben.*

Eine **Sequenz** *ist ein als $\Gamma \vdash \Delta$ geschriebenes geordnetes Paar, wobei Γ und Δ Mengen von Formeln mit Constraints sind. Abkürzende Schreibweisen werden wie für Grundsequenzen (Def. 5.15) verwendet.*

Die Anwendung einer Grundsubstitution $\sigma \in \mathbf{Sub}^0$ auf eine Sequenz $\Gamma \vdash \Delta$ ergibt eine wie folgt definierte Grundsequenz:

$$\sigma(\Gamma \vdash \Delta) := \{\sigma(\gamma) \mid \gamma \ll C \in \Gamma, \sigma \in \mathrm{Sat}(C)\} \vdash \{\sigma(\delta) \mid \delta \ll D \in \Delta, \sigma \in \mathrm{Sat}(D)\}.$$

Bei Anwendung einer Grundsubstitution auf eine Sequenz werden also nur diejenigen Formeln beibehalten, deren Constraint von der Substitution erfüllt wird.

Für die Gleichheitsregeln wird eine bestimmte Sorte von Substitutionen auf den Metavariablen benötigt. Diese Substitutionen entsprechen den Unifikatoren in einem Tableaukalkül mit freien Variablen. Man muß dafür sorgen, daß die Anwendung einer solchen Substitution auf eine ganze Formel wieder eine ganze Formel ergibt. Dies wird von den in Definition 5.3 eingeführten *ganzen* Substitutionen gewährleistet. Das folgende Lemma zeigt, wie aus einer ganzen Substitution μ ein Constraint $\mathrm{Con}(\mu)$ generiert werden kann, der von einer Grundsubstitution σ genau dann erfüllt wird, wenn σ in einem bestimmten Sinne eine Verfeinerung von μ ist.

[6] Es gilt $C_1 \equiv C_2$ gdw. $C_1 \sqsubseteq C_2$ und $C_2 \sqsubseteq C_1$, sowie $C_1 \sqsubseteq C_2$ gdw. $C_1 \sqcup C_2 \equiv C_2$.

6.3 Definition (Con)
Zu $\mu \in \mathbf{Sub}^*$ wird ein Constraint Con(μ) *definiert durch:*

$$\mathrm{Con}(\mu) := \bigsqcup_{X \in \mathrm{dom}(\mu)} X = \mu(X).$$

6.4 Lemma (Substitutionen vs. Constraints)
Für alle $\mu, \nu \in \mathbf{Sub}^*$ und $\sigma \in \mathbf{Sub}^0$ gilt:

- $\sigma \in \mathrm{Sat}(\mathrm{Con}(\mu))$ *gdw.* $\sigma = \sigma \circ \mu$.
- $\mathrm{Con}(\mu) \sqsubseteq \mathrm{Con}(\nu)$ *gdw.* $\nu = \nu \circ \mu$ *gdw. es existiert eine Subst.* τ *mit* $\nu = \tau \circ \mu$.

Beweis. Es gilt $\sigma \in \mathrm{Sat}(\mathrm{Con}(\mu)) \Longleftrightarrow \sigma \in \mathrm{Sat}(X = \mu(X))$ für alle $X \in \mathrm{dom}(\mu)$ $\Longleftrightarrow \sigma(X) = \sigma(\mu(X))$ für alle $X \in \mathrm{dom}(\mu)$ $\Longleftrightarrow \sigma = \sigma \circ \mu$. Für den zweiten Punkt bemerke man, daß für beliebige Terme $t, s \in \mathbf{Term}^*$ gilt: $t = s$ gdw. f. alle $\sigma \in \mathbf{Sub}^0$ gilt: $\sigma(t) = \sigma(s)$. Daraus folgt, daß für beliebige Substitutionen $\mu, \mu' \in \mathbf{Sub}^*$ gilt: $\mu = \mu'$ gdw. f. alle $\sigma \in \mathbf{Sub}^0$ gilt: $\sigma \circ \mu = \sigma \circ \mu'$. Sei nun $\mathrm{Con}(\mu) \sqsubseteq \mathrm{Con}(\nu)$. Für beliebige $\sigma \in \mathbf{Sub}^0$ ist $\sigma \circ \nu = \sigma \circ \nu \circ \nu$, da ν idempotent ist[7] $\Longrightarrow \sigma \circ \nu \in \mathrm{Sat}(\mathrm{Con}(\nu))$ $\Longrightarrow \sigma \circ \nu \in \mathrm{Sat}(\mathrm{Con}(\mu))$ $\Longrightarrow \sigma \circ \nu = \sigma \circ \nu \circ \mu$, also aufgrund der Vorüberlegung $\nu = \nu \circ \mu$. Die dritte Aussage ist triviale Folge der zweiten. Für den Schritt von der dritten zur ersten Aussage sei $\nu = \tau \circ \mu$ und $\sigma \in \mathrm{Sat}(\mathrm{Con}(\nu))$, also $\sigma = \sigma \circ \nu$. Dann ist $\sigma = \sigma \circ \nu = \sigma \circ \tau \circ \mu = \sigma \circ \tau \circ \mu \circ \mu = \sigma \circ \nu \circ \mu = \sigma \circ \mu \Longrightarrow \sigma \in \mathrm{Sat}(\mathrm{Con}(\mu))$. ■

In Abbildung 6.1 sind die Beweisregeln für den Kalkül $\mathcal{M}_E$ angegeben. Deren Verwendung regelt folgende Definition

6.5 Definition ($\mathcal{M}_E$-Beweis)
Ein $\mathcal{M}_E$-**Beweis** *für eine geschlossene Formel $\phi \in \mathbf{Form}^0$ ist ein Baum, dessen Knoten mit Sequenzen markiert sind, so daß*

- *der Wurzelknoten mit der Sequenz $\vdash \phi \ll \bot$ markiert ist, und*
- *für jeden mit Σ markierten inneren Knoten, dessen Nachkommen mit $\Sigma_1, \ldots, \Sigma_k$ markiert sind,*

$$\frac{\Sigma_1 \quad \ldots \quad \Sigma_k}{\Sigma}$$

  Instanz einer der Regeln von $\mathcal{M}_E$ ist.

Ein $\mathcal{M}_E$-Beweis heißt **geschlossen**, *falls er endlich ist, und eine Grundsubstitution $\sigma \in \mathbf{Sub}^0$ existiert, so daß $\sigma(\Gamma \vdash \Delta)$ für jedes Blatt $\Gamma \vdash \Delta$ ein $\mathcal{G}_E$-Axiom (sic) ist.*

Da die Anwendung von σ eine Grundsequenz erzeugt, können hier die Axiome von $\mathcal{G}_E$ verwendet werden.

Der Beweisabschluß läßt sich auch durch eine (Kombination von) Erfüllbarkeitsaussage(n) für Constraints charakterisieren:

[7] Ganze Substitutionen sind laut Definition 5.3 idempotent

Abbildung 6.1.: Beweisregeln des Kalküls $\mathcal{M}_E$

$$\frac{\phi \ll C, \psi \ll C, \Gamma \vdash \Delta}{\phi \wedge \psi \ll C, \Gamma \vdash \Delta}\ \wedge l \qquad \frac{\Gamma \vdash \phi \ll C, \Delta \quad \Gamma \vdash \psi \ll C, \Delta}{\Gamma \vdash \phi \wedge \psi \ll C, \Delta}\ \wedge r$$

$$\frac{\Gamma \vdash \phi \ll C, \Delta}{\neg\phi \ll C, \Gamma \vdash \Delta}\ \neg l \qquad \frac{\phi \ll C, \Gamma \vdash \Delta}{\Gamma \vdash \neg\phi \ll C, \Delta}\ \neg r$$

$$\frac{[x/\epsilon x.\phi](\phi) \ll C, \Gamma \vdash \Delta}{\exists x.\phi \ll C, \Gamma \vdash \Delta}\ \exists l \qquad \frac{\Gamma \vdash [x/X](\phi) \ll C, \exists x.\phi \ll C, \Delta}{\Gamma \vdash \exists x.\phi \ll C, \Delta}\ \exists r$$

wobei $X \in \mathbf{M}$ im Beweis neu ist.

$$\frac{[y/\mu(t)](\phi') \ll C', s \doteq t \ll C_1, \phi \ll C_2, \Gamma \vdash \Delta}{s \doteq t \ll C_1, \phi \ll C_2, \Gamma \vdash \Delta}\ \overset{\rightarrow}{\equiv} l$$

wobei $y \in \mathbf{V}$, $\mu \in \mathbf{Sub}^*$, $\mu(\phi) = [y/\mu(s)](\phi')$, und $C' := C_1 \sqcup C_2 \sqcup \mathrm{Con}(\mu)$.

$\overset{\leftarrow}{\equiv} l$, $\overset{\rightarrow}{\equiv} r$ und $\overset{\leftarrow}{\equiv} r$ entsprechend.

$$\frac{\Gamma \vdash \exists x.\phi, \Delta \quad [x/\epsilon x.\phi](\phi), \Gamma \vdash \Delta}{\Gamma \vdash \Delta}\ \epsilon\textit{-cut}$$

wobei $\epsilon x.\phi \in \mathbf{Term}^*$.

$$\frac{\Gamma \vdash [x/\mathbf{c}](\phi) \leftrightarrow [y/\mathbf{c}](\psi), \Delta \quad \epsilon x.\phi \doteq \epsilon y.\psi, \Gamma \vdash \Delta}{\Gamma \vdash \Delta}\ \textit{ext}$$

wobei $\epsilon x.\phi$, $\epsilon y.\psi \in \mathbf{Term}^*$, $\mathbf{c} := \epsilon z.\neg\Big([x/z](\phi) \leftrightarrow [y/z](\psi)\Big)$, $z \in \mathbf{V} \setminus (\mathrm{bv}(\phi) \cup \mathrm{bv}(\psi))$.

$$\frac{\Gamma \vdash \phi, \Delta \quad \Gamma, \phi \vdash \Delta}{\Gamma \vdash \Delta}\ \textit{cut}$$

wobei $\phi \in \mathbf{Form}^*$.

$$\frac{\Gamma \vdash \Delta}{\phi \ll C, \Gamma \vdash \Delta}\ \textit{weak-l} \qquad \frac{\Gamma \vdash \Delta}{\Gamma \vdash \phi \ll C, \Delta}\ \textit{weak-r}$$

6.6 Definition (abschließender Constraint)
Sei $\Sigma = \Gamma \vdash \Delta$ *eine Sequenz.* C *heißt ein* **abschließender Constraint** *für* $\Gamma \vdash \Delta$*, falls* $p(s_1, \ldots, s_n) \ll C_1 \in \Gamma$*,* $p(t_1, \ldots, t_n) \ll C_2 \in \Delta$ *und*

$$C = C_1 \sqcup C_2 \sqcup \bigsqcup_{i=1}^{n} (s_i = t_i),$$

oder $s \doteq t \ll C_1 \in \Delta$ *und*

$$C = C_1 \sqcup (s = t).$$

Seien Σ_i*,* $i = 1, \ldots, k$ *alle Blätter eines endlichen* $\mathcal{M}_E$*-Beweises und* C_i *abschließende Constraints für* Σ_i *für alle* i*. Dann heißt*

$$C := \bigsqcup_{i=1}^{k} C_i$$

ein **abschließender Constraint** *des Beweises.*[8]

6.7 Satz (Abschluß via Constraints)
Ein endlicher $\mathcal{M}_E$*-Beweis ist genau dann geschlossen, wenn für ihn ein erfüllbarer abschließender Constraint existiert.*

Beweis. Triviale Folgerung aus der Definition der Axiome und der Grundsubstitution auf Sequenzen. ■

An dieser Stelle ist ein Wort zur Herkunft dieser Regeln angebracht. Sie entstanden im wesentlichen durch Einführung von Metavariablen in den Grundkalkül $\mathcal{G}_E$, und zwar derart, daß durch Anwendung der abschließenden Grundsubstitution ein gültiger Grund-Beweis entsteht. Wie in Abschnitt 4.4 erläutert, ist dieses Verhalten für die Beweispräsentation vorteilhaft.

Die ersten sechs sind Anpassungen der Regeln von $\mathcal{G}_E$ auf einen Kalkül mit Constraints und Metavariablen. Bei ihrer Anwendung werden Constraints einfach von den unteren an die oberen Formeln „vererbt". Die Regeln *ϵ-cut* und *ext* lassen nun *ganze* Terme zu, und nicht nur Grundterme, da diese unter der abschließenden Grundsubstitution zu Grundtermen werden.

Die Regeln *weak-l*, *weak-r* und *cut* sind gegenüber $\mathcal{G}_E$ neu dazugekommen. Die beiden als Abschwächungsregeln bekannten *weak*-Regeln erlauben es, eine nicht mehr benötigte Formel aus der Sequenz zu entfernen. Diese Regeln waren im ursprünglichen Sequenzenkalkül von Gentzen nötig, da dort nur Sequenzen $\phi \vdash \phi$ als Axiome verwendet wurden. In unserem Kontext sind sie wichtig, da sie es ermöglichen, nicht mehr benötigte Formeln aus einer Sequenz zu entfernen, was bei interaktiver Verwendung sehr zur Übersichtlichkeit beitragen kann. Diese Regeln wurden in den Beispielen schon mehrfach implizit verwendet; ihre Korrektheit, die im nächsten Abschnitt bewiesen wird, rechtfertigt diese Praxis nachträglich.

Die Regel *cut* ist bekanntlich auch nicht für einen vollständigen Kalkül nötig, durch ihre Verwendung können jedoch für gewisse Formelklassen nicht-elementar kürzere Beweise gefunden werden. Wie die *ϵ-cut* und *ext*-Regeln führt die *cut* Regel Formeln mit

[8] Falls für ein Blatt kein abschließender Constraint existiert, so hat der Beweis auch keinen abschließenden Constraint.

dem schwächsten Constraint $\perp$ ein. Dies ist sinnvoll, da jeder Abschluß, der sich mit einem stärkeren Constraint finden ließe, erst recht mit $\perp$ möglich ist.

Die Gleichheitsregeln sind gegenüber $\mathcal{G}_E$ komplizierter geworden. Dies hängt damit zusammen, daß *zwei* Formeln im Spiel sind, die mit unterschiedlichen Constraints versehen sein können. Die Gleichheitsanwendung muß möglich sein, falls sie nach Anwendung der abschließenden Grundsubstitution im Grundkalkül möglich wäre. Für die Regel $\overset{\rightarrow}{=} l$ muß also gelten: $\sigma(\phi) = [y/\sigma(s)](\phi')$ für eine Formel $\phi' \in \mathbf{Form}^0$. Daraus erzeugt die Regel des Grundkalküls die neue Formel $[y/\sigma(t)](\phi')$. Speziell der Automat sollte sich jedoch bei einer Anwendung der Gleichheitsregeln noch nicht auf eine Grundsubstitution festlegen müssen. Daher wird die Regel mit der ganzen Substitution μ formuliert, die dem Unifikator in den Gleichheitsregeln automatischer Beweiser entspricht. Der hinzukommende Constraint $\mathrm{Con}(\mu)$ stellt gerade sicher, daß die Regel unter der abschließenden Substitution σ anwendbar ist.

Orientiert man sich an den Gleichheitsregeln in anderen constraintbasierten Verfahren zur automatischen Gleichheitsbehandlung, so kommt man zu einer Regel der Art

$$\frac{[y/\mu(t)](\mu(\phi)) \ll C', s \doteq t \ll C_1, [y/s']\phi \ll C_2, \Gamma \vdash \Delta}{s \doteq t \ll C_1, [y/s']\phi \ll C_2, \Gamma \vdash \Delta} \doteq'$$

wobei $\mu \in \mathbf{Sub}^*$, $\mu(s) = \mu(s')$ und $C' = C_1 \sqcup C_2 \sqcup \mathrm{Con}(\mu)$. Man sieht leicht, daß die Regel $\overset{\rightarrow}{=} l$ stärker als $\doteq'$ ist. Wenn $\doteq'$ anwendbar ist, gilt nämlich $\mu([y/s']\phi) = [y/\mu(s')](\mu(\phi)) = [y/\mu(s)]\phi'$, wenn $\phi' := \mu(\phi)$ gewählt wird. Damit ist $\overset{\rightarrow}{=} l$ anwendbar, und da $[y/\mu(t)](\phi') = [y/\mu(t)](\mu(\phi))$, erzeugen die beiden Regeln dieselben Sequenzen.

Für den interaktiven Gebrauch ist die Regel $\doteq'$ jedoch nicht stark genug. Sie erlaubt zum Beispiel nicht den folgenden Schluß:

$$\frac{p(g(f(a))) \ll X = g(a), a \doteq f(a), p(X) \vdash}{a \doteq f(a), p(X) \vdash} \overset{\rightarrow}{=} l$$

Wenn der Benutzer aber für die Metavariable X die Belegung $g(a)$ angegeben hat, möchte er die Regel so anwenden können. Schließlich steht in seinen Augen in der unteren Sequenz statt $p(X)$ die Formel $p(g(a))$, auf die die Gleichung natürlich anwendbar ist.

Die Gleichung wird hier, wie man sagt, „unterhalb einer Variablenposition“ angewandt. Automatische Verfahren verbieten derartige Anwendungen, da sie sehr indeterministisch sind. Es könnte ja für X jeder beliebige Term eingesetzt werden, der a enthält. Der automatische Teil eines integrierten Beweissystems, welches die Gleichheitsregeln des Kalküls $\mathcal{M}_E$ zur Verfügung stellt, wird diese also nur in eingeschränkter Weise anwenden.

Die Gleichheitsregeln von $\mathcal{M}_E$ sind auf den ersten Blick erheblich komplizierter als die Regeln in Grundkalkülen, die üblicherweise für interaktive Beweissysteme verwendet werden. Es wurde in den vorangegangenen Absätzen versucht, die Notwendigkeit dieser zusätzlichen Komplexität zu belegen. In Abschnitt 6.6 wird gezeigt werden, wie die Gleichheitsregeln für den menschlichen Benutzer handhabbar gemacht werden können, solange dieser den Kalkül wie einen Grundkalkül verwendet. Will er jedoch die Möglichkeiten von Constraints und Metavariablen ausschöpfen, wird er sich auch mit der erhöhten Komplexität der Gleichheitsregeln abfinden müssen.

6.4.1. Korrektheit

Es soll nun die Korrektheit des Kalküls $\mathcal{M}_E$ bezüglich der extensionalen Semantik gezeigt werden. Dabei wird die Extensionalität nur für die Regel *ext* benötigt, woraus die

Korrektheit des Kalküls *ohne* diese Regel für die substitutive Semantik folgt. Diese Tatsache ist von Nutzen, wenn der Kalkül nicht nur – mit Hilfe des Einbettungssatzes – zum Beweisen prädikatenlogischer Formeln, sondern auch für Aussagen mit ϵ-Termen verwendet werden soll, deren Gültigkeit von der gewählten Semantik abhängt. Will man also die S-Gültigkeit einer Formel nachweisen, so kann dies geschehen, indem man auf die *ext*-Regel verzichtet.

6.8 Satz (Korrektheit von $\mathcal{M}_E$)
Falls für eine geschlossene Formel $\phi \in \mathbf{Form}^0$ ein geschlossener $\mathcal{M}_E$-Beweis existiert, ist ϕ E-allgemeingültig.

Beweis. Es ist für den Beweis nützlich, die Gültigkeitsrelation auf Grundsequenzen fortzusetzen. Für eine Struktur $\mathcal{S}$ und eine Grundsequenz $\Gamma \vdash \Delta$ sei

$$\mathcal{S} \models (\Gamma \vdash \Delta) \qquad \text{gdw.} \qquad \mathcal{S} \not\models \gamma \text{ für ein } \gamma \in \Gamma, \text{ oder } \mathcal{S} \models \delta \text{ für ein } \delta \in \Delta.$$

Eine Grundsequenz $\Gamma \vdash \Delta$ heißt E-allgemeingültig, falls $\mathcal{S} \models (\Gamma \vdash \Delta)$ für alle E-Strukturen $\mathcal{S}$. Man sieht leicht, daß für eine geschlossene Formel $\phi \in \mathbf{Form}^0$ gilt: ϕ ist genau dann E-allgemeingültig, wenn $\vdash \phi$ E-allgemeingültig ist. Wir zeigen nun für beliebige Sequenzen $\Gamma \vdash \Delta$:

> Wenn ein $\mathcal{M}_E$-Beweis mit Wurzel $\Gamma \vdash \Delta$ existiert, der mit $\sigma \in \mathbf{Sub}^0$ geschlossen ist, dann ist $\sigma(\Gamma \vdash \Delta)$ E-allgemeingültig.

Da ϕ keine Metavariablen enthält, ist $\sigma(\vdash \phi \ll \bot) = (\vdash \phi)$, woraus die zu beweisende Korrektheit folgt.

Da geschlossene Beweise *per definitionem* endlich sind, kann über die Baumhöhe induziert werden. Für den Induktionsanfang muß gezeigt werden, daß $\sigma(\Sigma)$ allgemeingültig ist, falls $\sigma(\Sigma)$ ein $\mathcal{G}_E$-Axiom ist, also kurz, daß alle $\mathcal{G}_E$-Axiome allgemeingültig sind. Im Induktionsschritt betrachtet man die erste (unterste) Regelanwendung

$$\frac{\Sigma_1 \quad \ldots \quad \Sigma_k}{\Sigma}$$

des Beweises. Offenbar existieren für alle Σ_i Beweise, die mit σ geschlossen sind. Laut Induktionsannahme sind daher alle $\sigma(\Sigma_i)$ allgemeingültig. Insgesamt muß also noch gezeigt werden:

1. Alle $\mathcal{G}_E$-Axiome sind allgemeingültig.
2. Für jede Regel und jede Substitution $\sigma \in \mathbf{Sub}^0$ gilt: Falls $\sigma(\Sigma_i)$ für alle Prämissen Σ_i der Regel allgemeingültig ist, so ist auch $\sigma(\Sigma)$ allgemeingültig.

Zu 1. Die Grundsequenz $\Gamma \vdash \Delta$ sei $\mathcal{G}_E$-Axiom. Es können zwei Fälle auftreten:

- $p(t_1, \ldots, t_n) \in \Gamma \cap \Delta$ für eine atomare Formel $p(t_1, \ldots, t_n)$. Für jede Struktur gilt entweder $\mathcal{S} \not\models p(t_1, \ldots, t_n)$, oder $\mathcal{S} \models p(t_1, \ldots, t_n)$, also $\mathcal{S} \models (\Gamma \vdash \Delta)$.
- $s \doteq s \in \Gamma$ für einen Grundterm $s \in \mathbf{Term}^0$. Für jede Struktur $\mathcal{S}$ gilt $\mathcal{S} \models s \doteq s$, also auch $\mathcal{S} \models (\Gamma \vdash \Delta)$.

Zu 2. Die Aussage soll für jede Regel separat gezeigt werden.

$\wedge l$ Sei $\sigma(\phi \ll C, \psi \ll C, \Gamma \vdash \Delta)$ allgemeingültig. Gemäß der Definition der Gültigkeit von Grundsequenzen, sowie der Anwendung von Substitutionen auf Sequenzen, können für eine beliebige extensionale Struktur $\mathcal{S} = (\mathcal{D}, \mathcal{I}, \mathcal{A})$ folgende Fälle eintreten:

- $\mathcal{S} \not\models \sigma(\gamma)$ für ein $\gamma \ll D \in \Gamma$ mit $\sigma \in \mathrm{Sat}(D)$. Dann gilt $\mathcal{S} \models \sigma(\phi \wedge \psi \ll D, \Gamma \vdash \Delta)$, da $\gamma \ll D$ auch im Antezedenten der unteren Sequenz steht.
- $\mathcal{S} \models \sigma(\delta)$ für ein $\delta \ll D \in \Delta$ mit $\sigma \in \mathrm{Sat}(D)$. Dann gilt $\mathcal{S} \models \sigma(\phi \wedge \psi \ll D, \Gamma \vdash \Delta)$ wie im ersten Fall. Diese beiden trivialen Fälle sollen in Zukunft nicht mehr erwähnt werden. Als „interessant“ werden die Fälle bezeichnet, in denen $\mathcal{S} \models \sigma(\gamma)$ für alle $\gamma \ll D$ mit $\sigma \in \mathrm{Sat}(D)$ und $\mathcal{S} \not\models \sigma(\delta)$ für alle $\delta \ll D \in \Delta$ mit $\sigma \in \mathrm{Sat}(D)$ gelten.
- $\mathcal{S} \not\models \sigma(\phi)$, und $\sigma \in \mathrm{Sat}(C)$. Dann gilt $\mathcal{S} \not\models \sigma(\phi \wedge \psi)$, also abermals $\mathcal{S} \models \sigma(\phi \wedge \psi \ll C, \Gamma \vdash \Delta)$.
- $\mathcal{S} \not\models \sigma(\psi)$, und $\sigma \in \mathrm{Sat}(C)$. Dieser Fall verläuft wie der vorige.

$\wedge r$ Seien $\sigma(\Gamma \vdash \phi \ll C, \Delta)$ und $\sigma(\Gamma \vdash \psi \ll C, \Delta)$ allgemeingültig. Interessant ist nur der Fall, daß $\mathcal{S} \models \sigma(\phi)$, $\mathcal{S} \models \sigma(\psi)$ und $\sigma \in \mathrm{Sat}(C)$. Dann gilt offenbar $\mathcal{S} \models \sigma(\phi \wedge \psi)$, also $\mathcal{S} \models \sigma(\Gamma \vdash \phi \wedge \psi \ll C, \Delta)$.

$\neg l$, $\neg r$ Die Beweise ähneln den beiden vorigen.

$\exists l$ Es gelte $\mathcal{S} \not\models \sigma([x/\epsilon x.\phi]\phi)$, also durch Hineinziehen von σ: $\mathcal{S} \not\models [x/\epsilon x.\sigma(\phi)](\sigma(\phi))$. (Man bemerke, daß σ nur auf Metavariablen wirkt, x also nicht verändert.) Die Substitutivität liefert: $\mathcal{S}, \{x \leftarrow \mathcal{A}(\epsilon x.\sigma(\phi))\} \not\models \sigma(\phi)$, und mit der Auswahleigenschaft: $\mathcal{S} \not\models \exists x.\sigma(\phi)$. Nun kann σ wieder nach außen gezogen werden, und man erhält $\mathcal{S} \not\models \sigma(\exists x.\phi)$, woraus wie bisher die Gültigkeit der unteren Sequenz unter σ folgt.

$\exists r$ Es gelte $\mathcal{S} \models \sigma([x/X]\phi)$, und $\sigma \in \mathrm{Sat}(C)$. Setzt man die Instantiierung für X ein, erhält man $\mathcal{S} \models [x/\sigma(X)](\sigma(\phi))$, und mit der Substitutivität $\mathcal{S}, \{x \leftarrow \mathrm{val}(\mathcal{S}, \sigma(X))\} \models \sigma(\phi)$. Sicher gilt also $\mathcal{S} \models \exists x.\sigma(\phi)$, und damit $\mathcal{S} \models \sigma(\exists x.\phi)$.

$\stackrel{\rightarrow}{=} l$ Im interessanten Fall gilt hier $\mathcal{S} \not\models \sigma([y/\mu(t)]\phi)$ und $\sigma \in \mathrm{Sat}(C')$. Es gilt dann $\sigma \in \mathrm{Sat}(C_1)$ und $\sigma \in \mathrm{Sat}(C_2)$. Falls $\mathcal{S} \not\models \sigma(s \doteq t)$ oder $\mathcal{S} \not\models \sigma(\phi)$, ist die Aussage gezeigt. Angenommen $\mathcal{S} \models \sigma(s \doteq t)$ und $\mathcal{S} \models \sigma(\phi)$. Weil $\sigma \in \mathrm{Sat}(\mathrm{Con}(\mu))$, gilt laut Lemma 6.4 $\sigma = \sigma \circ \mu$. Es ist also $\sigma(\phi) = \sigma(\mu(\phi)) = \sigma([y/\mu(s)]\phi') = [y/\sigma(s)]\sigma(\phi')$. Wegen der Substitutivität gilt also $\mathcal{S}, \{y \leftarrow \mathrm{val}(\mathcal{S}, \sigma(s))\} \models \sigma(\phi')$. Da $\mathcal{S} \models \sigma(s) \doteq \sigma(t)$, gilt $\mathcal{S}, \{y \leftarrow \mathrm{val}(\mathcal{S}, \sigma(t))\} \models \sigma(\phi')$, also $\mathcal{S} \models [y/\sigma(t)]\sigma(\phi')$. Da $[y/\sigma(t)]\sigma(\phi') = [y/\sigma(\mu(t))]\sigma(\phi') = \sigma([y/\mu(t)](\phi'))$, folgt $\mathcal{S} \models \sigma([y/\mu(t)](\phi'))$, was ein Widerspruch ist.

$\stackrel{\rightarrow}{=} r$, $\stackrel{\leftarrow}{=} l$, $\stackrel{\leftarrow}{=} r$ Der Beweis entspricht dem für $\stackrel{\rightarrow}{=} l$.

ϵ-cut Hier gelten im interessanten Fall $\mathcal{S} \models \sigma(\exists x.\phi)$ und $\mathcal{S} \not\models \sigma([x/\epsilon x.\phi](\phi))$. Es gilt $\mathcal{S} \models \sigma(\exists x.\phi) \implies \mathcal{S} \models \exists x.\sigma(\phi) \stackrel{\text{Ausw.-Eig.}}{\implies} \mathcal{S}, \{x \leftarrow \mathcal{A}(\epsilon x.\sigma(\phi))\} \models \sigma(\phi) \stackrel{\text{Subst.}}{\implies} \mathcal{S} \models [x/\epsilon x.\sigma(\phi)](\sigma(\phi)) \implies \mathcal{S} \models \sigma([x/\epsilon x.\phi](\phi))$, dieser Fall kann also nicht eintreten.

ext Im interessanten Fall gelten $\mathcal{S} \models \sigma([x/\mathbf{c}](\phi) \leftrightarrow [y/\mathbf{c}](\psi))$ und $\mathcal{S} \not\models \sigma(\epsilon x.\phi \doteq \epsilon y.\psi)$. Hineinziehen von σ ergibt: $\mathcal{S} \models [x/\mathbf{c}'](\sigma(\phi)) \leftrightarrow [y/\mathbf{c}'](\sigma(\psi))$ mit $\mathbf{c}' :=$

$\sigma(\mathbf{c}) = \epsilon z.\neg([x/z](\sigma(\phi)) \leftrightarrow [y/z](\sigma(\psi)))$. Laut Auswahleigenschaft gilt damit $\mathcal{S} \models \neg\exists z.\neg([x/z](\sigma(\phi)) \leftrightarrow [y/z](\sigma(\psi)))$. Daraus folgt, daß für alle Trägerelemente d gilt: $\mathcal{S}, \{x \leftarrow d\} \models \sigma(\phi)$ gdw. $\mathcal{S}, \{y \leftarrow d\} \models \sigma(\psi)$. Da $\mathcal{S}$ extensional ist, folgt $\mathrm{val}(\mathcal{S}, \epsilon x.\sigma(\phi)) = \mathrm{val}(\mathcal{S}, \epsilon y.\sigma(\psi))$, also $\mathcal{S} \models \epsilon x.\sigma(\phi) = \epsilon y.\sigma(\psi)$. Da σ nur Metavariablen verändert, folgt daraus $\mathcal{S} \models \sigma(\epsilon x.\phi \doteq \epsilon y.\psi)$, womit dieser Fall ausgeschlossen ist.

cut Der interessante Fall kann nicht eintreten, da dann $\mathcal{S} \models \sigma(\phi)$ und $\mathcal{S} \not\models \sigma(\phi)$ gelten müßten.

weak-l, weak-r Trivial: Es gibt gar keinen interessanten Fall.

■

6.4.2. Vollständigkeit

Die Vollständigkeit des Kalküls $\mathcal{M}_E$ wird durch „lifting“ gezeigt, das heißt, es wird aus einem $\mathcal{G}_E$ Beweis ein $\mathcal{M}_E$ Beweis für dieselbe Formel konstruiert, und die Vollständigkeit von $\mathcal{G}_E$ verwendet. Am Ende dieses Abschnitts wird auf die Vollständigkeit eines $\mathcal{G}_S$ entsprechenden eingeschränkten Kalküls eingegangen, der für einen automatischen Beweiser geeignet ist.

6.9 Satz (Vollständigkeit von $\mathcal{M}_E$)
Für jede E-allgemeingültige geschlossene Formel $\phi \in \mathbf{Form}^0$ existiert ein geschlossener $\mathcal{M}_E$-Beweis.

Beweis. Aufgrund der Vollständigkeit des Kalküls $\mathcal{G}_E$ existiert für ϕ ein geschlossener $\mathcal{G}_E$-Beweis B_0. Sei K die Menge der Knoten von B_0, auf die die $\exists r$-Regel angewandt wird und $t_k \in \mathbf{Term}^0$ die im Knoten $k \in K$ gewählte Instantiierung. Ferner sei eine injektive Abbildung $K \longrightarrow \mathbf{M}$, $k \longmapsto X_k$ gegeben. Nun wird eine Grundsubstitution $\sigma \in \mathbf{Sub}^0$ gewählt mit $\sigma(X_k) := t_k$ für alle $k \in K$.[9] Es soll nun aus B_0 ein $\mathcal{M}_E$-Beweis B mit folgenden Eigenschaften konstruiert werden:

- Die Wurzel von B ist mit $\vdash \phi \ll \bot$ markiert
- B hat dieselbe Baumstruktur wie B_0.
- Falls ein Knoten in B mit einer Sequenz $\Gamma \vdash \Delta$ markiert ist, dann ist der entsprechende Knoten in B_0 mit $\sigma(\Gamma \vdash \Delta)$ belegt.

Da B_0 geschlossen ist, sind alle Blätter von B_0 mit $\mathcal{G}_E$-Axiomen markiert. Demnach sind alle Blätter von B mit Sequenzen $\Gamma \vdash \Delta$ markiert, für die $\sigma(\Gamma \vdash \Delta)$ Axiome sind. Also ist B definitionsgemäß mit σ geschlossen, und die Aussage ist gezeigt.

Die Konstruktion erfolgt induktiv, beginnend mit der Wurzel. Hier gilt sicherlich $\sigma(\vdash \phi \ll \bot) = (\vdash \phi)$. Für jeden Beweisschritt

$$\frac{\Sigma_0^1 \quad \dots \quad \Sigma_0^k}{\Sigma_0} \; R_0$$

[9] Der zu konstruierende $\mathcal{M}_E$-Beweis B wird nur die Metavariablen X_k verwenden, daher ist der Wert von σ auf anderen Metavariablen gleichgültig.

in B_0 und jede Sequenz Σ mit $\sigma(\Sigma) = \Sigma_0$ wird nun eine $\mathcal{M}_E$-Regelanwendung

$$\frac{\Sigma^1 \quad \dots \quad \Sigma^k}{\Sigma}\ R$$

in B angegeben mit $\sigma(\Sigma^i) = \Sigma_0^i$ für $i = 1, \dots, k$. Dies geschieht durch Fallunterscheidung nach der $\mathcal{G}_E$-Regel R_0.

$\wedge l$ Es ist $\Sigma_0 = (\phi_0 \wedge \psi_0, \Gamma_0 \vdash \Delta_0)$ und $\Sigma_0^1 = (\phi_0, \psi_0, \Gamma_0 \vdash \Delta_0)$. Daher muß $\Sigma = (\phi \wedge \psi \ll C, \Gamma \vdash \Delta)$ sein mit $\sigma \in \mathrm{Sat}(C)$, $\phi_0 = \sigma(\phi)$, $\psi_0 = \sigma(\psi)$ und $(\Gamma_0 \vdash \Delta_0) = \sigma(\Gamma \vdash \Delta)$. Dann kann aber die $\mathcal{M}_E$-Regel $\wedge l$ auf Σ angewandt werden, was zu $\Sigma^1 = (\phi \ll C, \psi \ll C, \Gamma \vdash \Delta)$ führt. Offenbar gilt $\sigma(\Sigma^1) = \Sigma_0^1$. Die Korrespondenz zwischen den mit $_0$ indizierten Symbolen in B_0 und denen in B, sowie die Bedingung $\sigma \in \mathrm{Sat}(C)$ soll im Rest des Beweises nicht mehr explizit ausgeführt werden. Man mache sich jeweils klar, daß Σ tatsächlich die angegebene Form haben muß, wenn $\sigma(\Sigma) = \Sigma_0$.

$\wedge r$ $\Sigma_0 = (\Gamma_0 \vdash \phi_0 \wedge \psi_0, \Delta_0)$, $\Sigma_0^1 = (\Gamma_0 \vdash \phi_0, \Delta_0)$, $\Sigma_0^2 = (\Gamma_0 \vdash \psi_0, \Delta_0) \implies \Sigma = (\Gamma \vdash \phi \wedge \psi \ll C, \Delta)$. Anwendung von $\wedge r$ auf Σ ergibt $\Sigma^1 = (\Gamma \vdash \phi \ll C, \Delta)$ und $\Sigma^2 = (\Gamma \vdash \psi \ll C, \Delta)$. Offensichtlich gilt $\sigma(\Sigma^i) = \Sigma_0^i$.

$\neg l$, $\neg r$ Ähnlich den beiden vorigen.

$\exists l$ $\Sigma_0 = (\exists x.\phi_0, \Gamma_0 \vdash \Delta_0)$, $\Sigma_0^1 = ([x/\epsilon x.\phi_0](\phi_0), \Gamma_0 \vdash \Delta_0) \implies \Sigma = (\exists x.\phi \ll C, \Gamma \vdash \Delta)$, da x nicht von σ verändert wird. Anw. v. $\exists l$ auf Σ erg. $\Sigma^1 = ([x/\epsilon x.\phi](\phi) \ll C, \Gamma \vdash \Delta)$. Es ist $\sigma([x/\epsilon x.\phi](\phi)) = [x/\epsilon x.\sigma(\phi)](\sigma(\phi)) = [x/\epsilon x.\phi_0](\phi_0)$, was diesen Fall abschließt.

$\exists r$ $\Sigma_0 = (\Gamma_0 \vdash \exists x.\phi_0, \Delta_0)$, $\Sigma_0^1 = (\Gamma_0 \vdash [x/t_k](\phi_0), \exists x.\phi_0, \Delta_0)$, wobei $k \in K$ der mit Σ_0 markierte Knoten ist $\implies \Sigma = (\Gamma \vdash \exists x.\phi \ll C, \Delta)$. Anwendung von $\exists r$ unter Verwendung der (zwangsläufig neuen!) Metavariablen X_k ergibt $\Sigma^1 = (\Gamma \vdash [x/X_k](\phi) \ll C, \exists x.\phi \ll C, \Delta)$. Es gilt $\sigma([x/X_k](\phi)) = [x/\sigma(X_k)](\sigma(\phi)) = [x/t_k](\phi_0)$.

$\overset{\rightrightarrows}{=} l$ $\Sigma_0 = (s_0 \doteq t_0, [y/s_0](\phi_0), \Gamma_0 \vdash \Delta_0)$, $\Sigma_0^1 = ([y/t_0](\phi_0), s_0 \doteq t_0, [y/s_0](\phi_0)\Gamma_0 \vdash \Delta_0)$. Hier ist die Bestimmung von Σ etwas schwieriger. Es muß $\Sigma = (s \doteq t \ll C_1, \psi \ll C_2, \Gamma \vdash \Delta)$ sein, wobei $\sigma(\psi) = [y/s_0](\phi_0)$ und $\sigma \in \mathrm{Sat}(C_1) \cap \mathrm{Sat}(C_2)$. Um die $\overset{\rightrightarrows}{=} l$-Regel von $\mathcal{M}_E$ anwenden zu können, nimmt man $\mu := \sigma$. Da σ Grundsubstitution ist, gilt $\sigma = \sigma \circ \sigma$, also $\sigma \in \mathrm{Sat}(\mathrm{Con}(\sigma))$, und insgesamt $\sigma \in \mathrm{Sat}(C')$. Ferner ist $\mu(\psi) = \sigma(\psi) = [y/s_0](\phi_0) = [y/\sigma(s)](\phi_0)$. Es übernimmt also ϕ_0 die Rolle von ψ'. Damit ergibt die Regelanwendung $\Sigma^1 = ([y/\sigma(t)](\phi_0) \ll C', s \doteq t \ll C_1, \psi \ll C_2, \Gamma \vdash \Delta)$, womit dieser Fall erledigt ist, da $\sigma(t) = t_0$.

ϵ-cut, *ext* Da $\mathbf{Term}^0 \subseteq \mathbf{Term}^+$, können die Regelanwendungen direkt übernommen werden.

■

Der automatische Teil eines Beweissystems wird einige Regeln des Kalküls $\mathcal{M}_E$ – wenn überhaupt – nur mit gewissen Einschränkungen verwenden. Warum die Regeln *cut* und *ext*, sowie die beiden *weak*-Regeln nicht verwendet werden können, wurde schon erläutert. Die *ϵ-cut*-Regel sollte allerdings vom Automaten verwendbar sein. Man erinnere sich, daß im Kalkül $\mathcal{G}_S$ der Indeterminismus bei der Auswahl des ϵ-Terms für diese Regel vermindert wurde, indem nur ϵ-Terme herangezogen wurden, die in einer atomaren Formel der Sequenz vorkommen, also $\epsilon x.\phi \in \mathrm{GT}(\Gamma \vdash \Delta)$. Auch die $\exists r$-Regel unterlag einer solchen Einschränkung. Es soll nun gezeigt werden, daß die *ϵ-cut* Regel des Kalküls $\mathcal{M}_E$ in gleicher Weise eingeschränkt werden kann.[10]

6.10 Satz (Vollständigkeit von $\mathcal{M}_S$)
Sei $\mathcal{M}_S$ der Kalkül, der sich von $\mathcal{M}_E$ nur durch die veränderte Regel

$$\frac{\Gamma \vdash \exists x.\phi, \Delta \quad [x/\epsilon x.\phi](\phi), \Gamma \vdash \Delta}{\Gamma \vdash \Delta}\ \epsilon\text{-}cut$$
$$\textit{wobei } \epsilon x.\phi \in \mathrm{GT}(\Gamma \vdash \Delta).$$

unterscheidet, wobei $\mathrm{GT}(\Gamma \vdash \Delta)$ *entsprechend Definition 5.16 die Menge der in atomaren Formeln der Sequenz vorkommenden* ganzen *Terme ist.*

*Für jede S-allgemeingültige geschlossene Formel $\phi \in$ **Form**^{0} existiert ein geschlossener $\mathcal{M}_S$-Beweis.*

Der Beweis verläuft größtenteils identisch zum Beweis der Vollständigkeit von $\mathcal{M}_E$, nur wird jetzt $\mathcal{G}_S$ statt $\mathcal{G}_E$ verwendet. Der Fall für die $\exists r$-Regel braucht trotz deren hinzugekommener Einschränkung im Grundkalkül nicht angepaßt zu werden.

Für die *ϵ-cut* Regel ist ein Term $\epsilon x.\phi_0 \in \mathrm{GT}(\Gamma_0 \vdash \Delta_0)$ gegeben. Es ist zu zeigen daß $\epsilon x.\phi \in \mathrm{GT}(\Gamma \vdash \Delta)$ für ein ϕ mit $\phi_0 = \sigma(\phi)$, wobei $\Gamma_0 \vdash \Delta_0 = \sigma(\Gamma \vdash \Delta)$. Man betrachte das Vorkommen von $\epsilon x.\phi_0$ als Teilterm von t_0 in einer atomaren Formel $p(\ldots, t_0, \ldots) \in \Gamma_0 \cup \Delta_0$. Es gibt eine entsprechende Formel $p(\ldots, t, \ldots) \in \Gamma \cup \Delta$ mit $t_0 = \sigma(t)$. Da $\epsilon x.\phi_0$ also Teilterm von $\sigma(t)$ ist, ergeben sich zwei Fälle:

Beweisskizze.
1. t enthält bereits einen Teilterm $\epsilon x.\phi$, der dem Vorkommen von $\epsilon x.\phi_0$ in t_0 entspricht, also $\epsilon x.\phi_0 = \sigma(\epsilon x.\phi)$. Dann ist $\epsilon x.\phi \in \mathrm{GT}(\Gamma \vdash \Delta)$.

2. t enthält eine Metavariable X_k, so daß $\epsilon x.\phi_0$ Teilterm von $\sigma(X_k) = t_k$ ist. Aufgrund der Einschränkung an die $\exists r$-Regel von $\mathcal{G}_S$ muß $t_k \in \mathrm{GT}(\Sigma_0')$ sein, wobei Σ_0' die Sequenz im Knoten k von B_0 ist. Dieser Knoten muß im Beweis *unterhalb* des gerade betrachteten Knotens liegen, da die im Knoten k von B neu eingeführte Metavariable X_k in $\Gamma \vdash \Delta$ vorkommt. Der Knoten k in B enthält eine Sequenz Σ', für die nun eine ähnliche Fallunterscheidung bezüglich des Vorkommens von $\epsilon x.\phi$ durchgeführt werden kann. Dieser Prozeß werde solange wiederholt, bis Fall 1 eintritt. Dies muß irgendwann geschehen, da die betrachteten Knoten immer näher an der Wurzel liegen.[11]

   Man findet also einen Knoten k^* in B mit Sequenz $\Gamma^* \vdash \Delta^*$ und $\epsilon x.\phi \in \mathrm{GT}(\Gamma^* \vdash \Delta^*)$. Da der konstruierte Beweis B alle atomaren Formeln nach oben „vererbt", muß auch $\epsilon x.\phi \in \mathrm{GT}(\Gamma \vdash \Delta)$ sein.

■

Damit ist die Grundlage für ein automatisches Beweissystem geschaffen, das für die ϵ-Logik mit substitutiver Semantik vollständig ist. Es soll nun noch eine Variante der *ϵ-cut*-Regel vorgestellt werden, die beim automatischen Beweisen eventuell Vorteile bringen könnte.

[10] Auch hier braucht man sogar nur diejenigen ϵ-Terme zu betrachten, die *nicht* von der $\exists l$-Regel eingeführt werden, siehe S. 46.

[11] Es sollte klar sein, wie diese Argumentation zu einem strengen induktiven Beweis gemacht werden kann.

Bei der Erläuterung der Regeln von $\mathcal{M}_E$ wurde bemerkt, daß es nicht sinnvoll ist, in den Regeln *ϵ-cut* und *ext* Constraints zu verwenden, da diese den Beweis nur verlängern können. Im Kontext des automatischen Beweisens ist trotzdem die Verwendung einer *ϵ-cut* Regel der Form

$$\frac{\Gamma \vdash \exists x.\phi \ll C, \Delta \quad [x/\epsilon x.\phi](\phi) \ll C, \Gamma \vdash \Delta}{\Gamma \vdash \Delta}\ \epsilon\text{-}cut$$
$$\text{wobei } \epsilon x.\phi \in \mathrm{GT}(\Gamma \vdash \Delta).$$

zu erwägen, wobei der Automat für C den Constraint jener atomaren Formel in $\Gamma \vdash \Delta$ verwenden würde, in der $\epsilon x.\phi$ vorkommt. Daß diese Variante der Regel korrekt ist, ist unmittelbar klar. Auch das Vollständigkeitsargument für $\mathcal{M}_S$ läßt sich auf die angegebene Verwendung übertragen.

Welche Variante zu bevorzugen ist, läßt sich wohl nur anhand einer Implementierung feststellen. Die Regel mit Constraints erlaubt weniger Regelanwendungen und Abschlüsse, verkleinert also den Suchraum. Andererseits werden Beweise evtl. größer. Es handelt sich also um eine neue Inkarnation des klassischen Dilemmas von Kalkülen für automatische Beweissysteme!

6.4.3. Verhältnis zur δ^{++}-Regel

In Abschnitt 5.1 wurde bereits verlangt, daß der zu entwickelnde Kalkül mindestens so stark sein sollte wie ein Tableaukalkül mit freien Variablen und δ^{++}-Skolemisierung. Es soll nun gezeigt werden, daß der Kalkül $\mathcal{M}_S$ dank der Verwendung von ϵ-Termen sogar exponentiell kürzere Beweise erlaubt als ein Kalkül mit δ^{++}-Regel.

Dazu soll zunächst eine Metavariablen/Sequenzen-Version der δ^{++}-Regel angegeben werden, um den Vergleich zwischen den „positiven“ Sequenzenbeweisen und den „negativen“ Tableaubeweisen zu erleichtern. Die δ^{++}-Regel ist eine Skolemisierungsregel, entspricht also der $\exists l$-Regel von $\mathcal{M}_E$. Sie wird in [7] mittels einer Äquivalenzrelation zwischen existenziell quantifizierten Formeln definiert, nämlich

$$\exists x.\phi \sim \exists y.\psi \quad \text{gdw.} \quad \exists x.\phi \text{ und } \exists y.\psi \text{ sind bis auf Variablenumbenennung gleich}$$

Dabei bezieht sich die Umbenennung auf Metavariablen, ebenso wie gebundene Variablen. Jeder Äquivalenzklasse $[\exists x.\phi]$ bezüglich $\sim$ wird ein Skolemsymbol $f_{[\exists x.\phi]}$ zugeordnet, dessen Stelligkeit die Anzahl der Metavariablen in $\exists x.\phi$ ist. Damit lautet die Regel wie folgt:

$$\frac{[x/f_{[\exists x.\phi]}(X_1, \ldots, X_n)](\phi), \Gamma \vdash \Delta}{\exists x.\phi, \Gamma \vdash \Delta}\ \delta^{++}$$

wobei $\{X_1, \ldots, X_n\} = \mathrm{mv}(\exists x.\phi)$. Um in einer ϵ-Logik die volle Stärke dieser Regel zu erreichen, muß man zu einer Semantik übergehen, die gegenüber gebundener Umbenennung stabil ist, siehe Abschnitt 5.7. Darauf soll hier verzichtet werden, denn:

- Bei jeder Anwendung der δ^{++}-Regel muß eine Implementierung ermitteln, ob schon ein Skolemsymbol für eine bis auf gebundene Umbenennung gleiche Formel eingeführt wurde. Dies kann, je nach den verwendeten Datenstrukturen, zeitaufwendig sein.

- Für den in der Praxis wichtigsten Fall – nämlich die Identifikation von Formeln, die aus sukzessiven Anwendungen der γ-Regel auf die gleiche Formel entstehen – reicht die Umbenennung von *freien* Variablen.

- Aus genau diesen Gründen berücksichtigt die Implementierung der Erfinder der δ^{++}-Regel in $_3T^AP$ gar nicht die Gleichheit bis auf gebundene Umbenennung.

In der folgenden Diskussion wird also von einer δ^{++}-Regel ausgegangen, die auf einer schwächeren Äquivalenzrelation beruht, nämlich

$$\exists x.\phi \;\sim\; \exists y.\psi \quad \text{gdw.} \quad \exists x.\phi \text{ und } \exists y.\psi \text{ sind bis auf Umb. von Metavariablen gleich.}$$

Man sieht leicht, daß δ^{++}-Skolemisierung durch ϵ-Terme simuliert werden kann. Man muß nur jeden Term

$$f_{[\exists x.\phi]}(t_1, \ldots, t_n),$$

der in einem Beweis vorkommt, durch einen ϵ-Term

$$[X_1/t_1, \ldots, X_n/t_n](\epsilon x.\phi)$$

ersetzen. Alle Teilterme des ursprünglichen Terms kommen auch (evtl. in übersetzter Form) in der Übersetzung vor, daher bleibt jede Gleichungseinsetzung möglich. Die anderen Regeln werden durch die Unterschiedlichkeit der Terme gar nicht berührt. Abschlüsse, die im Kalkül mit δ^{++}-Regel möglich waren, sind auch im übersetzten Beweis möglich, da die angegebene Übersetzung funktional ist, gleiche Terme also wieder gleichen Termen zugeordnet werden, und mit der Anwendung von Grundsubstitutionen kommutiert.

Nun soll eine Formelklasse angegeben werden, für die in $\mathcal{M}_S$ exponentiell kürzere Beweise möglich sind als in einem Kalkül mit der δ^{++}-Regel. Diese Formelklasse ähnelt der, die in [7] verwendet wird, um nachzuweisen, daß die δ^{++}-Regel stärker als die δ^{+}-Regel ist. Da sich diese Betrachtungen auf die Verwendung durch ein automatisches Beweissystem beziehen, soll die *cut*-Regel in den Beweisen nicht verwendet werden. Es sei

$$\begin{aligned} \phi_0 &:= a \doteq a \\ \phi_{n+1} &:= \exists x.\Big(\phi_n \wedge \big(p_n(x,a,b) \rightarrow \exists y.\forall x.p_n(x,y,b) \wedge \exists z.\forall x.p_n(x,a,z)\big)\Big) \quad \text{für } i \in \mathbb{N}. \end{aligned}$$

Dabei soll $\wedge$ stärker binden als $\rightarrow$. Der Beweis zu ϕ_0 hat natürlich in beiden Kalkülen nur einen Ast. In $\mathcal{M}_S$ ergibt sich folgender Beweis für ϕ_{n+1}, wobei alle Constraints $\perp$ sind, und daher weggelassen werden.

$$\dfrac{\dfrac{\vdots}{\vdash \phi_n} \qquad \dfrac{\dfrac{\dfrac{\dfrac{p_n(X,a,b) \vdash p_n(\epsilon x.\neg p_n(x,Y,b),Y,b)}{p_n(X,a,b) \vdash \forall x.p_n(x,Y,b)}\,\forall r}{p_n(X,a,b) \vdash \exists y.\forall x.p_n(x,y,b)}\,\exists r \qquad \dfrac{\dfrac{p_n(X,a,b) \vdash p_n(\epsilon x.\neg p_n(x,a,Z),a,Z)}{p_n(X,a,b) \vdash \forall x.p_n(x,a,Z)}\,\forall r}{p_n(X,a,b) \vdash \exists z.\forall x.p_n(x,a,z)}\,\exists r}{p_n(X,a,b) \vdash \exists y.\forall x.p_n(x,y,b) \wedge \exists z.\forall x.p_n(x,a,z)}\,\wedge r}{\vdash p_n(X,a,b) \rightarrow (\ldots \wedge \ldots)}\,\rightarrow r}{\dfrac{\vdash \phi_n \wedge (p_n(X,a,b) \rightarrow (\ldots \wedge \ldots))}{\vdash \phi_{n+1}}\,\exists r}\,\wedge r$$

Wenn im linken Ast rekursiv der entsprechende Beweis für ϕ_n eingesetzt wird, ist der Beweis mit

$$\sigma = [X/\epsilon x.\neg p_n(x,a,b), Y/a, Z/b]$$

geschlossen. Der Beweis für ϕ_n hat offenbar $2n+1$ Äste. Wie sieht nun der kürzest mögliche Beweis für ϕ_{n+1} mit der δ^{++}-Regel aus?

Ein möglicher Beweis ist:

$$
\dfrac{\dfrac{\dfrac{\vdots}{\vdash \phi_n} \qquad \dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\vdots}{\ldots \vdash \phi_n, \ldots} \quad \dfrac{p_n(X,a,b), p_n(X',a,b) \vdash p_n(g(Y),Y,b)}{p_n(X,a,b) \vdash p_n(X',a,b) \to \ldots, p_n(g(Y),Y,b)} \to r}{p_n(X,a,b) \vdash \phi_n \wedge (p_n(X',a,b) \to \ldots), p_n(g(Y),Y,b)} \wedge r}{p_n(X,a,b) \vdash p_n(g(Y),Y,b), \phi_{n+1}} \exists r}{p_n(X,a,b) \vdash \forall x.p_n(x,Y,b), \phi_{n+1}} \forall r}{p_n(X,a,b) \vdash \exists y.\forall x.p_n(x,y,b), \phi_{n+1}} \exists r \quad \dfrac{\dfrac{p_n(X,a,b) \vdash p_n(h(Z),a,Z), \phi_{n+1}}{p_n(X,a,b) \vdash \forall x.p_n(x,a,Z), \phi_{n+1}} \forall r}{p_n(X,a,b) \vdash \exists z.\forall x.p_n(x,a,z), \phi_{n+1}} \exists r}{p_n(X,a,b) \vdash \exists y.\forall x.p_n(x,y,b) \wedge \exists z.\forall x.p_n(x,a,z), \phi_{n+1}} \wedge r}{\vdash p_n(X,a,b) \to (\ldots \wedge \ldots), \phi_{n+1}} \to r}{\vdash \phi_n \wedge (p_n(X,a,b) \to (\ldots \wedge \ldots)), \phi_{n+1}} \wedge r}{\vdash \phi_{n+1}} \exists r
$$

wobei $g := f_{[\exists x.\neg p_n(x,Y,b)]}$ und $h := f_{[\exists x.\neg p_n(x,a,Z)]}$, also $g \neq h$. Der Beweis ist mit

$$\sigma = [X/h(b), X'/g(a), Y/a, Z/b]$$

geschlossen, wenn an zwei Stellen rekursiv der Beweis von ϕ_n eingesetzt wird. Es ist zwar möglich, diesen Beweis umzustellen, indem man etwa die $\exists r$-Regel gleich zu Anfang des Beweises zweimal anwendet. Man sieht jedoch, daß jeder Beweis für ϕ_{n+1}, der die δ^{++}-Regel verwendet, zwei Kopien des Beweises für ϕ_n enthalten muß. Daher hat jeder Beweis für ϕ_n mindestens 2^n Äste. Daraus ergibt sich folgende Aussage:

6.11 Satz (Beweiskomplexität ϵ vs. δ^{++})
*Es existiert eine Familie von allgemeingültigen Formeln $\phi_n \in$ **Form**^{0}, $n \in \mathbb{N}$ der Prädikatenlogik erster Stufe, so daß für die Anzahl $b^\epsilon(n)$ (resp. $b^{++}(n)$) von Ästen des kürzesten Beweises für ϕ_n in $\mathcal{M}_E$ (resp. mit der δ^{++}-Regel) gilt:*

- $b^\epsilon(n) \in O(n)$.
- $b^{++}(n) \in \Omega(2^n)$.

An dieser Stelle sei noch bemerkt, daß durch die Verwendung von universellen Variablen auch mit der δ^{++}-Regel Beweise mit $O(n)$ Ästen für die angegebene Formelklasse gefunden werden können, da die Formel $p_n(X,a,b) \to (\ldots\wedge\ldots)$ bezüglich X als universell erkannt werden kann. Die Formelklasse kann jedoch ohne weiteres so abgeändert werden, daß dies nicht mehr möglich ist. Dazu muß in der Definition von ϕ_{n+1} nur die rekursive Verwendung von ϕ_n durch $\psi(x) \to \phi_n$ ersetzt werden, wobei $\psi(x)$ eine allgemeingültige Formel ist, die x enthält. Da die Allgemeingültigkeit von ψ unentscheidbar ist, kann für jedes Verfahren zum Finden von universellen Variablen ein ψ angegeben werden, daß so komplex ist, daß die Universalität der für x eingeführten Metavariable X von dem Verfahren nicht mehr entdeckt werden kann.

6.5. Termordnungen für ϵ-Terme

Automatische Verfahren zur Gleichheitsbehandlung verwenden meist sog. Reduktionsordnungen. Dies sind noethersche Partialordnungen auf der Menge aller Terme, die mit Substitution und Termersetzung verträglich sind. Häufig wird noch verlangt, daß eine Reduktionsordnung auf den Grundtermen total ist. In der Literatur findet man zahlreiche Beispiele für Reduktionsordnungen, siehe etwa [2]. Diese sind jedoch alle nur für

„gewöhnliche“ Terme, wie die der klassischen Prädikatenlogik, definiert. Soll ein Verfahren, das Reduktionsordnungen verwendet, an eine Logik mit ϵ-Termen angepaßt werden, muß eine Reduktionsordnung auf ϵ-Termen verwendet werden.

Statt nun eine neue Reduktionsordnung zu definieren, soll in diesem Abschnitt gezeigt werden, wie man aus einer Reduktionsordnung für „gewöhnliche“ Terme eine solche für ϵ-Terme gewinnen kann.

6.12 Definition (Reduktionsordnung)
Eine Partialordnung $\succ$ auf **Term*** *heißt* **Reduktionsordnung**, *falls sie folgende Bedingungen erfüllt:*

- $\succ$ *ist noethersch, d.h. es existiert keine unendliche Kette* $s_0 \succ s_1 \succ s_2 \succ \ldots$
- *Für alle* $\sigma \in \mathbf{Sub}^*$ *und alle ganzen Terme* $s, t \in \mathbf{Term}^*$ *mit* $s \succ t$ *gilt:* $\sigma(s) \succ \sigma(t)$.
- *Für alle Terme* $u, s, t \in \mathbf{Term}^*$ *mit* $s \succ t$ *und jede Metavariable* $X \in \mathrm{mv}(u)$ *gilt:* $[X/s](u) \succ [X/t](u)$.[12]

Es ist nun möglich, ganze Terme, die ϵ-Terme enthalten können, als ϵ-freie Terme über einer modifizierten Signatur zu interpretieren. Man setzt

$$\mathbf{F}' := \mathbf{F} \,\dot\cup\, \mathbf{P} \,\dot\cup\, \mathbf{V} \,\dot\cup\, \{\doteq, \wedge, \neg\} \,\dot\cup\, \{\exists x. \mid x \in \mathbf{V}\} \,\dot\cup\, \{\epsilon x. \mid x \in \mathbf{V}\} \qquad (*)$$

Dabei werden Prädikatensymbole als Funktionssymbole gleicher Stelligkeit interpretiert, Variablen $x \in V$ als Konstantensymbole, $\doteq$ und $\wedge$ als zweistelliges Funktionssymbol mit „infix“-Notation, und $\neg$, sowie $\exists x.$ und $\epsilon x.$ als einstellige Funktionssymbole. Unter dieser Interpretation enthält ein ganzer Term keine Objektvariablen mehr, da diese als Konstantensymbole interpretiert werden. Es gibt auch keine ϵ-Terme mehr, sondern nur noch Terme mit den neuen Funktionssymbolen „$\epsilon x.$“. Die Terme sind allerdings nicht notwendigerweise Grundterme, da sie ja noch Metavariablen enthalten können.

Sei I die Abbildung, die einem ganzen Term über der ursprünglichen Signatur den mittels der modifizierten Signatur interpretierten Term zuordnet. Da ganze Substitutionen nur auf Metavariablen wirken, verhalten sich offenbar ganze Terme unter ganzen Substitutionen bei dieser Interpretation genauso wie sonst. Das heißt, es ist $I(\sigma(t)) = I(\sigma)(I(t))$ für beliebige $\sigma \in \mathbf{Sub}^*$ und $t \in \mathbf{Term}^*$, wobei $I(\sigma)$ die durch $I(\sigma)(X) := I(\sigma(X))$ definierte Substitution auf Termen der modifizierten Signatur ist. Ferner ist I offenbar injektiv. Damit erhält man leicht folgende Aussage:

6.13 Satz (ϵ-Reduktionsordnung)
Sei $\succ_0$ eine Reduktionsordnung für die ϵ-freien Terme über der durch $(*)$ gegebenen modifizierten Signatur. Dann ist die durch

$$s \succ t \quad \text{gdw.} \quad I(s) \succ_0 I(t)$$

definierte Relation ist eine Reduktionsordnung auf den ganzen Termen **Term***. *Falls $\succ_0$ auf den Grundtermen der modifizierten Signatur total ist, so ist $\succ$ total auf den Grundtermen* **Term**^{0}.

[12] Diese Bedingung wird meistens mit Hilfe von Positionen notiert. Die vorliegende Definition ist äquivalent, paßt besser zu den in dieser Arbeit verwendeten Gleichheitsregeln und kommt ohne die Definition von Termpositionen aus.

Beweis. Daß $\succ$ noethersche Partialordnung ist, folgt direkt aus der entsprechenden Eigenschaft für $\succ_0$. Ferner gilt für Terme s, t mit $s \succ t$: $I(s) \succ_0 I(t) \implies I(\sigma)(I(s)) \succ_0 I(\sigma)(I(t)) \implies I(\sigma(s)) \succ_0 I(\sigma(t)) \implies \sigma(s) \succ \sigma(t)$, sowie für beliebige Terme u und Metavariablen $X \in \mathrm{mv}(u)$: $I(s) \succ_0 I(t) \overset{X \in \mathrm{mv}(I(u))}{\implies} [X/I(s)]I(u) \succ_0 [X/I(t)]I(u) \implies I([X/s])I(u) \succ_0 I([X/t])I(u) \implies I([X/s](u)) \succ_0 I([X/t](u)) \implies [X/s](u) \succ [X/t](u)$. Sei schließlich $\succ_0$ total auf Grundtermen, und $s, t \in \mathbf{Term}^0$, $s \neq t$. $I(s)$ und $I(t)$ sind Grundterme. Da I injektiv ist, ist $I(s) \neq I(t) \implies I(s) \succ_0 I(t)$ oder $I(t) \succ_0 I(s) \implies s \succ t$ oder $t \succ s$. ■

Es läßt sich auf diesem Wege also jede der in der Literatur angegebenen Reduktionsordnungen auf ϵ-Terme erweitern. Für eine lexikalische Pfadordnung muß die zugrundeliegende Ordnung auf den Funktions- und Konstantensymbolen natürlich die in der Signatur $\mathbf{F}'$ hinzugekommenen Symbole einbeziehen, für eine Knuth-Bendix-Ordnung müssen den zusätzlichen Symbolen Gewichte zugeordnet werden, etc. Welche Wahl von Ordnung, bzw. Gewichten praktisch sinnvoll ist, bleibt noch zu erforschen. Einige Experimente mit einer Implementierung bringen hier wahrscheinlich schnell Aufschluß. Zu bemerken ist noch, daß auch bei klassischen Skolemisierungsverfahren die Reduktionsordnung erweitert werden muß, um die im Laufe des Beweises eingeführten Skolemsymbole zu berücksichtigen.

6.6. Präsentation von Beweisen mit Constraints

In Abschnitt 4.5 wurde bereits ein Ansatz zur Präsentation von Beweisen mit Metavariablen und Constraints beschrieben. Nun stehen die technischen Hilfsmittel zur Verfügung, um diesen Ansatz genauer zu beschreiben und zu verfeinern.

Es wurde nach einem Mechanismus gesucht, der es dem Benutzer erlaubt, die von ihm intendierte Instantiierung von Metavariablen anzugeben. Da Constraints für den Kalkül ohnehin notwendig sind, wurde vorgeschlagen, die Constraintsprache für diese Angabe zu verwenden. Das System verwaltet also getrennt vom Beweisbaum einen Constraint, den der Benutzer nach Belieben verändern kann. Dieser Constraint soll künftig als *Benutzer-Constraint* bezeichnet werden. Das System verwendet den Benutzer-Constraint einerseits zur Beweispräsentation, also zur Darstellung des aktuellen Beweisbaums für den interaktiv beweisenden Benutzer. Andererseits wurde bereits angedeutet, daß der Benutzer mit Hilfe dieses Constraints den Suchraum des automatischen Beweisers einschränken kann. Darauf wird im folgenden Abschnitt eingegangen.

Zunächst soll die Beweispräsentation diskutiert werden. Es geht darum, zu einem gegebenen Beweisbaum und einem Benutzer-Constraint, eine Darstellung des ersteren zu finden, die die durch den Benutzer-Constraint beschriebene intendierte Instantiierung der Metavariablen berücksichtigt. Zum Beispiel gilt im Falle eines Constraints

$$X = a \sqcup f(X) = Y$$

für jede erfüllende Grundsubstitution $\sigma(X) = a$ und $\sigma(Y) = f(a)$. Daher ist es sicherlich angebracht, beim Anzeigen des Beweises für den Benutzer anstelle der Metavariablen im Normalfall die entsprechenden Terme einzusetzen. Wenn der Benutzer nun

$$f(X) = Y \sqcup Z = f(Y)$$

angibt, werden keine bestimmten Grundterme mehr als Instantiierung der Variablen erzwungen. Dennoch gilt natürlich $\sigma(Y) = f(\sigma(X))$ und $\sigma(Z) = f(f(\sigma(X)))$ für jede erfüllende Substitution σ. Man könnte also statt Y und Z die Terme $f(X)$ und $f(f(X))$ anzeigen. Verwendet man jedoch eine komplexere Constraintsprache, etwa eine mit Ordnungsconstraints $s \prec t$, oder einer Disjunktion $C_1 \sqcap C_2$, wird die Lage komplizierter. Was soll etwa mit einem Constraint

$$f(X) \prec Y \ \sqcap \ Z = f(Y)$$

geschehen? Wenn der Benutzer derartige Constraints eingibt, ist es wahrscheinlich vernünftig, die entsprechenden Metavariablen stehen zu lassen.

Man kann diesen Ansatz formalisieren, indem man jedem Benutzer-Constraint B eine ganze Substitution $\mathrm{Pr}(B)$ zuordnet, die auf alle Formeln des Beweises angewandt wird, wenn dieser angezeigt werden soll. Nun fordert man, daß für eine Metavariable nur dann bei der Präsentation durch $\mathrm{Pr}(B)$ ein anderer Term eingesetzt werden darf, wenn dadurch keine Information über mögliche Abschlüsse verloren geht. Jede Grundinstanz des *tatsächlichen* Beweisbaums mit Metavariablen sollte auch Instanz des *angezeigten* Beweisbaums sein. Dies ist der Fall, wenn für jede Grundsubstitution $\sigma \in \mathrm{Sat}(B)$, die den Benutzer-Constraint erfüllt, und jede Formel ϕ gilt:

$$\sigma(\phi) = \sigma(\mathrm{Pr}(B)(\phi)).$$

Daher fordert man $\sigma = \sigma \circ \mathrm{Pr}(B)$ für alle $\sigma \in \mathrm{Sat}(B)$, oder mit Hilfe von Lemma 6.4: $\mathrm{Sat}(B) \subseteq \mathrm{Sat}(\mathrm{Con}(\mathrm{Pr}(B))$, d.h.

$$\mathrm{Con}(\mathrm{Pr}(B)) \sqsubseteq B. \tag{1}$$

Andererseits sollte der Präsentationsmechanismus stark genug gewählt werden, um zumindest die aus Substitutionen abgeleiteten Constraints $\mathrm{Con}(\mu)$ wieder verlustfrei in Substitutionen umzuwandeln, das heißt

$$\mathrm{Pr}(\mathrm{Con}(\mu))) \approx \mu, \tag{2}$$

wobei $\mu \approx \mu'$, falls Substitutionen ν, ν' existieren mit $\nu \circ \mu = \mu'$ und $\nu' \circ \mu' = \mu$, bzw. laut Lemma 6.4 gleichwertig

$$\mathrm{Con}(\mathrm{Pr}(\mathrm{Con}(\mu))) \equiv \mathrm{Con}(\mu). \tag{2'}$$

Es war schon öfters davon die Rede, daß der Benutzer den Kalkül wie einen Grundkalkül verwenden können soll, wenn er die zusätzlichen Möglichkeiten von Constraints und Metavariablen nicht nutzen will. Anhand der geforderten Eigenschaften von Pr soll nun gezeigt werden, wie dies verwirklicht werden kann. Bei der Verwendung eines Grundkalküls muß der Benutzer bei jeder Anwendung der $\exists r$-Regel einen Grundterm t_i angeben, mit dem die quantifizierte Formel instantiiert wird. Im Kalkül $\mathcal{M}_E$ wird die Formel statt dessen mit einer Metavariablen X_i instantiiert. Der gewünschte Effekt wird nun erreicht, wenn bei der Regelanwendung der Constraint $X_i = t_i$ zum Benutzer-Constraint B hinzugenommen wird, d.h. der neue Benutzer-Constraint lautet $B \sqcup (X_i = t_i)$. Insgesamt wird der Benutzer-Constraint dann die Form

$$B = \bigsqcup_{i=1}^{k} (X_i = t_i) = \mathrm{Con}([X_i/t_i]) \tag{$*$}$$

haben, also laut (2): $\mathrm{Pr}(B) \approx [X_i/t_i]$, bzw. mit (2′): $\mathrm{Con}(\mathrm{Pr}(B)) \equiv B$. Da es sich bei $[X_i/t_i]$ um eine Grundsubstitution handelt, gilt sogar $\mathrm{Pr}(B) = [X_i/t_i]$. Die Präsentationsfunktion sorgt also dafür, daß überall anstelle der Metavariablen wie erwünscht die entsprechenden Grundterme angezeigt werden. Die Anwendung der Gleichheitsregeln wird sich auf solche Fälle beschränken, wo (in der Terminologie der $\overset{\rightarrow}{=}l$-Regel von Abb. 6.1) die Darstellung des Terms s, also der Grundterm $\mathrm{Pr}(B)(s)$, Teilterm der Darstellung von ϕ ist, d.h. $\mathrm{Pr}(B)(\phi) = [y/\,\mathrm{Pr}(B)(s)](\phi')$ für ein ϕ'. Dies ist der Fall, wenn man eine Substitution μ mit $\mathrm{Pr}(B) = \mathrm{Pr}(B) \circ \mu$ wählt, also $\mathrm{Con}(\mu) \sqsubseteq \mathrm{Con}(\mathrm{Pr}(B)) \equiv B$. Wenn im folgenden davon geredet wird, den Kalkül „als Grundkalkül zu verwenden“, ist gemeint, daß der Benutzer-Constraint die Form (∗) mit Grundtermen t_i hat, und nur Formeln mit Constraints $C \sqsubseteq B$ vorkommen.

Wenn man nun zuläßt, daß der Benutzer beliebige ganze Terme, und nicht nur Grundterme, eingeben darf, so ergibt sich für B immer noch die Form (∗), wobei $[X_i/t_i]$ jetzt allerdings eine beliebige ganze Substitution sein kann. Es gilt also weiterhin $\mathrm{Pr}(B) \approx [X_i/t_i]$, nur die Gleichheit gilt im allgemeinen nicht mehr. Praktisch bedeutet dies Folgendes: Wenn der Benutzer eine Instantiierung $t = Y$ für die Metavariable X eingibt, kann die Funktion Pr für den generierten Constraint $X = Y$ entweder $[X/Y]$ oder $[Y/X]$ erzeugen. Bei einer Implementierung sollte darauf geachtet werden, daß sich Pr in solchen Fällen möglichst intuitiv verhält. Diese Verwendungsweise gibt dem Benutzer gegenüber einem reinen Grundkalkül die Möglichkeit, sich durch Verwendung von Constraints wie $X = f(a, b) \sqcup Y = g(h(i(X)))$ in einer Instantiierung bequem auf eine andere Instantiierung zu beziehen, wobei sich nachträgliche Änderungen der ersten Instantiierung auf die spätere übertragen. Ferner wird ihm die Möglichkeit gegeben, „uninteressante“ oder maschinell leicht zu findende Teile einer Instantiierung offenzulassen. Man bemerke, daß in solchen Fällen nicht alle verwendeten Metavariablen im Beweis vorzukommen brauchen. Durch Constraints wie $X = f(g(f(Y, Y)), a)$ kann der Benutzer eine Metavariable Y einführen, die als Platzhalter für einen nicht näher spezifizierten Grundterm dient. Um jedoch bei einer einfachen Präsentation zu bleiben, muß er sich bei der Anwendung von Gleichheitsregeln nach wie vor darauf beschränken, nur solche Gleichungen anzuwenden, deren Anwendbarkeit sich aus der Präsentation ergibt, die also nur Constraints $C \sqsubseteq B$ einführen. Es wird künftig gesagt werden, der Benutzer „verwende den Kalkül als quasi-Grundkalkül“, wenn $B = \mathrm{Con}(\mu)$ für eine beliebige ganze Substitution μ ist, und der Beweis nur Formeln mit Constraints $C \sqsubseteq B$ enthält.

Der Benutzer-Constraint soll nicht nur die Darstellung der Formeln selber beeinflussen, sondern auch die Darstellung der zu diesen gehörigen Constraints. In einem Grundkalkül existieren keine Constraints, daher sollte der Benutzer normalerweise nicht mit diesen nicht konfrontiert werden, solange er nur die Möglichkeiten eines Grundkalküls nutzt. Bei gegebenem Benutzer-Constraint B können die Formeln in einem Beweis nach den zu ihnen gehörigen Constraints C in drei „Typen“ aufgeteilt werden:

- $C \sqsubseteq B$. Dies gilt insbesondere für $C = \bot$. Alle Instantiierungen der Metavariablen, die der Benutzer in B zuläßt, erfüllen auch C. Diese Constraints sind für den Benutzer unwichtig, da sie Einschränkungen angeben, die schwächer als die von ihm selbst angegebenen sind. Wenn der Kalkül als (quasi-)Grundkalkül benutzt wird, gilt außerdem $\mathrm{Con}(\mathrm{Pr}(B)) \equiv B$, und dann folgt $\sigma \in \mathrm{Sat}(C)$ für alle σ mit $\sigma = \sigma \circ \mathrm{Pr}(B)$, d.h. die durch den Constraint gegebene Einschränkung wird bereits durch die zum Zwecke der Präsentation vorgenommene Substitution erfüllt. Es empfiehlt sich also, solche Constraints im Normalfall gar nicht anzuzeigen, sondern nur die zugehörigen Formeln. Die Constraints erlangen Bedeutung, wenn der Benutzer B so abändert, daß $C \sqsubseteq B$ nicht mehr gilt.

- $C \sqcup B$ ist unerfüllbar. Solange der Benutzer-Constraint nicht geändert wird, kann die durch C eingeschränkte Formel für einen Beweisabschluß, der B erfüllt, nicht verwendet werden. Die Formel ist daher innerhalb des vom Benutzer durch B gegebenen Suchraums für Instantiierungen der Metavariablen überflüssig. Sie sollte demnach überhaupt nicht angezeigt werden. Wenn sich der automatische Beweiser bei der Beweissuche an den Benutzer-Constraint hält, sollte er solche Formeln überhaupt nicht erzeugen. Sie können also nur entstehen, indem der Benutzer B ändert.

- $C \sqcup B$ ist erfüllbar, aber $C \not\sqsubseteq B$. Auch dieser Typ von Formeln tritt nicht auf, solange der Kalkül als quasi-Grundkalkül verwendet wird. Wenn solche Formeln vorhanden sind, werden also offenbar die Möglichkeiten von Constraints und Metavariablen stärker genutzt. Daher müssen solche Formeln *mit* den zugehörigen Constraints angezeigt werden.

Man bemerke, daß es zur automatischen Feststellung dieser Typen nötig ist, die Subsumtion von Constraints zu entscheiden. Es sollte bei der Wahl der Constraintsprache also darauf geachtet werden, daß dies effizient möglich ist. Wenn kein effizientes Entscheidungsverfahren existiert, könnte eine hinreichende Bedingung implementiert werden, und in der obigen Klassifizierung „$C \sqsubseteq B$“ durchgehend ersetzt werden durch „$C \sqsubseteq B$ ist vom System feststellbar“. Dann würden nur solche Constraints unterdrückt werden, für die die hinreichende Bedingung sicherstellt, daß sie vom Benutzer-Constraint subsumiert werden.

Sollte sich in der praktischen Anwendung herausstellen, daß häufig längere Konjunktionen $C_1 \sqcup \ldots \sqcup C_k$ als Constraints auftauchen, wäre eine Verfeinerung dieses Ansatzes in Erwägung zu ziehen: Man teilt die Komponenten C_i in die eben beschriebenen Typen auf. Tritt eine Komponente des zweiten Typs auf, so ist die ganze Formel mit Constraint vom zweiten Typ und wird nicht angezeigt. Ansonsten wird die Formel angezeigt, aber nur mit den Constraint-Komponenten vom dritten Typ.

In jedem Fall empfiehlt es sich wohl, dem Benutzer mehrere Abstraktionsstufen bei der Präsentation anzubieten. Zwischen dem beschriebenen Verfahren, und der vollständigen Präsentation aller Formeln mit Constraints, könnte etwa eine Präsentation angeboten werden, in der immerhin auf das Vorhandensein von nicht angezeigten Formeln/Constraints aufmerksam gemacht wird.

An dieser Stelle kann auch eine weitere Frage geklärt werden: Wie kann das System dem Benutzer die Gleichheitsregeln anbieten, ohne ihn mit deren voller Komplexität zu konfrontieren? Hier geht es nicht mehr, wie bisher, um die optimale *Darstellung* eines bestehenden Beweises, sondern darum, wie Beweisregeln dem Benutzer *angeboten* werden können. In einem Grundkalkül genügt es, zur Anwendung einer Gleichheitsregel, die anzuwendende Gleichung und den zu ersetzenden Teilterm einer Formel zu identifizieren. Seien zum Beispiel die Sequenz

$$X \doteq f(Y), p(Z, b) \vdash$$

und der aktuelle Benutzer-Constraint

$$X = h(Y) \ \sqcup \ Y = a \ \sqcup \ Z = g(h(a), c) \tag{†}$$

gegeben. Mit $\mathrm{Pr}(B) = [X/h(a), Y/a, Z/g(h(a), c)]$ wird dem Benutzer die Sequenz

$$h(a) \doteq f(a), p(g(h(a), c), b) \vdash$$

angezeigt. Nun möchte dieser die Gleichung wie folgt anwenden:[13]

$$\frac{p(g(f(a), c), b), h(a) \doteq f(a), p(g(h(a), c), b) \vdash}{h(a) \doteq f(a), p(g(h(a), c), b) \vdash} \stackrel{\rightarrow}{=} l \tag{‡}$$

Es wäre ungeschickt, vom Benutzer die Eingabe der zur Anwendung der $\stackrel{\rightarrow}{=} l$-Regel nötigen Substitution μ zu verlangen. Schließlich bezieht sich diese auf Metavariablen, für die er bereits mit Hilfe von B Instantiierungen angegeben hat. Das System sollte also selber eine sinnvolle Wahl für μ treffen können. Die einfachste Wahl ist sicherlich $\mu := \mathrm{Pr}(B)$. Die neue Formel erhält dann den zusätzlichen Constraint $\mathrm{Con}(\mu) = B$ und ist somit vom ersten Typ.

Diese Wahl könnte jedoch ungeschickt sein, da Änderungen des Benutzer-Constraints, die gar nicht die Anwendbarkeit der Gleichung betreffen, trotzdem zur Verletzung der Bedingung $C \sqsubseteq B$ führen können. Soll die neue Formel unter einer möglichst großen Menge von Benutzer-Constraints noch zum ersten Typ zählen, so muß $\mathrm{Con}(\mu)$ offenbar so schwach wie möglich gewählt werden. Das System sollte für μ also eine möglichst *allgemeine* Substitution wählen, unter der die vom Benutzer gewählte Gleichung an der angegebenen Position anwendbar ist. Die allgemeinste Substitution μ, die in obigem Beispiel die Regelanwendung erlaubt, ist $\mu = [Z/g(X, U)]$, wobei U eine neue Metavariable ist. Das System kann also die Regelanwendung

$$\frac{p(g(f(Y), U), b) \ll (Z = g(X, U)),\ X \doteq f(Y), p(Z, b) \vdash}{X \doteq f(Y), p(Z, b) \vdash} \stackrel{\rightarrow}{=} l$$

erzeugen. Da dem Benutzer die Sequenz (‡) präsentiert werden soll, muß ein Constraint $U = c$ zum Benutzer-Constraint (†) hinzugenommen werden. Es ist fraglich, ob eine Änderung des Benutzer-Constraints durch das System wünschenswert ist. Ferner müßte bei einer nachträglichen Änderung von B, die zum Beispiel $Z = g(h(a), d)$ fordert, auch der hinzugekommene Constraint für U zu $U = d$ angepaßt werden, wenn die resultierende Formel weiterhin vom ersten Typ sein soll. Eventuell müßte eine neue Sorte von Constraints eingeführt werden, die besagen, daß ein Term an einer bestimmten Position in einem anderen vorkommt, etwa der Form $U = Z_{.2} \ldots$

Man sieht also, daß auch die Wahl der *allgemeinsten* Substitution für μ einige Schwierigkeiten mit sich bringt. Es wurde daher nicht versucht, hierfür eine Theorie zu entwickeln.[14]

Eine Wahl für μ, die zwar einfach, aber nicht zu restriktiv ist erhält man, wenn man mit $\mu := \mathrm{Pr}(B)|_{\mathrm{mv}(s) \cup \mathrm{mv}(\phi)}$ die vom Benutzer-Constraint gegebene Substitution auf die in den Formeln tatsächlich vorkommenden Metavariablen einschränkt. Es sollte in einer Implementierung zunächst dieses Verfahren erprobt werden. Wenn sich herausstellt, daß Benutzer an B häufig nachträglich größere Änderungen vornehmen, kann die Implementierung einer allgemeineren Wahl von μ erwogen werden.

[13] Aus diesem Beispiel wird übrigens ersichtlich, wie wichtig die Anwendung von Gleichungen „unterhalb von Variablenpositionen" für einen integrierten Kalkül ist.

[14] Man bemerke, daß es sich hier *nicht* um den beim automatischen Beweisen üblichen allgemeinsten Unifikator von s mit einem Teilterm von ϕ handelt!

In einem Kalkül ohne Metavariablen ist jeder Ast eines Beweises geschlossen oder offen, je nachdem, ob es sich beim Blatt um ein Axiom handelt, oder nicht. Der Beweis ist geschlossen, wenn alle seine Äste geschlossen sind. In einem Kalkül mit Metavariablen sind Äste immer nur bezüglich einer bestimmten Instantiierung derselben offen oder geschlossen, und der Beweis ist abgeschlossen, falls eine Instantiierung existiert, unter der alle Äste geschlossen sind. Wenn jedoch der Benutzer auf die Verwendung von Metavariablen verzichtet, und für diese immer konkrete Instantiierungen angibt, so ist es wünschenswert, ihn darüber zu informieren, welche Äste *unter der derzeitigen Instantiierung* geschlossen sind. Ähnlich wie dies für Formeln geschehen ist, kann auch bei Sequenzen $\Gamma \vdash \Delta$ – und damit Ästen – zwischen drei Typen unterschieden werden.

- $C \sqsubseteq B$ für einen abschließenden Constraint C von $\Gamma \vdash \Delta$. Dann ist $\sigma(\Gamma \vdash \Delta)$ ein $\mathcal{G}_E$-Axiom für *jede* Grundsubstitution $\sigma \in \mathrm{Sat}(B)$. Der Benutzer-Constraint garantiert also den Abschluß eines Astes mit Blatt $\Gamma \vdash \Delta$.

- $C \sqcup B$ ist unerfüllbar für jeden abschließenden Constraint C von $\Gamma \vdash \Delta$. Dann ist $\sigma(\Gamma \vdash \Delta)$ unter *keiner* Grundsubstitution $\sigma \in \mathrm{Sat}(B)$ ein $\mathcal{G}_E$-Axiom. Der Ast ist also offen.

- $C \sqcup B$ ist erfüllbar für einen abschließenden Constraint C von $\Gamma \vdash \Delta$, aber $C' \not\sqsubseteq B$ für alle abschließenden Constraints C' von $\Gamma \vdash \Delta$. Damit ist $\sigma(\Gamma \vdash \Delta)$ mit einigen $\sigma \in \mathrm{Sat}(B)$ ein $\mathcal{G}_E$-Axiom, mit anderen nicht. Ob der Beweis insgesamt geschlossen ist, hängt davon ab, ob es eine abschließende Substitution gibt, die auch alle anderen Äste abschließt. Bei der Verwendung als Grundkalkül kann diese Sorte von Blättern nicht erzeugt werden, da dann $B = \mathrm{Con}([X_i/t_i])$ für Grundterme t_i. Damit ist $\sigma \in \mathrm{Sat}(B)$ auf den vorkommenden Metavariablen eindeutig festgelegt, $\sigma(\Gamma \vdash \Delta)$ also für alle σ gleich. Bei der Verwendung als quasi-Grundkalkül kann diese Sorte von Ästen allerdings durchaus vorkommen.

Bei der Präsentation des Beweisbaums wird eine visuelle Unterscheidung zwischen Ästen mit diesen verschiedenen Typen von Blättern die interaktive Benutzung sicherlich erleichtern. Insbesondere treten bei der Verwendung als Grundkalkül nur Sequenzen der beiden ersten Typen, also „offene“ und „geschlossene“ Äste auf.

6.7. Steuerung des automatischen Beweisers mit Constraints

Im letzten Abschnitt wurde die Verwendung des Benutzer-Constraints zur Beweispräsentation erläutert. In einem integrierten System hat er aber noch eine andere wichtige Anwendung. Er bietet nämlich eine Schnittstelle, über die der Benutzer den Suchraum für die automatische Beweissuche einschränken kann.

In einem typischen Szenario hat der Benutzer einen Teilbeweis erarbeitet, und sich dabei auf einen quasi-Grundkalkül beschränkt. Er hat also mit Hilfe des Benutzer-Constraints für einige der im Beweis vorkommenden Metavariablen mehr oder weniger konkrete Instantiierungen angegeben. Nun möchte er einen oder mehrere Äste des Beweises automatisch schließen lassen. Dann sollte der automatische Beweiser den Benutzer-Constraint als Hinweis auf einen möglichen Beweisabschluß respektieren, und nach einem

Beweis suchen, der mit einer Substitution geschlossen ist, die den Benutzer-Constraint erfüllt.

Natürlich kann der Benutzer einen unerfüllbaren Constraint angeben, worauf der Automat keinen Beweis finden kann, auch wenn die zu beweisende Formel gültig ist. Respektiert der Automat den Benutzer-Constraint, kann die Vollständigkeit also verlorengehen. Das Verfahren kann dann nur noch relativ zum Benutzer-Constraint vollständig sein:

> Falls ein Beweisbaum für eine Sequenz existiert, der mit einer Substitution $\sigma \in \mathrm{Sat}(B)$ geschlossen ist, wird ein solcher von dem automatischen Verfahren gefunden.

Beim Entwurf der automatischen Beweisprozedur muß darauf geachtet werden, daß diese Bedingung erfüllt ist. Es liegt in der Verantwortung des Benutzers, daß B den Suchraum nicht so stark einschränkt, daß er keinen geschlossenen Beweis mehr enthält.

Es gibt offenbar genau zwei Stellen, an denen der Benutzer-Constraint vom Automaten respektiert werden sollte, nämlich

1. bei der Suche nach einer abschließenden Substitution, und

2. bei der Anwendung von Gleichheitsregeln.

Die restlichen vom Automaten verwendeten Regeln des Kalküls beziehen sich nämlich nur auf die *Formel*-Struktur, die von der Instantiierung der Metavariablen, die ja Terme sind, unabhängig ist.

Laut Satz 6.7 muß für den Test, ob ein gegebener Beweis abgeschlossen ist, für jeden Ast ein abschließender Constraint C_i gefunden werden, so daß die Konjunktion $\bigsqcup C_i$ dieser Constraints erfüllbar ist. Verlangt man, daß nur abschließende Substitutionen berücksichtigt werden, die den Benutzer-Constraint erfüllen, so braucht man nur nach abschließenden Constraints C_i zu suchen, mit denen $B \sqcup \bigsqcup C_i$ erfüllbar ist.

Nun stellt sich die Frage, ob es sinnvoll ist, B beim Abschlußtest zu berücksichtigen. Schließlich ist der Beweis geschlossen, sobald ein erfüllbarer abschließender Constraint gefunden wird, unabhängig davon, ob es eine abschließende Substitution gibt, die auch B erfüllt. Es ist durchaus denkbar, daß der Automat, wenn er B nicht berücksichtigt, einen Abschluß findet, den der Benutzer übersehen hat. Trotzdem sollte B beim Abschlußtest im allgemeinen berücksichtigt werden, und zwar aus Effizienzgründen. Um dies zu erläutern, soll der Abschlußtest genauer betrachtet werden.

Wenn jedes Blatt des Beweisbaums nur einen abschließenden Constraint besitzt, d.h. *entweder* genau ein Paar von Formeln $p(\ldots) \ll \ldots \in \Gamma$, $p(\ldots) \ll \ldots \in \Delta$, *oder* genau eine Gleichung $s \doteq t \ll \ldots \in \Delta$, dann ist für den Abschlußtest einfach die Erfüllbarkeit der Konjunktion all dieser Constraints zu überprüfen. B dazuzunehmen bringt dann natürlich keinen Vorteil. Falls für einen der Äste kein abschließender Constraint existiert, ist der Beweis bestimmt nicht geschlossen, und es ist irrelevant ob B berücksichtigt wird.

Im allgemeinen können jedoch pro Blatt mehrere abschließende Constraints auftreten. Der Abschlußtest muß dann eventuell sämtliche Kombinationen von abschließenden Constraints durchprobieren, was zu einem in der Anzahl von Ästen exponentiellen Aufwand führt. Um den Aufwand in der Praxis möglichst gering zu halten, wird man versuchen, möglichst viele Kombinationen aus der Suche auszuschließen. Dies geschieht, indem man sukzessive (etwa mit Backtracking) abschließende Constraints für die Äste auswählt, die mit den bisher ausgewählten Constraints konsistent sind. Der folgende Algorithmus gibt

„abgeschlossen" zurück, falls es einen abschließenden Constraint zu einer Menge von Blättern gibt, die von einer Substitution in Sat($Akku$) erfüllt wird, und versagt (**fail**) sonst.

proc $\mathit{Abschlußtest}(\mathit{Blätter}, \mathit{Akku})$
if Akku erfüllbar **then**
  if $\mathit{Blätter} = \emptyset$ **then**
    return „abgeschlossen."
  else
    wähle $\Sigma \in \mathit{Blätter}$
    wähle mit backtracking: abschließender Constraint C von Σ
    return $\mathit{Abschlußtest}(\mathit{Blätter} \setminus \Sigma, C \sqcup \mathit{Akku})$
else
  fail

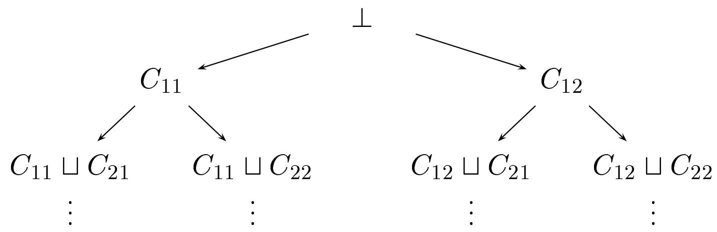
Der gewöhnliche Abschlußtest kann durch

$$\mathit{Abschlußtest}(\mathit{Blätter}, \perp)$$

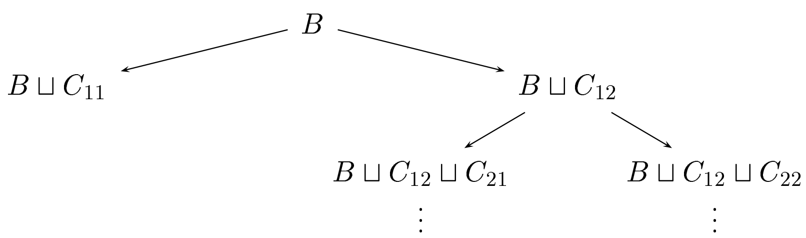
vorgenommen werden. Der Abschlußtest unter Berücksichtigung von B wird durch

$$\mathit{Abschlußtest}(\mathit{Blätter}, B)$$

realisiert. Bei letzterem Aufruf können manche Pfade im Backtracking-Suchbaum schneller abgebrochen werden, da die akkumulierten Constraints stärker, und daher eher unerfüllbar sind. Der erste Aufruf möge etwa den folgenden Suchbaum ergeben, wobei die Kanten für rekursive Aufrufe stehen, in den Knoten der jeweilige Wert von $Akku$ notiert ist, und die Verzweigungen die unterschiedlichen Wahlen für den abschließenden Constraint repräsentieren.



Die Suche unterhalb von C_{11} möge erfolglos, aber langwierig sein. Erst im rechten Ast werde die richtige Kombination von abschließenden Constraints gefunden. Wenn nun der Benutzer B so wählt, daß $B \sqcup C_{11}$ unerfüllbar ist, ergibt der Aufruf mit B den folgenden Baum:



Im linken Ast führt $B \sqcup C_{11}$ sofort zu **fail**. Der linke Teilbaum braucht also nicht weiter durchsucht zu werden.

Man bemerke, daß es für diese Diskussion unerheblich ist, ob der Automat jeweils versucht, den ganzen Baum auf einmal abzuschließen, oder – wie bei Tableaubeweisern üblich – sich zunächst für den Abschluß eines Astes entscheidet, bevor er an den verbleibenden Ästen arbeitet. Letzteres Verfahren läßt sich durch einen vom Automaten verwendeten Constraint A modellieren, der ähnlich wie der Benutzer-Constraint gehandhabt wird: In A werden die abschließenden Constraints gesammelt, die zu den derzeit gewählten Astabschlüssen gehören. Wenn der Beweis mit diesen Abschlüssen fehlschlägt, wird durch Backtracking auf eine frühere Kombination von Beweisbaum und A zurückgesetzt. Bei diesem Vorgehen entspricht A dem Akkumulator *Akku* des oben angegeben Algorithmus.

Die separate Änderbarkeit von Beweisbaum und Benutzer-Constraint B überträgt sich natürlich auf den „Automaten-Constraint“ A. Es ergibt sich also gegenüber den meisten automatischen Verfahren – die Substitutionen auf den Beweis anwenden, und diese daher auch zusammen mit dem Beweisbaum zurücksetzen müssen – eine größere Flexibilität für das Backtracking bei der Beweissuche. Dadurch werden Möglichkeiten für verfeinerte Beweis-Suchstrategien eröffnet.

Außer beim Abschlußtest sollte der Benutzer-Constraint bei der automatischen Anwendung der Gleichheitsregeln berücksichtigt werden, und zwar auf zweierlei Weise: Erstens sollte keine Formel erzeugt werden, deren Constraint C dem Benutzer-Constraint B widerspricht, für die also $C \sqcup B$ unerfüllbar ist. Zweitens sollte auch der Automat Gleichungen unterhalb einer Variablenposition anwenden, solange durch B für die Variable eine hinreichend konkrete Instantiierungen festgelegt wird. Sei etwa die Sequenz

$$f(a) \doteq b, p(f(X)) \vdash q(Y)$$

gegeben. Ein typisches Verfahren zur automatischen Gleichheitsbehandlung würde die Gleichung $f(a) \doteq b$ auf die Formel $p(f(X))$ anwenden, um eine neue Formel $p(b) \ll X{=}a$ zu erhalten. Andererseits würde es sicherlich nicht versuchen, eine Instantiierung für Y zu raten, um die Gleichung in dieser Richtung auf $q(Y)$ anzuwenden.

Ist nun zusätzlich der Benutzer-Constraint

$$B \;=\; (X = c) \sqcup (Y = g(f(a)))$$

gegeben, wird gerade das gegenteilige Verhalten erwünscht! Für die Präsentation wird $\mathrm{Pr}(B) = [X/c, Y/g(f(a))]$ verwendet, dem Benutzer wird also die Sequenz

$$f(a) \doteq b, p(f(c)) \vdash q(g(f(a)))$$

präsentiert. Die Anwendung der Gleichung auf die Formel $p(f(X))$ würde der vom Benutzer intendierten Instantiierung von X widersprechen. Sie wäre für den durch B eingeschränkten Suchraum überflüssig, da die resultierende Formel mit dem Constraint $C = (X = a)$ versehen würde, worauf $B \sqcup C$ unerfüllbar wäre. Die Gleichung sollte allerdings auf die Formel $q(Y)$ angewandt werden, und zwar mit einer Substitution $\mu = [Y/g(f(a))]$, da der Benutzer explizit diese Belegung verlangt hat. Das „Raten“ einer Instantiierung für Y wurde vom Benutzer übernommen.

Ein Verfahren zur automatischen Gleichheitsbehandlung kann recht einfach an diese Erfordernisse angepaßt werden. Betroffen ist der Teil des Verfahrens, der darüber entscheidet, welche Gleichung in welcher Richtung auf welche Formel angewandt wird. Diese Entscheidung geschieht anhand der Formeln in der Sequenz (bzw. auf dem Tableau-Ast), sowie deren Constraints und zusätzlichen Informationen, die die Richtung der Gleichheitsanwendung einschränken. Man kann davon ausgehen, daß dieses Verfahren nie eine Regel anwenden wird, die zu einer Formel mit unerfüllbarem Constraint führt, da eine solche Formel für den Abschluß nicht nützlich ist. Um nun die Entscheidung über die Anwendung von Gleichheitsregeln vom aktuellen Benutzer-Constraint abhängig zu machen, kann dem Verfahren statt der eigentlichen Sequenz Σ eine Sequenz $\Sigma \triangleleft B$ übergeben werden, die eine Konkretisierung von Σ bezüglich des Benutzer-Constraints B darstellt. Man definiert

$$(\Gamma \vdash \Delta) \triangleleft B := \{\mathrm{Pr}(B)(\gamma) \ll C \sqcup B \mid \gamma \ll C \in \Gamma\} \vdash \{\mathrm{Pr}(B)(\delta) \ll C \sqcup B \mid \delta \ll C \in \Delta\}.$$

Auf die Formeln der Sequenz wird also $\mathrm{Pr}(B)$ angewandt, wodurch die vom Benutzer angegebene Instantiierung vom Automaten nutzbar gemacht wird. Für die Entscheidung des Automaten macht es so keinen Unterschied mehr, ob ein Term in der tatsächlichen Sequenz vorkommt, oder nur vom Benutzer für die Instantiierung einer Metavariablen gewünscht wurde.

Die Constraints werden konjunktiv mit B verknüpft. Das hat zur Folge, daß (vom Benutzer) unerwünschte Regelanwendungen, die auf Σ zu einer Formel vom zweiten Typ führen, auf $\Sigma \triangleleft B$ direkt eine Formel mit dem unerfüllbaren Constraint $C \sqcup B$ ergeben, und daher vom Automaten nicht in Betracht gezogen werden.

Entscheidet sich der Automat für die Anwendung einer Gleichung $\mathrm{Pr}(B)(s \doteq t)$ auf eine Formel $\mathrm{Pr}(B)(\phi)$ in $\Sigma \triangleleft B$ mit der Substitution μ_0, so ist eine entsprechende Regel-Anwendung auch auf Σ möglich, im allgemeinen allerdings mit einer anderen Substitution μ. Für die Regel-Anwendung auf $\Sigma \triangleleft B$ gelte

$$\mu_0(\mathrm{Pr}(B)(\phi)) = [y/\mu_0(\mathrm{Pr}(B)(s))](\phi').$$

Dann gilt mit $\mu := \mu_0 \circ \mathrm{Pr}(B)$ offenbar

$$\mu(\phi) = [y/\mu(s)](\phi'), \tag{$*$}$$

woraus die Anwendbarkeit der Gleichung $s = t$ auf die Formel ϕ in Σ folgt.

Die Wahl von $\mu := \mu_0 \circ (B)$ kann unvorteilhaft sein, und zwar aus denselben Gründen, die auf Seite 94 für die interaktive Verwendung der Gleichheitsregeln angegeben wurde. Die einfachste Verfeinerung besteht wieder darin, sich auf die tatsächlich vorkommenden Metavariablen zu beschränken, und $\mu := \mu_0 \circ \mathrm{Pr}(B)|_{\mathrm{mv}(s) \cup \mathrm{mv}(\phi)}$ zu nehmen. Man sieht leicht, daß auch bei dieser Wahl $(*)$ gilt.

Es sollte noch erwähnt werden, daß die vorgestellte Methode zur Entscheidung über automatische Gleichheitsanwendungen *kein* Implementierungsvorschlag ist. Für eine tatsächliche Implementierung würde ein bestehendes Verfahren so geändert werden, daß es sich wie hier vorgeschlagen verhält. Die explizite Berechnung von $\Sigma \triangleleft B$ wäre in der Praxis natürlich zu ineffizient.

6.8. Zusammenfassung

Die Einschränkung einzelner Formeln von Sequenzen durch Constraints wurde aus zwei Richtungen motiviert, nämlich erstens anhand der Notwendigkeit von Einschränkungen an Gleichheitsregeln in einem Kalkül mit Metavariablen, und zweitens aus den Bedürfnissen effizienter Verfahren zur automatischen Gleichheitsbehandlung. Es wurden Kalküle mit Metavariablen und Constraints angegeben, die „geliftete“ Versionen der Grundkalküle in Kapitel 5 sind. Es wurde nachgewiesen, daß diese Kalküle korrekt und vollständig sind, und daß ein automatisch verwendbarer Kalkül mit ϵ-Termen exponentiell kürzere Beweise erlaubt als ein Kalkül mit δ^{++}-Skolemisierung.

Nach einer kurzen Diskussion von Reduktionsordnungen auf ϵ-Termen für die automatische Gleichheitsbehandlung wurde die Rolle eines globalen, vom Benutzer eingegebenen Constraints für die Beweispräsentation einerseits, und die Suchraumeinschränkung andererseits erläutert.

Im ganzen Kapitel wurde die verwendete Constraintsprache so abstrakt wie möglich gehalten, um die Diskussion von künftigen Entwicklungen im Bereich der constraintbasierten Gleichheitsbehandlung unabhängig zu halten.

7. Universeller Präfix statt universeller Variablen

Das Konzept von *universellen Variablen* wurde entwickelt, um in Tableaukalkülen die Anzahl benötigter γ-Regelanwendungen zu reduzieren. Jede Anwendung der γ-Regel führt nämlich eine neue freie Variable ein, für die eine Instantiierung gefunden werden muß. Je weniger freie Variablen in einem Beweis vorkommen, desto weniger muß wegen der Wahl ungeschickter Instantiierungen auf Backtracking zurückgegriffen werden. Die Verringerung der Anzahl zu instantiierender Variablen ist insbesondere auch für die automatische Gleichheitsbehandlung wichtig, worauf in Abschnitt 7.5 eingegangen wird.

Es soll nun zunächst die Definition und Verwendung von universellen Variablen in Tableaukalkülen erläutert werden. Es wird festegestellt werden, daß das Verfahren nicht ohne weiteres in den Kalkül für ein integriertes Beweissystem übernommen werden kann. Es wird jedoch eine Alternative mit vergleichbarer Stärke vorgestellt werden, die besser geeignet ist.

Schließlich soll erklärt werden, wie das Verfahren zur automatischen Gleichheitsbehandlung von Beckert [4] in den vorgestellten Rahmen einzubetten ist.

7.1. Universelle Variablen in Tableaukalkülen

Für Tableaukalküle werden universelle Variablen wie folgt definiert:[1]

> Sei ϕ eine Formel auf einem Ast B eines Tableaus T für $\Phi \subseteq \mathbf{Form}$. Es entstehe T', indem $\forall x.\phi$ für eine Variable $x \in \mathbf{V}$ zu B hinzugefügt wird. Dann heißt ϕ auf B *universell* bezüglich x, falls $T \models T'$, wobei T und T' mit der Disjunktion ihrer Äste identifiziert werden, welche sich wiederum als Konjunktion der auf ihnen befindlichen Formeln verstehen. [...]

Man bezeichnet oft auch die Variablen als universell bezüglich der Formel. Die Abschlußregeln für Tableaux mit universellen Variablen erlauben nun auf unterschiedlichen Ästen unterschiedliche Instantiierungen von freien Variablen, vorausgesetzt, diese sind bezüglich jener Formeln universell, die für den Abschluß verwendet werden.

Ein einfaches Beispiel soll dieses Konzept verdeutlichen: Da der Tableaukalkül ein Widerlegungskalkül ist, geht der Beweis von einer unerfüllbaren Formel aus. Sei

$$\phi = (\neg p(a) \vee \neg q(b)) \wedge \forall x.(p(x) \wedge q(x)).$$

[1] Es handelt sich hier um eine Übersetzung der Definition in [5].

Ein Tableaubeweis ohne universelle Variablen für ϕ hat folgende Form:

$$
\begin{array}{c}
(\neg p(a) \vee \neg q(b)) \wedge \forall x.(p(x) \wedge q(x)) \\
\neg p(a) \vee \neg q(b) \\
\forall x.(p(x) \wedge q(x)) \\
p(X_1) \wedge q(X_1) \\
p(X_1) \\
q(X_1) \\
\begin{array}{cc}
\neg p(a) & \neg q(b) \\
\bullet & p(X_2) \wedge q(X_2) \\
{[X_1/a]} & p(X_2) \\
 & q(X_2) \\
 & \bullet \\
 & {[X_2/b]}
\end{array}
\end{array}
$$

Die gestrichelten Pfeile entsprechen den *zwei* nötigen Instantiierungen der γ-Formel $\forall x.(p(x) \wedge q(x))$. In diesem Tableau sind die Formeln $p(X_1)$ und $q(X_1)$ bezüglich X_1 universell. Daher darf X_1 auf den beiden Ästen verschieden instantiiert werden:

$$
\begin{array}{c}
(\neg p(a) \vee \neg q(b)) \wedge \forall x.(p(x) \wedge q(x)) \\
\neg p(a) \vee \neg p(b) \\
\forall x.(p(x) \wedge q(x)) \\
p(X_1) \wedge q(X_1) \\
p(X_1) \\
q(X_1) \\
\begin{array}{cc}
\neg p(a) & \neg p(b) \\
\bullet & \bullet \\
{[X_1/a]} & {[X_1/b]}
\end{array}
\end{array}
$$

Einen besonders großen Vorteil bringt die Erkennung universeller Variablen bei universell quantifizierten Gleichungen wie $\forall x, y. f(x, y) = f(y, x)$ oder $\forall x, y, z. f(x, f(y, z)) = f(f(x, y), z)$. Diese brauchen nämlich nur einmal instantiiert werden, und können dann beliebig oft angewandt werden. Außerdem können auf solche universell quantifizierte Gleichungen Vervollständigungsverfahren angewandt werden, die zu einer viel effizienteren Gleichheitsbehandlung führen als Verfahren, die nur mit gewöhnlichen „rigiden" Variablen arbeiten.

Es ist unentscheidbar, ob eine Gleichung auf einem Tableau-Ast bezüglich einer gegebenen Variablen universell ist. Daher verwenden Implementierungen hinreichende Kriterien, die es erlauben, mit geringem Aufwand zumindest einen Teil der Variablen als universell zu identifizieren. In $_3TAP$ ist zum Beispiel folgendes Kriterium implementiert:

Eine Formel ϕ wird bezüglich einer Variablen X als universell erkannt, falls[2]

[2]Das Kriterium bezieht sich nicht auf die Gleichheitsregeln, da die Gleichheit in $_3TAP$durch einen Hintergrundbeweiser behandelt wird.

- ϕ durch eine γ-Regelanwendung entstanden ist, und X die neu eingeführte freie Variable ist, oder
- ϕ durch Anwendung einer α, γ oder δ-Regel aus einer bezüglich X universellen Formel entstanden ist, oder
- ϕ durch Anwendung einer β-Regel aus einer bezüglich X universellen Formel entstanden ist, und X in keiner anderen Teilformel $\beta_i \neq \phi$ frei vorkommt.

Mit diesem Kriterium wurden gute Erfahrungen gemacht, und es ist zweifelhaft, ob ein stärkeres Verfahren sich bewähren würde. Dieses könnte zwar gelegentlich zusätzliche freie Variablen entdecken, wäre aber sicherlich auch langsamer.

In dieser Version des Verfahrens werden universelle Variablen als Hintergrundinformation zu den Formeln mitgeführt. In einem Kalkül für ein integriertes Beweissystem ist dies nachteilhaft, da der Benutzer nicht auf den ersten Blick erkennen kann, welche Variablen universell sind. Eine visuelle Kennzeichnung wäre denkbar, etwa durch eine andere Farbgebung oder Schriftart für die Variablensymbole.[3] Es ist jedoch möglich, universelle Variablen zu simulieren, ohne einen neuen Typ von Variablen oder Formeln einführen zu müssen.

7.2. Beweisen mit universellem Präfix

Betrachten wir zunächst den $\mathcal{M}_E$-Beweis zu der Formel ϕ im Beispiel des letzten Abschnitts. Wir müssen zeigen, daß $\neg\phi$ allgemeingültig ist. Um das Wesentliche nicht durch aussagenlogische Beweisschritte zu verhüllen, fangen wir mit der Sequenz $\forall x.(p(x) \wedge q(x)) \vdash p(a) \wedge p(b)$ an.

$$\dfrac{p(X_1), q(X_1), \forall x.(p(x) \wedge q(x)) \vdash p(a) \qquad \dfrac{\dfrac{p(X_2), q(X_2), p(X_1), q(X_1) \vdash q(b)}{p(X_2) \wedge q(X_2), p(X_1), q(X_1) \vdash q(b)}\wedge l}{p(X_1), q(X_1), \forall x.(p(x) \wedge q(x)) \vdash q(b)}\forall l}{\dfrac{\dfrac{p(X_1), q(X_1), \forall x.(p(x) \wedge q(x)) \vdash p(a) \wedge p(b)}{p(X_1) \wedge q(X_1), \forall x.(p(x) \wedge q(x)) \vdash p(a) \wedge p(b)}\wedge l}{\forall x.(p(x) \wedge q(x)) \vdash p(a) \wedge p(b)}\forall l}\wedge r$$

Der Beweis entspricht genau dem ersten im vorigen Abschnitt angegebenen Tableaubeweis und ist mit der Substitution $[X_1/a, X_2/b]$ geschlossen.

Laut der oben gegebenen Definition besagt die Universalität der Formeln $p(X_1)$ und $q(X_1)$ des Beispiels bezüglich der Variable X_1, daß auch $\forall X_1.p(X_1)$, bzw. $\forall X_1.q(X_1)$ auf den Ast geschrieben werden dürften. In $\mathcal{M}_E$ ist das natürlich nicht möglich, da Metavariablen nicht gebunden werden. Statt dessen kann man aber die alte Variable x beibehalten: Man kann den zweiten Tableaubeweis in $\mathcal{M}_E$ nachbilden, indem man eine Regel

$$\frac{\forall x.\phi, \forall x.\psi, \Gamma \vdash \Delta}{\forall x.(\phi \wedge \psi), \Gamma \vdash \Delta}\ \forall\wedge l$$

[3] Bei einer Identifikation über die Farbgebung wäre allerdings ein Farbmonitor Voraussetzung zur Benutzung des Systems, und eine schriftliche Diskussion würde einen Farbdruck erfordern. Überdies könnte ein farbenblinder Benutzer das System nicht bedienen. Bei einer Trennung durch unterschiedliche Schrifttypen ist die Gefahr der Verwechslung zu groß.

verwendet. Daß diese Regel korrekt ist, ist leicht einzusehen. Mit ihr erhält man folgenden Beweisanfang:

$$\dfrac{\dfrac{\forall x.p(x), \forall x.q(x) \vdash p(a) \quad \forall x.p(x), \forall x.q(x) \vdash q(b)}{\forall x.p(x), \forall x.q(x) \vdash p(a) \wedge q(b)}\ \wedge r}{\forall x.(p(x) \wedge q(x)) \vdash p(a) \wedge q(b)}\ \forall\wedge l \tag{$*$}$$

Es gibt nun zwei Möglichkeiten, um diesen Beweis abzuschließen: Die eine Möglichkeit bestünde darin, in jedem Ast einmal die $\forall l$-Regel anzuwenden, und den Beweis

$$\dfrac{\dfrac{\dfrac{p(X_1), \forall x.p(x), \forall x.q(x) \vdash p(a)}{\forall x.p(x), \forall x.q(x) \vdash p(a)}\ \forall l \quad \dfrac{q(X_2), \forall x.p(x), \forall x.q(x) \vdash p(a)}{\forall x.p(x), \forall x.q(x) \vdash p(a)}\ \forall l}{\forall x.p(x), \forall x.q(x) \vdash p(a) \wedge q(b)}\ \wedge r}{\forall x.(p(x) \wedge q(x)) \vdash p(a) \wedge q(b)}\ \forall\wedge l$$

wie bisher mit der Substitution $[X_1/a, X_2/b]$ abzuschließen. Die Alternative folgt näher dem Verfahren im Tableaukalkül, wo die Abschlußregeln erweitert werden. Dies würde bedeuten, daß eine Sequenz der Form $\forall x.p(x) \vdash p(a)$ als Axiom angesehen wird, und der Beweis $(*)$ bereits geschlossen ist. In dieser Arbeit wird die letztere Alternative verfolgt, was jedoch erst später begründet werden wird. Zunächst sollen die Regeln des Kalküls erläutert werden.

7.2.1. Regeln mit universellem Präfix

Im Beispiel wurde die Universalität der Formeln $p(X_1)$ und $q(X_1)$ bezüglich der Variablen X_1 explizit gemacht, indem die Formeln mit einem Allquantor für eine entsprechende Objektvariable versehen wurde. Die vorgestellte Regel $\forall\wedge l$ implementiert das im letzten Abschnitt angegebene hinreichende Kriterium für universelle Variablen, für den Fall, daß erst eine γ-, und dann eine α-Regel angewandt wird, in diesem Fall $\forall l$ und $\wedge l$. Nach diesem Schema soll nun für alle anderen Regeln eine „universalisierte" Version gefunden werden. Das resultierende Regelsystem wird so stark sein wie ein Tableaukalkül, der *das angegebene hinreichende Kriterium* verwendet. Ein stärkeres Kriterium würde die Formulierung neuer Regeln erfordern. Es wurde jedoch bereits bemerkt, daß das angegebene Kriterium einerseits einfach und andererseits für die praktische Verwendung leistungsfähig genug ist. Es wird sich zeigen, daß die gewonnenen Regeln eine recht klare Struktur haben. Es wäre sicherlich nicht gut, diese zu opfern, um ein etwas leistungsfähigeres Kriterium zu implementieren. In ihrer vollen Allgemeinheit – d.h. mit der Möglichkeit, ein beliebiges hinreichendes Kriterium zu verwenden – wären universelle Variablen nur dann in den Kalkül einbettbar, wenn die Universalität als Hintergrundinformation verwaltet wird, was bereits ausgeschlossen wurde.

Im allgemeinen kann eine Formel bezüglich mehrerer Variablen universell sein. Deshalb werden mehrere Allquantoren vor den Formeln nötig sein. Für ein Tupel von Variablen $\bar{u} = (u_1, \ldots, u_k)$ soll die abkürzende Schreibweise

$$\forall \bar{u}.\phi \;:=\; \forall u_1 \ldots \forall u_k.\phi$$

verwendet werden. Dabei wird $\bar{u}$ gelegentlich als Menge aufgefaßt, um etwa Schnitt, Disjunktheit, etc. bequem ausdrücken zu können. $\forall\bar{u}$ wird als *universeller Präfix* der Formel ϕ bezeichnet, wenn diese auf der linken Seite einer Sequenz steht. Für Formeln auf der rechten Seite wird entsprechend $\exists\bar{u}.\phi$ definiert. Vor einer Formel auf der rechten Seite wird $\exists\bar{u}$ als universeller Präfix bezeichnet. Für die Zwecke dieses Abschnitts sollen Allquantoren zur Formelsprache hinzugenommen werden. Diese werden jedoch nur für den universellen Präfix von Formeln auf der linken Seite benötigt. Alternativ wäre es möglich, $\neg\exists\bar{u}\neg$ als universellen Präfix auf der rechten Seite zu verwenden, aber das würde die Diskussion eher verschleiern. Eine Implementierung wird ohnehin über beide Quantoren verfügen.

Die meisten Regeln der bisher vorgestellten Kalküle betreffen jeweils den obersten Junktor oder Quantor einer Formel. Sei zerlegen die Formel so, daß dieser Junktor verschwindet. So kommt man schließlich zu atomaren Formeln. Um das Verhalten von universellen Variablen zu imitieren, soll nun ein universeller Präfix bei dieser Zerlegung umgangen werden, und jeweils der oberste Junktor oder Quantor zerlegt werden, der nicht zum universellen Präfix der Formel gehört. So entstehen Formeln, die aus einer atomaren Formel (Prädikat oder Gleichung) mit einem universellem Präfix bestehen. Solche Formeln mögen künftig als *universelle Atome* bezeichnet werden. Auf solche Formeln werden sich die Abschlußregeln beziehen. Zur Unterscheidung von den $\mathcal{M}_E$-Regeln erhalten die neuen Regeln zunächst Namen, denen $\forall$, bzw. $\exists$ vorangestellt ist. Dies wird später, wenn Mißverständnisse ausgeschlossen sind, nicht mehr getan werden.

Zunächst soll die $\wedge l$-Regel betrachtet werden. Diese betraf bislang eine Formel $\phi \wedge \psi$ auf der linken Seite der Sequenz. Die neue Regel muß also eine Formel $\forall\bar{u}.(\phi\wedge\psi)$ zerlegen. Bekanntlich sind $\forall\bar{u}.(\phi \wedge \psi)$ und $(\forall\bar{u}.\phi) \;\wedge\; (\forall\bar{u}.\psi)$ logisch äquivalent. Daher bietet sich die Regel

$$\frac{\forall\bar{u}.\phi, \forall\bar{u}.\psi, \Gamma \vdash \Delta}{\forall\bar{u}.(\phi \wedge \psi), \Gamma \vdash \Delta}\ \forall\wedge l$$

an. Die Regelanwendung

$$\frac{\forall x.\forall y.p(x), \forall x.\forall y.q(y)}{\forall x.\forall y.(p(x) \wedge q(y))}\ \forall\wedge l$$

macht jedoch eine Schwäche dieser Regel deutlich: Es sollten Allquantoren für Variablen entfernt werden, die in den resultierenden Formeln nicht mehr vorkommen. Notiert man mit $\bar{u}\cap M$ das Tupel, welches entsteht, indem von $\bar{u}$ nur die Variablen in M beibehalten werden, so gelangt man zur Regel

$$\frac{\forall\bar{u}_1.\phi, \forall\bar{u}_2.\psi, \Gamma \vdash \Delta}{\forall\bar{u}.(\phi \wedge \psi), \Gamma \vdash \Delta}\ \forall\wedge l$$

$$\text{wobei } \bar{u}_1 := \bar{u} \cap \mathrm{fv}(\phi) \text{ und } \bar{u}_2 := \bar{u} \cap \mathrm{fv}(\psi).$$

Die $\wedge r$-Regel macht etwas mehr Probleme: Es wird eine Regel benötigt, die die Konjunktion in $\exists\bar{u}.(\phi \wedge \psi)$ auf der rechten Seite zerlegt. Die Regel

$$\frac{\Gamma \vdash \exists\bar{u}.\phi, \Delta \quad \Gamma \vdash \exists\bar{u}.\psi, \Delta}{\Gamma \vdash \exists\bar{u}.(\phi \wedge \psi), \Delta}\ \text{☠}$$

ist *nicht korrekt*, wovon sich der Leser leicht selber überzeugen kann.

In Tableau-Nomenklatur ist $\wedge r$ eine β-Regel, und das Kriterium besagt, daß die Teilformeln von β-Formeln bezüglich denjenigen Variablen universell sind, bezüglich denen die gesamte β-Formel universell ist, und die *in keiner anderen Teilformel vorkommen*. Universelle Variablen, die in beiden Teilformeln vorkommen, verlieren also ihre Universalität. Wenn man wieder darauf achtet, daß im universellen Präfix nur Variablen quantifiziert werden, die auch tatsächlich vorkommen, erhält man folgende Regel:

$$\frac{\Gamma \vdash \exists \bar{u}_1.[\bar{y}/\bar{Y}](\phi), \Delta \quad \Gamma \vdash \exists \bar{u}_2.[\bar{y}/\bar{Y}](\psi), \Delta}{\Gamma \vdash \exists \bar{u}.(\phi \wedge \psi), \Delta}$$

$$\begin{array}{ll} \text{wobei} & \bar{u}_1 := \bar{u} \cap (\mathrm{fv}(\phi) \setminus \mathrm{fv}(\psi)), \\ & \bar{u}_2 := \bar{u} \cap (\mathrm{fv}(\psi) \setminus \mathrm{fv}(\phi)), \\ & \bar{y} := \bar{u} \cap (\mathrm{fv}(\phi) \cap \mathrm{fv}(\psi)) \text{ und} \\ & \bar{Y} \text{ neue Metavariablen sind.} \end{array}$$

Diese Regel kann zum Beispiel wie folgt angewandt werden:

$$\frac{p(a,b), q(b,c) \vdash \exists x.p(x,Y) \quad p(a,b), q(b,c) \vdash \exists z.p(Y,z)}{p(a,b), q(b,c) \vdash \exists x.\exists y.\exists z.(p(x,y) \wedge q(y,z))}$$

Unter der Substitution $\sigma = [Y/b]$ werden die beiden entstandenen Sequenzen mit der später eingeführten erweiterten Abschlußregel als Axiome erkannt. Diese Regel hat allerdings immer noch einen Fehler. Man rufe sich die $\exists r$-Regel von $\mathcal{M}_E$ ins Gedächtnis:

$$\frac{\Gamma \vdash [x/X](\phi), \boxed{\exists x.\phi}, \Delta}{\Gamma \vdash \exists x.\phi, \Delta}\ \exists r$$

Es ist für die Vollständigkeit bekanntlich erforderlich, hier eine Kopie der ursprünglichen Formel beizubehalten, da noch weitere Instanzen von ihr benötigt werden können. Da die „universalisierte“ Version der Regel im allgemeinen Metavariablen einführt, und damit einige der Variablen instantiiert werden, muß auch hier die Formel $\exists \bar{u}.(\phi \wedge \psi)$ auf beide Äste kopiert werden. Die vollständige Regel lautet damit:

$$\frac{\Gamma \vdash \exists \bar{u}_1.[\bar{y}/\bar{Y}](\phi), \exists \bar{u}.(\phi \wedge \psi), \Delta \qquad \Gamma \vdash \exists \bar{u}_2.[\bar{y}/\bar{Y}](\psi), \exists \bar{u}.(\phi \wedge \psi), \Delta}{\Gamma \vdash \exists \bar{u}.(\phi \wedge \psi), \Delta}\ \exists\wedge r$$

$$\begin{array}{ll} \text{wobei} & \bar{u}_1 := \bar{u} \cap (\mathrm{fv}(\phi) \setminus \mathrm{fv}(\psi)), \\ & \bar{u}_2 := \bar{u} \cap (\mathrm{fv}(\psi) \setminus \mathrm{fv}(\phi)), \\ & \bar{y} := \bar{u} \cap (\mathrm{fv}(\phi) \cap \mathrm{fv}(\psi)) \text{ und} \\ & \bar{Y} \text{ neue Metavariablen sind.} \end{array}$$

Es ist bemerkenswert, daß von den Regeln, die der Automat anwenden muß, diese die einzige ist, die Metavariablen einführt. Dies wird aber verständlich, wenn man bedenkt, daß der Hauptzweck der Metavariablen (bzw. der freien Variablen im Tableaukalkül) die Kommunikation *zwischen* den Ästen ist. Die Anwendung der gleichen abschließenden Substitution σ auf alle Äste des Beweisbaums stellt sicher, daß die für γ-Formeln gewählte Instantiierung zwischen den Ästen konsistent ist. Im Lichte dieser Feststellung ist es denn auch natürlich, daß Metavariablen von *verzweigenden* Regeln eingeführt werden.

Ein Vergleich des hier vorgestellten Ansatzes mit dem Tableaukalkül ergibt, daß die Verwendung des universellen Präfix kürzere Beweise erlaubt: Hat man auf einem Tableau-Ast eine Formel $\forall x.(p(x) \wedge (q(x) \vee r(x)))$, und benötigt ein Beweis zwei Instanzen der Disjunktion $q(x) \vee r(x)$, so wird eine zweite Instanz der gesamten Formel erzeugt, worauf abermals eine α-Regelanwendung nötig ist. Mit den hier vorgestellten Regeln wird in einem solchen Fall nur eine zweite Instanz der Disjunktion erzeugt. Von $\forall x.p(x)$ werden keine Instanzen benötigt, da es sich bereits um ein universelles Atom handelt. Dieser Effekt wurde bislang nicht quantitativ untersucht.

Wenn ϕ und ψ keine gemeinsamen freien Variablen haben, braucht die $\exists\wedge r$-Regel keine neuen Metavariablen einzuführen. Da die Regel in diesem Fall keine Instantiierung vornimmt, ist auch das Kopieren der ursprünglichen Formel unnötig. Man kann eine spezielle Version der Regel für diesen Fall einführen, nämlich:

$$\frac{\Gamma \vdash \exists \bar{u}_1.\phi, \Delta \qquad \Gamma \vdash \exists \bar{u}_2.\psi, \Delta}{\Gamma \vdash \exists \bar{u}.(\phi \wedge \psi), \Delta}\ \exists\wedge r'$$

$$\begin{array}{rl} \text{falls} & \mathrm{fv}(\phi) \cap \mathrm{fv}(\psi) = \emptyset, \\ \text{wobei} & \bar{u}_1 := \bar{u} \cap \mathrm{fv}(\phi), \text{ und} \\ & \bar{u}_2 := \bar{u} \cap \mathrm{fv}(\psi). \end{array}$$

Der automatische Beweiser sollte diese Regel anwenden, wann immer dies möglich ist, da durch sie unnötige Mehrfach-Instantiierungen vermieden werden.

Die beiden Negationsregeln bringen keine Überraschungen: sie beruhen auf der Tatsache, daß $\forall \bar{u}.\neg\phi$ und $\neg\exists \bar{u}.\phi$, bzw. $\exists \bar{u}.\neg\phi$ und $\neg\forall \bar{u}.\phi$ logisch äquivalent sind. Sie lauten

$$\frac{\Gamma \vdash \exists \bar{u}.\phi, \Delta}{\forall \bar{u}.\neg\phi, \Gamma \vdash \Delta}\ \forall\neg l \qquad\qquad \frac{\forall \bar{u}.\phi, \Gamma \vdash \Delta}{\Gamma \vdash \exists \bar{u}.\neg\phi, \Delta}\ \exists\neg r$$

An diesen Regeln wird besonders deutlich, warum der universelle Präfix auf der rechten Seite aus Existenzquantoren bestehen muß.

Die $\exists l$-Regel von $\mathcal{M}_E$ führt einen ϵ-Term ein. Welche Formel muß der eingeführte ϵ-Term in einer $\forall\exists l$-Regel enthalten? Die Entsprechung zum Tableaukalkül mit universellen Variablen deutet darauf hin, den universellen Präfix zu ignorieren, und wie gewohnt zu skolemisieren. Die neue Formel ist bezüglich derselben Variablen universell wie die ursprüngliche. Man kommt so zu folgender Regel:

$$\frac{\forall \bar{u}.[x/\epsilon x.\phi](\phi), \Gamma \vdash \Delta}{\forall \bar{u}.\exists x.\phi, \Gamma \vdash \Delta}\ \forall\exists l$$

Diese Regel ist korrekt: $\epsilon x.\phi$ kann freie Variablen aus $\bar{u}$ enthalten. Wenn $\forall \bar{u}.\exists x.\phi$ gilt, existiert für jede Belegung dieser Variablen ein x, so daß ϕ erfüllt wird. Die Auswahleigenschaft stellt sicher, daß dann $\epsilon x.\phi$ für jede Belegung der Variablen $\bar{u}$ zu einem solchen x ausgewertet wird.

Was geschieht nun mit der $\exists r$-Regel? Sie müßte sich auf Formeln der Form $\exists \bar{u}.\exists x.\phi$ beziehen. Dann kann aber $\exists x$ ebenfalls zum universellen Präfix gezählt werden. Offenbar ist eine $\exists r$-Regel dank der stärkeren Abschlußregeln gar nicht nötig. Die bisherigen Regeln decken alle möglichen prädikatenlogischen Formeln ab. Übrig bleiben lediglich die universellen Atome, für deren Behandlung die Abschlußregeln zuständig sind.

Trotzdem soll eine $\exists\exists r$-Regel in den Kalkül aufgenommen werden, und zwar für die interaktive Verwendung: Es ist zu erwarten, daß die universellen Variablen mitunter das Verständnis der Beweise erschweren.[4] Benötigt der Benutzer eine bestimmte Instanz einer Formel, kann er mit Hilfe der $\exists\exists r$-Regel selber die Instantiierung vornehmen. Dabei soll es möglich sein, einen Teil der Variablen im universellen Präfix einer Formel zu instantiieren,

[4] Es ist nicht anzunehmen, daß dies oft geschieht: Beweisaufgaben sind meist „so universell wie möglich" gestellt, und die meisten verwendeten Formeln haben die Form $\forall \bar{u}(\phi \rightarrow \psi)$. Diese Formeln erfordern eine Aufspaltung des Beweises, und die Universalität der Variablen geht verloren.

und den Rest universell zu belassen. Man erhält so die folgende Regel:

$$\frac{\Gamma \vdash \exists \bar{u}'.[\bar{y}/\bar{Y}](\phi), \exists \bar{u}.\phi, \Delta}{\Gamma \vdash \exists \bar{u}.\phi, \Delta} \;\exists\exists r$$

$$\text{wobei} \quad \begin{array}{l} \bar{y} \subseteq \bar{u}, \\ \bar{Y} \text{ neue Metavariablen, und} \\ \bar{u}' := \bar{u} \setminus \bar{y} \end{array}$$

Da Allquantoren im universellen Präfix in die Formelsprache aufgenommen wurden, ist auch eine duale Regel für Allquantoren nötig, die in Abb. 7.1 angeführt ist.

Am kompliziertesten ist die Situation bei den Gleichheitsregeln. Ihnen wird daher ein eigener Abschnitt gewidmet. Zunächst sollen die restlichen Regeln von $\mathcal{M}_E$ betrachtet werden, die keine größeren Probleme bereiten.

Die ϵ-*cut*-Regel von $\mathcal{M}_E$ lautet

$$\frac{\Gamma \vdash \exists x.\phi, \Delta \quad [x/\epsilon x.\phi](\phi), \Gamma \vdash \Delta}{\Gamma \vdash \Delta} \;\epsilon\text{-}cut$$

$$\text{wobei } \epsilon x.\phi \in \mathbf{Term}^*.$$

Man erinnere sich, daß hier *ganze* Terme $\epsilon x.\phi$, und nicht nur Grundterme erlaubt werden, da der Automat diese Regel für ϵ-Terme anwendet, die in atomaren Formeln vorkommen. In solchen Termen können natürlich auch Metavariablen auftreten. Die Verwendung von ganzen Termen ermöglicht es dem Automaten, die Regel anzuwenden, ohne diese Metavariablen zu instantiieren.

Eine ganz ähnliche Argumentation führt zu der „universalisierten“ ϵ-*cut*-Regel. Da in dem neuen Kalkül atomare Formeln eventuell mit einem universellen Präfix versehen sein können, muß die ϵ-*cut*-Regel nun auch mit ϵ-Termen umgehen können, in denen universelle Variablen vorkommen. In Anlehnung an die Herleitung der ϵ-*cut*-Regel in Abschnitt 5.6.1 betrachte man folgende Ableitung:

$$\frac{\Gamma \vdash \forall \bar{u}.\exists x.\phi, \Delta \qquad \dfrac{\forall \bar{u}.[x/\epsilon x.\phi](\phi), \Gamma \vdash \Delta}{\forall \bar{u}.\exists x.\phi, \Gamma \vdash \Delta}\;\forall\exists l}{\Gamma \vdash \Delta} \;cut$$

Damit ist die Regel

$$\frac{\Gamma \vdash \forall \bar{u}.\exists x.\phi, \Delta \quad \forall \bar{u}.[x/\epsilon x.\phi](\phi), \Gamma \vdash \Delta}{\Gamma \vdash \Delta} \;\forall\epsilon\text{-}cut$$

$$\text{wobei } \forall \bar{u}.\exists x.\phi \in \mathbf{Form}^*.$$

validiert! Der automatische Beweiser wird diese Regel für ϵ-Terme $\epsilon x.\phi$ anwenden, die in universellen Atomen vorkommen, und $\bar{u} := \mathrm{fv}(\phi)$ setzen.

Die Anpassung der *ext*-Regel ist – wie nicht anders zu erwarten – etwas umfangreich, bringt aber nichts prinzipiell neues. Die Regel lautet:

$$\frac{\Gamma \vdash \forall \bar{u}.([x/\mathbf{c}](\phi) \leftrightarrow [y/\mathbf{c}])(\psi), \Delta \quad \forall \bar{u}.(\epsilon x.\phi \doteq \epsilon y.\psi), \Gamma \vdash \Delta}{\Gamma \vdash \Delta} \;ext$$

$$\text{wobei} \quad \begin{array}{l} \forall \bar{u}.(\epsilon x.\phi \doteq \epsilon y.\psi) \in \mathbf{Form}^*, \\ \mathbf{c} := \epsilon z.\neg\Big([x/z](\phi) \leftrightarrow [y/z](\psi)\Big), \text{ und} \\ z \in \mathbf{V} \setminus (\mathrm{bv}(\phi) \cup \mathrm{bv}(\psi) \cup \mathrm{fv}(\phi) \cup \mathrm{fv}(\psi)). \end{array}$$

Im linken Ast muß gezeigt werden, daß unter jeder Belegung der universellen Variablen $\bar{u}$ die Extensionen von $\epsilon x.\phi$ und $\epsilon y.\psi$ gleich sind. Wegen der Extensionalität folgt dann für jede Belegung, daß die beiden ϵ-Terme gleich ausgewertet werden.

Es ist etwas fraglich, wie sinnvoll diese Erweiterung bei der *ext*-Regel ist, da diese ohnehin nur vom Benutzer verwendet wird, der meistens weiß, welche Instanzen er benötigt. Allerdings richtet die Verallgemeinerung keinen Schaden an, da der Benutzer, wenn er keinen universellen Präfix angibt, wieder genau die alte Regel erhält.

Die Abschwächungsregeln und die *cut*-Regel können offenbar ohne Änderungen übernommen werden. Somit verbleiben nur noch die Gleichheitsregeln.

7.2.2. Gleichheitsregeln mit universellem Präfix

Bei den Gleichheitsregeln treten einige Schwierigkeiten auf, die damit zusammenhängen, daß die zur Anwendung einer Gleichheit nötige Substitution auch die universellen Variablen betreffen kann. Welche Fälle auftreten können, soll zunächst anhand einiger Beispiele gezeigt werden.

Gegeben sei die Sequenz

$$\forall u.f(u,b) \doteq c, \forall v.p(f(a,v)) \vdash,$$

die noch keine Metavariablen enthält. Betrachtet man die universellen Variablen u und v wie im Tableaukalkül als freie Variablen, ist die $\overset{\rightrightarrows}{=}l$-Regel mit der Substitution $[u/a, v/b]$ anwendbar, und man erhält die Formel $p(c)$. Die Universalität von u und v bewirkt, daß diese Formel nicht mit einem Constraint versehen zu werden braucht: Constraints sind Instantiierungs-Einschränkungen, und die Universalität von u und v stellt sicher, daß die Instantiierung möglich ist. Man erhält also eine Regelanwendung

$$\frac{p(c), \ldots \vdash}{\forall u.f(u,b) \doteq c, \forall v.p(f(a,v)) \vdash} \; \forall\overset{\rightrightarrows}{=}l$$

Es bereitet keine Probleme, in der Instantiierung für die universellen Variablen auch Metavariablen zuzulassen. Man kann etwa im Beispiel a und b durch X und Y ersetzen, und erhält mit der Substitution $[u/X, v/Y]$ die Regelanwendung

$$\frac{p(c), \ldots \vdash}{\forall u.f(u,Y) \doteq c, \forall v.p(f(X,v)) \vdash} \; \forall\overset{\rightrightarrows}{=}l$$

Im allgemeinen muß die Substitution natürlich auch auf die Metavariablen wirken, wobei dann Constraints nötig werden. Dies ist etwa in folgender Anwendung der Fall:

$$\frac{p(c) \ll Y = b, \ldots \vdash}{\forall u.f(u,Y) \doteq c, p(f(X,b)) \vdash} \; \forall\overset{\rightrightarrows}{=}l$$

Hier war die Substitution $[u/X, Y/b]$ nötig. Man sieht, daß der Constraint aus dem Teil der Substitution gebildet wird, der auf den Metavariablen wirkt. Doch was geschieht, wenn eine Metavariable durch die Substitution auf einen Term abgebildet wird, der eine universelle Variable enthält? Man betrachte folgende Sequenz:

$$f(X) \doteq c, \forall v.p(f(g(v))) \vdash$$

Die naheliegende Substitution lautet $[X/g(v)]$. Aber dann müßte die Formel $p(c)$ den Constraint $X = g(v)$ erhalten, was keinen Sinn macht, da v keine Metavariable ist, sondern eine Objektvariable, die in der Formel $\forall v.p(f(g(v)))$ gebunden ist, und daher nur dort eine Semantik hat.

Die Lösung besteht darin, für den Constraint eine neue Metavariable V einzuführen, die im Constraint die Rolle von v übernimmt. Man arbeitet also mit der Substitution $[v/V, X/g(V)]$. Mit dieser Substitution läßt sich die Regel wie folgt anwenden:

$$\frac{p(c) \ll X = g(V), \ldots}{f(X) \doteq c, \forall v.p(f(g(v))) \vdash} \; \forall\overset{\rightarrow}{=}l$$

Was bedeutet dieser Constraint? V kommt sonst noch nirgendwo vor. Daher schränkt $X = g(V)$ die Instantiierung von X auf Terme der Form „g von irgendetwas“ ein. Und genau für solche Terme kann natürlich auch die Gleichung angewandt werden. Wir werden in Lemma 7.1 zeigen, daß die Substitution durch die Einführung neuer Metavariablen immer so abgeändert werden kann, daß Metavariablen nur ganze Terme zugewiesen werden. Die auf Seite 94 angesprochenen Probleme bei der Einführung von neuen Metavariablen ergeben sich hier übrigens nicht, da die Metavariable nur im Constraint auftritt. Es ist also nicht erforderlich, daß $\mathrm{Pr}(B)$ für sie eine adäquate Präsentation liefert.

Bei den bisherigen Beispielen enthielt die von der Regel erzeugte neue Formel keine universellen Variablen mehr. Dies lag daran, daß alle universellen Variablen von der Substitution zu Grundtermen, bzw. ganzen Termen konkretisiert wurden. Im allgemeinen muß dies nicht nötig sein. Man betrachte etwa die folgende Regelanwendung, die offensichtlich korrekt ist:

$$\frac{\forall v.p(b, v), \ldots}{\forall u.f(u) \doteq b, \forall v.p(f(a), v) \vdash} \; \forall\overset{\rightarrow}{=}l$$

Hier genügte die Substitution $[u/a]$. Daher bleibt v universell. Es wird sich zeigen, daß *alle* Objektvariablen, die in der erzeugten Formel noch frei vorkommen, universell sind.

Um in der Definition der Regel, und in Beweisen über sie, die Instantiierung der Variablen im universellen Präfix von derjenigen der Metavariablen besser zu trennen, sollen nun *zwei* Substitutionen μ und ν verwendet werden. Dabei übernimmt μ dieselbe Rolle wie in $\mathcal{M}_E$, während ν nur auf den universellen Variablen wirkt. Die Regel lautet wie folgt:

$$\frac{\forall \bar{w}.[y/(\mu \circ \nu)(t)](\phi') \ll C', \forall \bar{u}.s \doteq t \ll C_1, \forall \bar{v}.\phi \ll C_2, \Gamma \vdash \Delta}{\forall \bar{u}.s \doteq t \ll C_1, \forall \bar{v}.\phi \ll C_2, \Gamma \vdash \Delta} \; \forall\overset{\rightarrow}{=}l$$

$$\begin{aligned}
\text{wobei}\quad & \bar{u},\ \bar{v} \text{ und } \mathrm{bv}(s \doteq t) \cup \mathrm{bv}(\phi) \text{ paarw. disjunkt,}\\
& (\mu \circ \nu)(\phi) = [y/(\mu \circ \nu)(s)](\phi'),\\
& \mu \in \mathbf{Sub}^*,\\
& \nu \in \mathbf{Sub}(\bar{u} \cup \bar{v}) \text{ idempotent, } \mathrm{fv}(\nu(\bar{u} \cup \bar{v})) \subseteq (\bar{u} \cup \bar{v})\\
& \bar{w} := \mathrm{fv}([y/(\mu \circ \nu)(t)](\phi')) \text{ und}\\
& C' := C_1 \sqcup C_2 \sqcup \mathrm{Con}(\mu)
\end{aligned}$$

Die erste Voraussetzung ermöglicht es, *eine* Substitution ν für *beide* universellen Präfixe zu verwenden, und vermeidet außerdem Kollisionen mit Variablen, die im Innern der Formeln/Terme gebunden werden. Man sieht unmittelbar, daß gebundene Umbenennung der Variablen im universellen Präfix trotz der Vorbehalte in 5.7 immer möglich ist. Die Bedingung $\mathrm{fv}(\nu(\bar{u} \cup \bar{v})) \subseteq (\bar{u} \cup \bar{v})$ bedeutet, daß $\mathrm{fv}(\nu(x)) \subseteq (\bar{u} \cup \bar{v})$ für alle $x \in \bar{u} \cap \bar{v}$. ν kann also keine neuen Objektvariablen einführen.

Insgesamt wird die Substitution $\mu \circ \nu$ angewandt. Da ν Metavariablen unverändert läßt, und μ nur ganze Terme liefert, bildet auch $\mu \circ \nu$ Metavariablen nur auf ganze Terme ab. Die Bedingung $\mathrm{fv}(\nu(x)) \subseteq (\bar{u} \cup \bar{v})$ sorgt dafür, daß auch ν keine Variablen einführen kann, die mit anderen gebundenen Variablen kollidieren. Auch diese Bedingung kann immer durch Umbenennung von Variablen erfüllt werden.

Die anderen drei Gleichheitsregeln sind entsprechend definiert. Man mache sich insbesondere klar, daß die Regelanwendung

$$\frac{\forall x.a \doteq f(x) \vdash \exists x.p(f(x)), p(a)}{\forall x.a \doteq f(x) \vdash p(a)} \; \exists \overrightarrow{\doteq} r$$

korrekt ist: Wenn $p(f(x))$ für ein x gezeigt wurde, folgt daraus mit der universellen Gleichung $p(a)$. Solche Gleichungsanwendungen, in denen in der Formel eine neue universelle Variable hinzukommt, sind etwas verwirrend. Sie treten jedoch in der Praxis nur selten auf, da Gleichungen meistens so angewandt werden, daß Terme einfacher werden, also insbesondere Variablen entfernt werden.

Diese Regeln kann ein System dem Benutzer natürlich noch weniger in dieser Form anbieten als die $\mathcal{M}_E$-Regeln! Trotz der größeren Komplexität treten jedoch gegenüber diesen keine neuen Probleme auf, d.h. die in Abschnitt 6.6 beschriebenen Methoden zur Präsentation der Gleichheitsregeln können auch auf die Regeln mit universellen Variablen angewandt werden. Das System kann die zur Anwendung der Gleichung nötige Substitution wie bereits diskutiert ermitteln, wobei es den universellen Präfix ignoriert, und die universellen Variablen wie Metavariablen behandelt. Das folgende Lemma besagt, daß zu einer so gewonnenen Substitution μ_0 Substitutionen μ und ν existieren, die die Anwendung der universalisierten Gleichheitsregel an derselben Position erlauben.

7.1 Lemma ($\forall \doteq$-Regeln mit *einer* Substitution)
Seien $\forall \bar{u}.s \doteq t$, $\forall \bar{v}.\phi$, $\phi' \in \mathbf{Form}^*$ und $\mu_0 \in \mathbf{Sub}(\bar{u} \cup \bar{v} \cup \mathbf{M})$ idempotent mit $\mathrm{fv}(\mu_0(\bar{u} \cup \bar{v})) \subseteq \bar{u} \cup \bar{v}$ und

$$\mu_0(\phi) = [y/\mu_0(s)](\phi'). \tag{1}$$

Dann existieren $\mu \in \mathbf{Sub}^*$ und $\nu, \xi \in \mathbf{Sub}(\bar{u} \cup \bar{v})$ mit ν idempotent, $\mathrm{fv}(\nu(\bar{u} \cup \bar{v})) \subseteq (\bar{u} \cup \bar{v})$ für alle $x \in \bar{u} \cap \bar{v}$ und

$$(\mu \circ \nu)(\phi) = [y/(\mu \circ \nu)(s)](\xi(\phi')) \tag{2}$$

Beweis. Wir verwenden ohne Beweis folgende Charakterisierung:

> Eine Substitution τ ist genau dann idempotent, wenn $\mathrm{fv}(\tau(x)) \cap \mathrm{dom}(\tau) = \emptyset$ für alle $x \in \mathrm{dom}(\tau)$.

Sei $A := \{u \in \mathbf{V} \mid u \in \mathrm{fv}(\mu_0(X)) \text{ für ein } X \in \mathbf{M}\}$. Da μ_0 idempotent, gilt $\mu_0(u) = u$ für $u \in A$. Sei $U_u \in \mathbf{M} \setminus \mathrm{dom}(\mu_0)$ eine neue Metavariable für jedes $u \in A$. Definiere:

$$\begin{aligned} \xi &:= [u/U_u \mid u \in A] \\ \mu &:= \xi \circ \mu_0|_{\mathbf{M}} \\ \nu &:= \xi \circ \mu_0|_{\mathbf{V}} \end{aligned}$$

Damit ist $(\mu \circ \nu)(X) = (\xi \circ \mu_0|_{\mathbf{M}} \circ \xi \circ \mu_0|_{\mathbf{V}})(X) = \xi(\mu_0(X))$ für $X \in \mathbf{M}$. Für $u \in A$ gilt $(\mu \circ \nu)(u) = U_u = \xi(\mu_0(u))$. Für $u \in \mathbf{V} \setminus A$ gilt schließlich: $(\mu \circ \nu)(u) = (\xi \circ \mu_0|_{\mathbf{V}})(u) = \xi(\mu_0(u))$. Also ist $\mu \circ \nu = \xi \circ \mu_0$. Damit folgt:

$$(\mu \circ \nu)(\phi) = \xi(\mu_0(\phi)) \overset{(1)}{=} \xi([y/\mu_0(s)](\phi')) = [y/(\xi \circ \mu_0)(s)](\xi(\phi')) = [y/(\mu \circ \nu)(s)](\xi(\phi')).$$

μ ist idempotent, da $U_u \notin \mathrm{dom}(\mu_0)$ und $\mu_0|_{\mathbf{M}}$ idempotent, für ν entsprechend. $\mathrm{fv}(\nu(x)) \subseteq (\bar{u} \cup \bar{v})$ folgt unmittelbar aus der entsprechenden Eigenschaft für μ_0. ∎

Man bemerke, daß $\xi(\phi')$ die Rolle von ϕ' in der universalisierten Gleichungsanwendung übernimmt. Dadurch ist sichergestellt, daß die Gleichung auch an der gewünschten, mit y markierten Stelle anwendbar ist.

Bei Verwendung universeller Variablen sind Gleichheitsregeln denkbar, die noch stärker als die angegebenen sind. Wegen der Universalität von u in der Gleichung wäre etwa die folgende Regelanwendung möglich:

$$\frac{\exists x.p(c,x) \vdash}{\forall u.f(u) = c, \exists x.p(f(x),x) \vdash}$$

Hier wurde also zugelassen, daß u mit der gebundenen Variable x instantiiert wird. Solche Regelanwendungen wurden aus zwei Gründen nicht betrachtet: Erstens sind die Bedingungen, unter denen solche Anwendungen erlaubt, sind nicht ganz einfach. Insbesondere sind sie innerhalb von ϵ-Termen nur bei extensionaler Semantik korrekt. Zweitens wird der Nachweis der Korrektheit viel komplizierter, da hier nicht mehr mit der Substitutivität argumentiert werden kann.

7.2.3. Beweisabschluß mit universellem Präfix

Mit diesen Regeln soll nun ein Kalkül $\mathcal{U}_E$ definiert werden. Die Regeln sind in Abb. 7.1 noch einmal zusammengefaßt. Bei den meisten Regeln wurden die Constraints weggelassen, um die Darstellung nicht noch unübersichtlicher zu machen. Mit Ausnahme der Gleichheitsregeln vererben alle Regeln die Constraints unverändert an die neuen Formeln.

Wie bereits erwähnt, existieren zwei Alternativen, um den Beweisabschluß zu definieren:

- Man wendet $\forall l$, bzw. $\exists r$ als letzte Regel auf jedem Ast an, um aus den universellen Variablen Metavariablen zu machen. Der Abschluß kann dann genauso definiert werden wie für $\mathcal{M}_E$.

- Man definiert neue Abschlußbedingungen, die einen universellen Präfix an den für den Abschluß verwendeten Formeln berücksichtigen.

Wir werden für den Kalkül $\mathcal{U}_E$ die zweite Alternative verfolgen. Mit der ersten Alternative müßte nämlich die abschließende Anwendung von $\forall l$, bzw. $\exists r$ jeweils wieder rückgängig gemacht werden, wenn ein Ast erweitert wird. Bei interaktiver Verwendung müßte dies entweder bei jedem Beweisschritt geschehen, oder der Benutzer müßte durch diese Regelanwendung explizit angeben, daß ein Ast an einer bestimmten Stelle geschlossen sein soll. Dies war bisher nicht nötig. Für die automatische Verwendung müßte die abschließende Regelanwendung nach jedem Schritt geschehen, um zu überprüfen, ob der Beweis geschlossen ist. Falls er es nicht ist, müßte die Regelanwendung vor einer Erweiterung des Beweisbaums wieder gelöscht werden, da sie dann nicht nötig ist, sondern nur eine überflüssige Instanz einführt. All dies entfällt mit einer stärkeren Abschlußbedingung.

Man möchte nun $\mathcal{U}_E$-Axiome so definieren, daß die Abschlußbedingung äquivalent zu der ist, die sich mit der ersten Alternative ergeben hätte. Die Definition muß also dafür sorgen, daß genau diejenigen Beweise abgeschlossen sind, die mit der $\mathcal{M}_E$-Bedingung

Abbildung 7.1.: Beweisregeln des Kalküls $\mathcal{U}_E$

$$\frac{\forall \bar{u}_1.\phi, \forall \bar{u}_2.\psi, \Gamma \vdash \Delta}{\forall \bar{u}.(\phi \wedge \psi), \Gamma \vdash \Delta}\ \wedge l \qquad \frac{\Gamma \vdash \exists \bar{u}.\phi, \Delta}{\forall \bar{u}.\neg\phi, \Gamma \vdash \Delta}\ \neg l \qquad \frac{\forall \bar{u}.\phi, \Gamma \vdash \Delta}{\Gamma \vdash \exists \bar{u}.\neg\phi, \Delta}\ \neg r \qquad \frac{\forall \bar{u}.[x/\epsilon x.\phi](\phi), \Gamma \vdash \Delta}{\forall \bar{u}.\exists x.\phi, \Gamma \vdash \Delta}\ \exists l$$

wobei $\bar{u}_1 := \bar{u} \cap \mathrm{fv}(\phi)$, und $\bar{u}_2 := \bar{u} \cap \mathrm{fv}(\psi)$.

$$\frac{\Gamma \vdash \exists \bar{u}_1.[\bar{y}/\bar{Y}](\phi), \exists \bar{u}.(\phi \wedge \psi), \Delta \qquad \Gamma \vdash \exists \bar{u}_2.[\bar{y}/\bar{Y}](\psi), \exists \bar{u}.(\phi \wedge \psi), \Delta}{\Gamma \vdash \exists \bar{u}.(\phi \wedge \psi), \Delta}\ \wedge r$$

wobei $\bar{u}_1 := \bar{u} \cap (\mathrm{fv}(\phi) \setminus \mathrm{fv}(\psi))$, $\bar{u}_2 := \bar{u} \cap (\mathrm{fv}(\psi) \setminus \mathrm{fv}(\phi))$,
$\bar{y} := \bar{u} \cap (\mathrm{fv}(\phi) \cap \mathrm{fv}(\psi))$, und
$\bar{Y}$ neue Metavariablen sind.

$$\frac{\Gamma \vdash \exists \bar{u}_1.\phi, \Delta \qquad \Gamma \vdash \exists \bar{u}_2.\psi, \Delta}{\Gamma \vdash \exists \bar{u}.(\phi \wedge \psi), \Delta}\ \wedge r'$$

falls $\mathrm{fv}(\phi) \cap \mathrm{fv}(\psi) = \emptyset$,
wobei $\bar{u}_1 := \bar{u} \cap \mathrm{fv}(\phi)$, und
$\bar{u}_2 := \bar{u} \cap \mathrm{fv}(\psi)$.

$$\frac{\Gamma \vdash \forall \bar{u}.\exists x.\phi, \Delta \qquad \forall \bar{u}.[x/\epsilon x.\phi](\phi), \Gamma \vdash \Delta}{\Gamma \vdash \Delta}\ \epsilon\text{-}cut$$

wobei $\forall \bar{u}.\exists x.\phi \in \mathbf{Form}^*$.

$$\frac{\Gamma \vdash \forall \bar{u}.([x/\mathbf{c}](\phi) \leftrightarrow [y/\mathbf{c}])(\psi), \Delta \qquad \forall \bar{u}.(\epsilon x.\phi \doteq \epsilon y.\psi), \Gamma \vdash \Delta}{\Gamma \vdash \Delta}\ ext$$

wobei $\forall \bar{u}.(\epsilon x.\phi \doteq \epsilon y.\psi \in \mathbf{Form}^*$, $\mathbf{c} := \epsilon z.\neg\Big([x/z](\phi) \leftrightarrow [y/z](\psi)\Big)$, und
$z \in \mathbf{V} \setminus (\mathrm{bv}(\phi) \cup \mathrm{bv}(\psi) \cup \mathrm{fv}(\phi) \cup \mathrm{fv}(\psi))$.

$$\frac{\Gamma \vdash \exists \bar{u}'.[\bar{y}/\bar{Y}]\phi), \exists \bar{u}.\phi, \Delta}{\Gamma \vdash \exists \bar{u}.\phi, \Delta}\ \exists r$$

wobei $\bar{y} \subseteq \bar{u}$,
$\bar{Y}$ neue Metavariablen, und
$\bar{u}' := \bar{u} \setminus \bar{y}$

$$\frac{\forall \bar{u}'.[\bar{y}/\bar{Y}](\phi), \forall \bar{u}.\phi, \Gamma \vdash \Delta}{\forall \bar{u}.\phi, \Gamma \vdash \Delta}\ \forall l$$

wobei $\bar{y} \subseteq \bar{u}$,
$\bar{Y}$ neue Metavariablen, und
$\bar{u}' := \bar{u} \setminus \bar{y}$

$$\frac{\Gamma \vdash \phi, \Delta \qquad \Gamma, \phi \vdash, \Delta}{\Gamma \vdash \Delta}\ cut$$

wobei $\phi \in \mathbf{Form}^*$.

$$\frac{\Gamma \vdash \Delta}{\phi \ll C, \Gamma \vdash \Delta}\ weak\text{-}l \qquad \frac{\Gamma \vdash \Delta}{\Gamma \vdash \phi \ll C, \Delta}\ weak\text{-}r$$

$$\frac{\forall \bar{w}.[y/(\mu \circ \nu)(t)](\phi') \ll C', \forall \bar{u}.s \doteq t \ll C_1, \forall \bar{v}.\phi \ll C_2, \Gamma \vdash \Delta}{\forall \bar{u}.s \doteq t \ll C_1, \forall \bar{v}.\phi \ll C_2, \Gamma \vdash \Delta}\ \overset{\rightarrow}{=} l$$

wobei $\bar{u}$, $\bar{v}$ und $\mathrm{bv}(s \doteq t) \cup \mathrm{bv}(\phi)$ paarw. disjunkt,
$(\mu \circ \nu)(\phi) = [y/(\mu \circ \nu)(s)](\phi')$,
$\mu \in \mathbf{Sub}^*$, $\nu \in \mathbf{Sub}(\bar{u} \cup \bar{v})$ idempotent, $\mathrm{fv}(\nu(\bar{u} \cup \bar{v})) \subseteq (\bar{u} \cup \bar{v})$
$\bar{w} := \mathrm{fv}([y/(\mu \circ \nu)(t)](\phi'))$, und $C' := C_1 \sqcup C_2 \sqcup \mathrm{Con}(\mu)$

abgeschlossen wären, wenn durch finale $\forall l/\exists r$ Regelanwendungen die universellen Präfixe entfernt werden. Durch diese Regelanwendungen werden die universellen Variablen durch Metavariablen ersetzt, die wiederum durch die abschließende Grundsubstitution σ zu Grundtermen werden. Diese Anwendung von σ wird in der folgenden Definition vorweggenommen. Da $\mathcal{U}_E$-Axiome Grundsequenzen sind, treten keine Constraints mehr auf.

7.2 Definition ($\mathcal{U}_E$-Axiome)
*Eine Grundsequenz $\Gamma \vdash \Delta$ heißt $\mathcal{U}_E$-**Axiom**, falls (mindestens) eine der folgenden Aussagen auf sie zutrifft:*

- *Es existieren Formeln $\forall \bar{u}.p(s_1,\ldots,s_n) \in \Gamma$ und $\exists \bar{v}.p(t_1,\ldots,t_n) \in \Delta$ mit $\bar{u} \cap \bar{v} = \emptyset$, sowie eine Grundsubstitution $\tau \in \mathbf{Sub}^0(\bar{u} \cup \bar{v})$ mit $\tau(s_i) = \tau(t_i)$ für alle $i = 1,\ldots,n$.*
- *Es existiert eine Formel $\exists \bar{u}.s \doteq t \in \Delta$, sowie eine Grundsubstitution $\tau \in \mathbf{Sub}^0(\bar{u})$ mit $\tau(s) = \tau(t)$.*

Die Substitution τ ist für den zweiten Punkt wichtig, um Axiome wie $\vdash \exists x.x = a$ zu erkennen. Man bemerke außerdem, daß hier wieder von disjunkten universellen Variablen ausgegangen wird. Diese Bedingung ist übrigens automatisch erfüllt, wenn man bemerkt, daß Objektvariablen grundsätzlich im Skopus eines Quantors auftreten, und daher zwei Variablen nur dann als gleich ansieht, wenn sie vom gleichen Quantor gebunden werden.

Nun können $\mathcal{U}_E$-Beweise definiert werden:

7.3 Definition ($\mathcal{U}_E$-Beweis)
*Ein $\mathcal{U}_E$-**Beweis** für eine geschlossene Formel $\phi \in \mathbf{Form}^0$ wird entsprechend Def. 6.5 definiert.*

*Ein $\mathcal{U}_E$-Beweis heißt **geschlossen**, falls er endlich ist, und eine Grundsubstitution $\sigma \in \mathbf{Sub}^0$ existiert, so daß $\sigma(\Gamma \vdash \Delta)$ für jedes Blatt $\Gamma \vdash \Delta$ ein $\mathcal{U}_E$-Axiom ist.*

Auch für den Abschluß von $\mathcal{U}_E$-Beweisen läßt sich eine Charakterisierung mit Constraints angeben. Die abschließenden Constraints müssen nun auch die universellen Variablen berücksichtigen, was wir dadurch realisieren, daß für jede universelle Variable im abschließenden Constraint eine Metavariable eingeführt wird. Man bemerke, daß diese Metavariablen nur für die Charakterisierung in Satz 7.5 benötigt werden. Sie werden nie im Beweisbaum oder im Benutzer-Constraint auftauchen.[5]

7.4 Definition (abschließender Constraint)
*Sei $\Gamma \vdash \Delta$ eine Sequenz. C heißt ein **abschließender Constraint** für $\Gamma \vdash \Delta$, falls $\forall \bar{u}.p(s_1,\ldots,s_n) \ll C_1 \in \Gamma$, $\exists \bar{v}.p(t_1,\ldots,t_n) \ll C_2 \in \Delta$ mit $\bar{u} \cap \bar{v} = \emptyset$ und*

$$C = C_1 \sqcup C_2 \sqcup \bigsqcup_{i=1}^{n} (\xi(s_i) = \xi(t_i)),$$

wobei $\xi := [\bar{u}/\bar{U}, \bar{v}/\bar{V}]$ jeder universellen Variable eine neue Metavariable zuordnet, oder $\forall \bar{v}.s \doteq t \ll C_1 \in \Delta$ und

$$C = C_1 \sqcup (\xi(s) = \xi(t)),$$

mit $\xi := [\bar{v}/\bar{V}]$.

*Ein **abschließender Constraint** eines Beweises ist wie für $\mathcal{M}_E$ als Konjunktion abschließender Constraints der Äste definiert, siehe Def. 6.6.*

[5] Es wäre auch möglich, Constraints für dieses Kriterium auf Objektvariablen zu erweitern. Für eine Implementierung ist dies evtl. das geschickteste Vorgehen.

7.5 Satz (Abschluß via Constraints)
Ein endlicher $\mathcal{U}_E$-Beweis ist genau dann geschlossen, wenn für ihn ein erfüllbarer abschließender Constraint existiert.

Beweis. Definition 7.3 bewirkt, daß ein Beweis genau dann geschlossen ist, wenn der Beweis abgeschlossen ist, der entsteht, wenn mit der $\forall l/\exists r$-Regel alle universellen Präfixe in den Blättern entfernt, und durch Metavariablen ersetzt werden. Dabei wird jede Formel $\forall\bar{u}.\phi$ im Antezedenten durch $\xi(\phi)$ ersetzt, und entsprechend im Sukzedenten. Nun kann Satz 6.7 angewandt werden, der die Charakterisierung für Beweise ohne universelle Variablen beinhaltet. ■

7.3. Korrektheit von $\mathcal{U}_E$

In diesem Abschnitt wird die Korrektheit von $\mathcal{U}_E$ gezeigt. Dabei gelten wieder die Bemerkungen am Anfang von Abschnitt 6.4.1 zur Korrektheit bezüglich der substitutiven Semantik, wenn die *ext*-Regel nicht verwendet wird.

7.6 Satz (Korrektheit von $\mathcal{U}_E$)
Falls für eine geschlossene Formel $\phi \in \mathbf{Form}^0$ ein geschlossener $\mathcal{U}_E$-Beweis existiert, ist ϕ E-allgemeingültig.

Beweis. Wie im Beweis der Korrektheit von $\mathcal{M}_E$ wird die Aussage verstärkt, und durch Induktion über die Höhe des Beweisbaums gezeigt:

> Wenn ein $\mathcal{U}_E$-Beweis mit Wurzel $\Gamma \vdash \Delta$ existiert, der mit $\sigma \in \mathbf{Sub}^0$ geschlossen ist, dann ist $\sigma(\Gamma \vdash \Delta)$ E-allgemeingültig.

Für den Induktionsanfang ist zu zeigen, daß alle $\mathcal{U}_E$-Axiome E-allgemeingültig sind. Wir zeigen dies nur für die Gleichungen; der Beweis für Sequenzen $\forall\bar{u}.p(\ldots),\ldots \vdash \forall\bar{v}.p(\ldots),\ldots$ verläuft ähnlich.

Sei $\Gamma \vdash \Delta$ also eine Grundsequenz mit $\exists\bar{v}.s \doteq t \in \Delta$, und $\tau \in \mathbf{Sub}^0(\bar{v})$ mit $\tau(s) = \tau(t)$. Für beliebige E-Strukturen $\mathcal{S}$, und Belegungen β gilt dann

$$\begin{aligned} & \mathrm{val}(\mathcal{S}, \beta\{v \leftarrow \mathrm{val}(\mathcal{S},\beta,\tau(v)) \mid v \in \bar{v}\}, s) \\ \stackrel{\text{Subst.}}{=} & \mathrm{val}(\mathcal{S},\beta,\tau(s)) \\ = & \mathrm{val}(\mathcal{S},\beta,\tau(t)) \\ \stackrel{\text{Subst.}}{=} & \mathrm{val}(\mathcal{S}, \beta\{v \leftarrow \mathrm{val}(\mathcal{S},\beta,\tau(v)) \mid v \in \bar{v}\}, t), \end{aligned}$$

also $\mathcal{S}, \beta\{v \leftarrow \mathrm{val}(\mathcal{S},\beta,\tau(v)) \mid v \in \bar{v}\} \models s \doteq t$, und daher $\mathcal{S},\beta \models \exists\bar{v}.s \doteq t$.

Von den Regeln sollen nur einige exemplarisch betrachtet werden. Die Beweise lassen sich größtenteils vom Beweis für $\mathcal{M}_E$ übertragen. Hier soll die Korrektheit der beiden interessantesten (und unübersichtlichsten) Regeln gezeigt werden, nämlich $\wedge r$ und $\stackrel{\rightarrow}{=} l$.

Zu $\wedge r$: Seien $\mathcal{S}$ eine E-Struktur, und $\sigma \in \mathbf{Sub}^0$ eine Grundsubstitution, so daß

$$\mathcal{S} \models \sigma(\Gamma \vdash \exists\bar{u}_1.[\bar{y}/\bar{Y}](\phi) \ll C, \exists\bar{u}.(\phi \wedge \psi) \ll C, \Delta)$$

und

$$\mathcal{S} \models \sigma(\Gamma \vdash \exists\bar{u}_2.[\bar{y}/\bar{Y}](\psi) \ll C, \exists\bar{u}.(\phi \wedge \psi) \ll C, \Delta).$$

Wir müssen zeigen, daß dann auch

$$\mathcal{S} \models \sigma(\Gamma \vdash \exists \bar{u}.(\phi \wedge \psi) \ll C, \Delta).$$

Interessant ist offenbar nur der Fall, daß $\sigma \in \mathrm{Sat}(C)$,

$$\mathcal{S} \models \sigma(\exists \bar{u}_1.[\bar{y}/\bar{Y}](\phi)) \quad \text{und} \quad \mathcal{S} \models \sigma(\exists \bar{u}_2.[\bar{y}/\bar{Y}](\psi)).$$

Da $\bar{u}_1$ und $\bar{u}_2$ disjunkt sind, existiert laut Semantik des $\exists$-Quantors eine Belegung β mit

$$\mathcal{S}, \beta \models \sigma([\bar{y}/\bar{Y}](\phi)), \quad \text{und} \quad \mathcal{S}, \beta \models \sigma([\bar{y}/\bar{Y}](\psi)).$$

Mit Hilfe des Substitutionslemmas erhält man

$$\mathcal{S}, \beta\{\bar{y} \leftarrow \mathrm{val}(\mathcal{S}, \beta, \sigma(\bar{Y}))\} \models \sigma(\phi), \quad \text{und} \quad \mathcal{S}, \beta\{\bar{y} \leftarrow \mathrm{val}(\mathcal{S}, \beta, \sigma(\bar{Y}))\} \models \sigma(\psi),$$

also

$$\mathcal{S}, \beta\{\bar{y} \leftarrow \mathrm{val}(\mathcal{S}, \beta, \sigma(\bar{Y}))\} \models \sigma(\phi \wedge \psi).$$

Daraus folgt offenbar

$$\mathcal{S} \models \sigma(\exists \bar{u}.\phi \wedge \psi),$$

da $\bar{u}$ alle freien Variablen von $\phi \wedge \psi$ enthalten muß, woraus die Aussage folgt.

Nach diesem Schema läßt sich auch die Korrektheit der anderen Regeln zeigen. Der wesentliche Trick besteht darin, den universellen Präfix mit Hilfe einer Belegung zu behandeln, ansonsten ähnelt die Argumentation der im Beweis für $\mathcal{M}_E$.

Zu $\overset{\rightrightarrows}{=}r$: Seien eine Grundsubstitution $\sigma \in \mathrm{Con}(\mu)$ und eine E-Struktur $\mathcal{S}$ gegeben mit

$$\mathcal{S} \models \sigma(\forall \bar{u}.s \doteq t) \tag{1}$$

und

$$\mathcal{S} \models \sigma(\forall \bar{v}.\phi). \tag{2}$$

Es muß gezeigt werden, daß dann auch[6]

$$\mathcal{S} \models \sigma(\forall \bar{w}.[y/(\mu \circ \nu)(t)](\phi')).$$

Wir zeigen dazu

$$\mathcal{S}, \beta \models \sigma([y/(\mu \circ \nu)(t)](\phi')) \tag{$\dagger$}$$

für alle Belegungen β. Sei also β beliebig. Wegen (1) gilt $\sigma(s) \doteq \sigma(t)$ unter allen Belegungen. Wir benutzen diese Tatsache nicht für β, sondern für

$$\beta' := \beta\{x \leftarrow \mathrm{val}(\mathcal{S}, \beta, \sigma(\nu(x))) \mid x \in \bar{u} \cup \bar{v}\},$$

um zu schließen:

$$\left.\begin{array}{ll}
\mathrm{val}(\mathcal{S}, \beta, \sigma(\nu(s))) & \\
= \mathrm{val}(\mathcal{S}, \beta, [x/\sigma(\nu(x)) \mid x \in \bar{u} \cup \bar{v}](\sigma(s))) & \\
= \mathrm{val}(\mathcal{S}, \beta', \sigma(s)) & \text{da } \mathrm{fv}(\nu(x)) \cap \mathrm{bv}(s) = \emptyset \\
= \mathrm{val}(\mathcal{S}, \beta', \sigma(t)) & \\
= \mathrm{val}(\mathcal{S}, \beta, [x/\sigma(\nu(x)) \mid x \in \bar{u} \cup \bar{v}](\sigma(s))) & \text{da } \mathrm{fv}(\nu(x)) \cap \mathrm{bv}(t) = \emptyset \\
= \mathrm{val}(\mathcal{S}, \beta, \sigma(\nu(t))) &
\end{array}\right\} (*)$$

[6] Der Leser vergewissere sich, daß dies tatsächlich der einzig interessante Fall ist! Vgl. etwa den Beweis für $\mathcal{M}_E$, S. 82.

Wegen (2) gilt entsprechend $\mathcal{S}, \beta' \models \sigma(\phi) \implies \mathcal{S}, \beta \models \sigma(\nu(\phi))$, da $\mathrm{fv}(\nu(x)) \cap \mathrm{bv}(\phi) = \emptyset$. Damit folgt

$$
\begin{array}{lll}
 & \mathcal{S}, \beta \models \sigma(\nu(\phi)) & \\
\Longrightarrow & \mathcal{S}, \beta \models \sigma((\mu \circ \nu)(\phi)) & \text{da } \sigma = \sigma \circ \mu \\
\Longrightarrow & \mathcal{S}, \beta \models \sigma([y/(\mu \circ \nu)(s)](\phi')) & \text{laut Definition d. Regel} \\
\Longrightarrow & \mathcal{S}, \beta\{y \leftarrow \mathrm{val}(\mathcal{S}, \beta, \sigma((\mu \circ \nu)(s)))\} \models \sigma(\phi') & \\
\Longrightarrow & \mathcal{S}, \beta\{y \leftarrow \mathrm{val}(\mathcal{S}, \beta, \sigma(\nu(s)))\} \models \sigma(\phi') & \text{da } \sigma = \sigma \circ \mu \\
\Longrightarrow & \mathcal{S}, \beta\{y \leftarrow \mathrm{val}(\mathcal{S}, \beta, \sigma(\nu(t)))\} \models \sigma(\phi') & \text{wegen } (*) \\
\Longrightarrow & \mathcal{S}, \beta\{y \leftarrow \mathrm{val}(\mathcal{S}, \beta, \sigma((\mu \circ \nu)(t)))\} \models \sigma(\phi') & \text{da } \sigma = \sigma \circ \mu \\
\Longrightarrow & \mathcal{S}, \beta \models \sigma([y/(\mu \circ \nu)(t)](\phi')) &
\end{array}
$$

was zu zeigen war, siehe (†). ∎

7.4. Vollständigkeit von $\mathcal{U}_S$

Die Vollständigkeit des gesamten Kalküls $\mathcal{U}_E$ ist natürlich eine triviale Folge der Vollständigkeit von $\mathcal{M}_E$: Die universellen Variablen können ja mit Hilfe der $\forall l/\exists r$-Regeln instantiiert werden. Dadurch lassen sich $\mathcal{M}_E$-Beweise direkt in $\mathcal{U}_E$-Beweise übersetzen.

Interessanter ist die Vollständigkeit des eingeschränkten Kalküls, der vom Automaten verwendet werden soll. In diesem werden universelle Variablen nur von der $\wedge r$-Regel durch Metavariablen ersetzt. Falls möglich, wird statt $\wedge r$ die $\wedge r'$-Regel benutzt, um unnötige Mehrfach-Instantiierungen zu vermeiden, und die Anwendung der ϵ-*cut*-Regel wird wieder auf ϵ-Terme eingeschränkt, die bereits im Beweis vorkommen.

Die Gleichheit wird hier nicht berücksichtigt: Die Vollständigkeit der Gleichheitsbehandlung ist eine Folge der Entsprechung zum Verfahren von Beckert, siehe 7.5.

7.7 Satz (Vollständigkeit von $\mathcal{U}_S$)
Sei $\mathcal{U}_S$ der Kalkül, der aus $\mathcal{U}_E$ entsteht, indem die Regeln folgendermaßen eingeschränkt werden:

- *Die Regeln ext, $\exists r$, $\forall l$, cut, weak-l, weak-r, sowie die Gleichheitsregeln dürfen nicht angewandt werden.*
- *Die Regel $\wedge r$ wird nur angewandt, falls* $\mathrm{fv}(\phi) \cap \mathrm{fv}(\psi) \neq \emptyset$.
- *Die ϵ-cut-Regel darf nur angewandt werden für ϵ-Terme, die in einem universellen Atom vorkommen, und deren freie Variablen im universellen Präfix dieser Formel gebunden werden.*

*Sei $\phi \in$ **Form**^{0} eine S-allgemeingültige Formel ohne Gleichheit. Dann existiert ein geschlossener $\mathcal{U}_S$-Beweis für ϕ.*

Beweis. Für den Beweis wird die Vollständigkeit von $\mathcal{M}_S$ ausgenutzt, siehe Satz 6.10. Da ϕ S-allgemeingültig ist, existiert ein geschlossener $\mathcal{M}_S$-Beweis B_0 für ϕ. Dieser soll in einen $\mathcal{U}_S$-Beweis B transformiert werden. Dazu wird zunächst eine Relation definiert, die die Entsprechung zwischen den Beweisen beschreibt. Constraints werden hier weggelassen, da sie für $\mathcal{U}_S$ nicht von Bedeutung sind.

Für zwei Sequenzen soll $\Sigma_0 \gtrdot \Sigma$ gelten (in Worten: „Σ_0 kann zu Σ universalisiert werden“), falls zu jeder Formel in Σ_0 eine entsprechende Formel mit universellem Präfix in Σ vorkommt, oder eine Formel, die aus jener mit Hilfe der $\mathcal{U}_S$-Regeln *außer* $\wedge r$ abgeleitet werden kann. Man definiert „$\gtrdot$" als kleinste reflexive, transitive Relation zwischen Sequenzen mit folgenden Eigenschaften:

1. $(\Gamma \vdash \Delta) \gtrdot (\Gamma_1 \vdash \Delta_1)$, falls $\Gamma \subseteq \Gamma_1$, und $\Delta \subseteq \Delta_1$.
2. $(\phi_0, \Gamma \vdash \Delta) \gtrdot (\forall \bar{u}.\phi, \Gamma \vdash \Delta)$, falls $\nu \in \mathbf{Sub}(\bar{u})$ existiert mit $\nu(u) \in \mathbf{M}$ für alle $u \in \bar{u}$, und $\phi_0 = \nu(\phi)$.
3. $(\Gamma \vdash \phi_0, \Delta) \gtrdot (\Gamma \vdash \exists \bar{u}.\phi, \Delta)$, falls $\nu \in \mathbf{Sub}(\bar{u})$ existiert mit $\nu(u) \in \mathbf{M}$ für alle $u \in \bar{u}$, und $\phi_0 = \nu(\phi)$.
4. $(\forall \bar{u}.(\phi \wedge \psi), \Gamma \vdash \Delta) \gtrdot (\forall \bar{u}_1.\phi, \forall \bar{u}_2.\psi, \Gamma \vdash \Delta)$, wobei $\bar{u}_1 := \mathrm{fv}(\phi)$ und $\bar{u}_2 := \mathrm{fv}(\psi)$.
5. $(\forall \bar{u}.\neg\phi, \Gamma \vdash \Delta) \gtrdot (\Gamma \vdash \exists \bar{u}.\phi, \Delta)$.
6. $(\Gamma \vdash \exists \bar{u}.\neg\phi, \Delta) \gtrdot (\forall \bar{u}.\phi, \Gamma \vdash \Delta)$.
7. $(\forall \bar{u}.\exists x.\phi, \Gamma \vdash \Delta) \gtrdot (\forall \bar{u}.[x/\epsilon x.\phi](\phi), \Gamma \vdash \Delta)$.
8. $(\Gamma \vdash \exists \bar{u}.(\phi \wedge \psi), \Delta) \gtrdot (\Gamma \vdash \exists \bar{u}_1.\phi, \Delta)$, falls $\mathrm{fv}(\phi) \cap \mathrm{fv}(\psi) = \emptyset$, wobei $\bar{u}_1 := \mathrm{fv}(\phi)$.
9. $(\Gamma \vdash \exists \bar{u}.(\phi \wedge \psi), \Delta) \gtrdot (\Gamma \vdash \exists \bar{u}_2.\psi, \Delta)$, falls $\mathrm{fv}(\phi) \cap \mathrm{fv}(\psi) = \emptyset$, wobei $\bar{u}_2 := \mathrm{fv}(\psi)$.

Wir werden zwei Eigenschaften von $\gtrdot$ benötigen, nämlich

(M) Falls $(\Gamma \cup \Gamma' \vdash \Delta \cup \Delta') \gtrdot \Sigma$, dann gilt $(\Gamma \vdash \Delta) \gtrdot \Sigma$.

(A) Falls $(\Gamma_i \vdash \Delta_i) \gtrdot (\Gamma'_i \vdash \Delta'_i)$ für $i = 1, 2$, so gilt
$$(\Gamma_1 \cup \Gamma_2 \vdash \Delta_1 \cup \Delta_2) \gtrdot (\Gamma'_1 \cup \Gamma'_2 \vdash \Delta'_1 \cup \Delta'_2).$$

Dabei ist (M) eine Monotonie-Eigenschaft, die unmittelbar aus dem ersten Punkt in der Definition und der Transitivität von $\gtrdot$ folgt. (A) ist eine Art Additivität, die aus dem Umstand folgt, daß die Universalisierung sich jeweils auf einzelne Formeln bezieht.

Wir werden nun angeben, wie aus einem geschlossenen $\mathcal{M}_S$-Beweis B_0 mit Wurzel Σ_0 ein geschlossener $\mathcal{U}_S$-Beweis B mit Wurzel Σ gewonnen werden kann, falls $\Sigma_0 \gtrdot \Sigma$. Da $\gtrdot$ reflexiv ist, kann so für $\Sigma_0 := \Sigma := (\vdash \phi)$ ein geschlossener $\mathcal{U}_S$-Beweis für ϕ konstruiert werden.

Der Beweis erfolgt durch Induktion über die Höhe des Beweisbaums B_0. Im Gegensatz zur Übersetzung im Beweis von Satz 6.9 entspricht nicht jeder Regelanwendung in B_0 eine Regelanwendungen in B. Speziell die $\exists r$-Anwendungen können in B einfach ausgelassen werden. Mit der Regel $\wedge r'$ können gelegentlich sogar Äste gespart werden.

Für den Induktionsanfang ist zu zeigen: Falls $\sigma(\Gamma_0 \vdash \Delta_0)$ ein $\mathcal{G}_E$-Axiom ist, und $(\Gamma_0 \vdash \Delta_0) \gtrdot (\Gamma \vdash \Delta)$, so ist $\sigma(\Gamma \vdash \Delta)$ ein $\mathcal{U}_E$-Axiom.[7] Seien $p(\bar{s}_0) \in \Gamma_0$ und $p(\bar{t}_0) \in \Delta_0$ mit $\sigma(\bar{s}_0) = \sigma(\bar{t}_0)$.[8] Wegen der Definition von $\gtrdot$ müssen Formeln $\forall \bar{u}.p(\bar{s}) \in \Gamma$ und $\forall \bar{v}.p(\bar{t}) \in \Delta$ existieren mit $\bar{s}_0 = [\bar{u}/\bar{U}]\bar{s}$ und $\bar{t}_0 = [\bar{v}/\bar{V}]\bar{t}$ für bestimmte Metavariablen $\bar{U}$, bzw. $\bar{V}$. Die Substitution ν, die für $\mathcal{U}_E$-Axiome benötigt wird kann nun definiert werden durch $\nu(\bar{u}) := \sigma(\bar{U})$, $\nu(\bar{v}) := \sigma(\bar{V})$. Es gilt dann: $\nu(\sigma(\bar{s})) = \sigma(\bar{s}_0) = \sigma(\bar{t}_0) = \nu(\sigma(\bar{t}))$.

[7] Die Axiome von $\mathcal{U}_S$ sind dieselben wie für $\mathcal{U}_E$, wobei die Gleichungsaxiome nicht benötigt werden.
[8] $\bar{s}$ und $\bar{t}$ sind Abkürzungen für Listen von Termen.

Der Induktionsschritt erfolgt durch Fallunterscheidung nach der untersten in B_0 angewandten Regel:

$$\frac{\Sigma_0^1 \quad \dots \quad \Sigma_0^k}{\Sigma_0}\ R_0$$

Gegeben ist jeweils eine Sequenz Σ mit $\Sigma_0 \gtrdot \Sigma$, und es muß ein geschlossener $\mathcal{U}_S$-Beweis für Σ angegeben werden. Dies soll nur für die Regeln $\exists r$, $\wedge l$, $\wedge r$ und ϵ-*cut* geschehen; die Argumentation für die anderen verläuft ähnlich.

Zu $\exists r$: Hier ist $\Sigma_0 = \Gamma_0 \vdash \exists x.\phi, \Delta_0$ und $\Sigma_0^1 = \Gamma_0 \vdash [x/X](\phi), \exists x.\phi, \Delta_0$. Damit gilt $\Sigma_0^1 \gtrdot \Sigma_0 \gtrdot \Sigma$. Die Induktionshypothese kann nun auf Σ_0^1 und Σ angewandt werden. Der $\exists r$-Beweisschritt kann in der Übersetzung also einfach ausgelassen werden.

Zu $\wedge l$: Es ist $\Sigma_0 = \phi_0 \wedge \psi_0, \Gamma_0 \vdash \Delta_0$ und $\Sigma_0^1 = \phi_0, \psi_0, \Gamma_0 \vdash \Delta_0$. Wegen (M) gilt $(\Gamma_0 \vdash \Delta_0) \gtrdot \Sigma$ und $(\phi_0 \wedge \psi_0 \vdash) \gtrdot \Sigma$. Da $\gtrdot$ als kleinste reflexive und transitive Relation mit den gegebenen Eigenschaften definiert wurde, muß eine Folge von Schritten existieren, die $\phi_0 \wedge \psi_0 \vdash$ durch sukzessive Anwendung der Punkte der Definition von $\gtrdot$ in Σ überführt. Dabei können auf $\phi \wedge \psi$ zunächst nur die Punkte 2 und 4 angewandt werden. Je nachdem, welche Schritte angewandt werden, sind zwei Fälle möglich:

1. Es ist $\Sigma = \forall \bar{u}.(\phi \wedge \psi), \Gamma \vdash \Delta$ für gewisse $\bar{u}, \phi, \psi$ mit $(\phi_0 \wedge \psi_0 \vdash) \gtrdot (\forall \bar{u}.(\phi \wedge \psi) \vdash)$. Anwendung der $\wedge l$-Regel auf Σ liefert die Sequenz $\Sigma_1 = \forall \bar{u}_1.\phi, \forall \bar{u}_2.\psi, \Gamma \vdash \Delta$. Es ist $(\phi_0, \psi_0 \vdash) \gtrdot (\forall \bar{u}_1.\phi, \forall \bar{u}_2.\psi \vdash)$.

$$\begin{array}{ll} \Sigma_0^1 = (\phi_0, \psi_0, \Gamma_0 \vdash \Delta_0) & \\ \quad \gtrdot (\forall \bar{u}_1.\phi, \forall \bar{u}_2.\psi, \forall \bar{u}.(\phi \wedge \psi), \Gamma \vdash \Delta) & \text{wegen (A)} \\ \quad \gtrdot (\forall \bar{u}_1.\phi, \forall \bar{u}_2.\psi, \Gamma \vdash \Delta) & \text{laut Def. von } \gtrdot \\ \quad = \Sigma^1. & \end{array}$$

   Damit läßt sich die Induktionshypothese auf Σ_0^1 und Σ^1 anwenden. In diesem Fall wird die Regelanwendung also übernommen.

2. $\Sigma = \Gamma \vdash \Delta$ und $\forall \bar{u}.(\phi \wedge \psi) \in \Gamma$ für *keine* $\bar{u}$, ϕ, ψ mit $(\phi_0 \wedge \psi_0 \vdash) \gtrdot (\forall \bar{u}.(\phi \wedge \psi) \vdash)$. Eine Verfolgung der Schritte, die zu $\Sigma_0 \gtrdot \Sigma$ führen, zeigt, daß es gewisse $\bar{u}$, ϕ, ψ geben muß mit

$$(\phi_0 \wedge \psi_0 \vdash) \gtrdot (\forall \bar{u}.(\phi \wedge \psi) \vdash) \gtrdot (\forall \bar{u}_1.\phi, \forall \bar{u}_2.\psi \vdash) \gtrdot \Sigma.$$

   Daraus folgt $(\phi_0, \psi_0 \vdash) \gtrdot (\forall \bar{u}_1.\phi, \forall \bar{u}_2.\psi \vdash) \gtrdot \Sigma$. Mit (A) bekommt man schließlich: $\Sigma_0^1 = (\phi_0, \psi_0, \Gamma_0 \vdash \Delta_0) \gtrdot \Sigma$, und die Induktionshypothese kann für Σ_0^1 und Σ angewandt werden. Hier braucht auf Σ keine Regel angewandt werden, da die Konjunktion in Σ bereits aufgespalten ist.

Nach diesem Schema verlaufen auch die Beweise für die Regeln $\exists l$, $\neg l$ und $\neg r$.

Zu $\wedge r$: Es ist $\Sigma_0 = \Gamma_0 \vdash \phi_0 \wedge \psi_0, \Delta_0$, $\Sigma_0^1 = \Gamma_0 \vdash \phi_0, \Delta_0$ und $\Sigma_0^2 = \Gamma_0 \vdash \psi_0, \Delta_0$. Wegen (M) gilt wieder $(\Gamma_0 \vdash \Delta_0) \gtrdot \Sigma$. Wie für $\wedge l$ muß zwischen zwei Fällen unterschieden werden:

1. Es ist $\Sigma = \Gamma \vdash \exists \bar{u}.(\phi \wedge \psi), \Delta$ für gewisse $\bar{u}, \phi, \psi$ mit $(\vdash \phi_0 \wedge \psi_0) \gtrdot (\vdash \exists \bar{u}.(\phi \wedge \psi))$. In diesem Fall wird auf Σ eine der Regeln $\wedge r$ oder $\wedge r'$ angewandt. Welche von beiden verwendet wird, hängt davon ab, ob die freien Variablen von ϕ und ψ disjunkt sind.

(a) $\text{fv}(\phi) \cap \text{fv}(\psi) \neq \emptyset$. Dann kann $\wedge r$ auf Σ angewandt werden. Die neu eingeführten freien Variablen $\bar{Y}$ werden dabei so gewählt, daß $(\vdash \phi_0) \gg (\vdash \exists \bar{u}_1.[\bar{y}/\bar{Y}](\phi))$ und $(\vdash \psi_0) \gg (\vdash \exists \bar{u}_2.[\bar{y}/\bar{Y}](\psi))$. Aus der Regelanwendung entstehen Sequenzen Σ^1 und Σ^2 mit $\Sigma^1 = \Gamma \vdash \exists \bar{u}_1.[\bar{y}/\bar{Y}](\phi), \exists \bar{u}.(\phi \wedge \psi), \Delta$ und Σ^2 entsprechend. Mit (A) gilt dann:

$$\Sigma_0^1 = (\Gamma_0 \vdash \phi_0, \Delta_0) \gg (\Gamma \vdash \exists \bar{u}_1.[\bar{y}/\bar{Y}](\phi), \exists \bar{u}.(\phi \wedge \psi), \Delta) = \Sigma^1$$

und entsprechend $\Sigma_0^2 \gg \Sigma^2$. Die Induktionshypothese kann also auf die beiden entstandenen Sequenzen angewandt werden. Die Verzweigung von B_0 überträgt sich auf B.

(b) $\text{fv}(\phi) \cap \text{fv}(\psi) = \emptyset$. In diesem Fall wird die $\wedge r'$-Regel auf Σ angewandt. Es entstehen die Sequenzen $\Sigma^1 = \Gamma \vdash \exists \bar{u}_1.\phi, \Delta$ und $\Sigma^2 = \Gamma \vdash \exists \bar{u}_2.\psi, \Delta$. Es gilt:

$$\begin{aligned} \Sigma_0^1 &= (\Gamma_0 \vdash \phi_0, \Delta_0) \\ &\gg (\Gamma \vdash \exists \bar{u}_1.\phi, \exists \bar{u}.(\phi \wedge \psi), \Delta) && \text{wegen (A)} \\ &\gg (\Gamma \vdash \exists \bar{u}_1.\phi, \Delta) && \text{laut Def. von } \gg \text{, vorletzter Punkt} \\ &= \Sigma^1 \end{aligned}$$

und entsprechend $\Sigma_0^2 \gg \Sigma^2$. Damit kann die Induktionshypothese auf die zwei entstandenen Äste angewandt werden. An dieser Stelle werden die beiden letzten Punkte in der Definition von $\gg$ benötigt, um sicherzustellen, daß die Formel $\exists \bar{u}.(\phi \wedge \psi)$ nicht beibehalten werden muß.

2. Es ist $\Sigma = \Gamma \vdash \Delta$ und *nicht* $\exists \bar{u}.(\phi \wedge \psi) \in \Delta$ für irgendwelche $\bar{u}$, ϕ, ψ mit $(\vdash \phi_0 \wedge \psi_0) \gg (\vdash \exists \bar{u}.(\phi \wedge \psi))$. Da $\Sigma_0 \gg \Sigma$ muß dann für gewisse $\bar{u}$, ϕ, ψ gelten:

$$(\vdash \phi_0 \wedge \psi_0) \gg (\vdash \exists \bar{u}.(\phi \wedge \psi)) \gg \left\{ \begin{array}{c} \vdash \exists u_1.\phi \\ \text{oder} \\ \vdash \exists u_2.\psi \end{array} \right\} \gg \Sigma$$

Sei o.B.d.A. $(\vdash \exists u_1.\phi) \gg \Sigma$; der andere Fall verläuft dazu symmetrisch. Dann ist $(\vdash \phi_0) \gg (\vdash \exists u_1.\phi) \gg \Sigma$. Mit $(\Gamma_0 \vdash \Delta_0) \gg \Sigma$ und (A) folgt:

$$\Sigma_0^1 = (\Gamma_0 \vdash \phi_0, \Delta_0) \gg \Sigma.$$

Damit kann die Induktionshypothese angewandt werden. Der rechte (bzw. im symmetrischen Fall der linke) Ast in B_0 wird für die Übersetzung nicht benötigt.

Zu ϵ-***cut*:** Es ist $\Sigma_0 = (\Gamma_0 \vdash \Delta_0)$, $\Sigma_0^1 = (\Gamma_0 \vdash \exists x.\phi_0, \Delta_0)$ und $\Sigma_0^2 = ([x/\epsilon x.\phi_0](\phi_0), \Gamma_0 \vdash \Delta_0)$, wobei $\epsilon x.\phi_0$ ein ganzer Term ist, der in einer atomaren Formel ψ_0 von Σ_0 vorkommt. Da ψ_0 atomar ist, und $\Sigma_0 \gg \Sigma$, muß Σ ein universelles Atom $\forall \bar{u}.\psi$, bzw. $\exists \bar{u}.\psi$ enthalten, das Universalisierung von ψ_0 ist. Dann muß ψ einen Teilterm $\epsilon x.\phi$ enthalten mit $\epsilon x.\phi_0 = \nu(\epsilon x.\phi)$, wobei $\nu \in \mathbf{Sub}(\bar{u})$ einige der universellen Variablen $\bar{u}$ zu Metavariablen machen kann. Insgesamt enthält $\epsilon x.\phi$ also nur freie Variablen, die im universellen Präfix von ψ gebunden werden, d.h. $\bar{u}_1 := \text{fv}(\epsilon x.\phi) \subseteq \bar{u}$. Damit darf die ϵ-*cut*-Regel von $\mathcal{U}_S$ auf $\Sigma = \Gamma \vdash \Delta$ angewandt werden, und man erhält $\Sigma^1 = \Gamma \vdash \forall \bar{u}_1.\exists x.\phi, \Delta$ und $\Sigma^2 = \forall \bar{u}_1.[x/\epsilon x.\phi](\phi), \Gamma \vdash \Delta$. Offensichtlich ist $\Sigma_0^i \gg \Sigma^i$, also kann die Induktionshypothese auf die beiden Äste angewandt werden. ■

Der interessanteste Aspekt dieses Beweises ist sicherlich der zweite Fall im Beweis für $\land r$: Hier kann und braucht jeweils nur einer von zwei Ästen in B_0 transformiert zu werden. Dieser Fall entsteht, wenn in B_0 eine Formel zweimal instantiiert wird, die in B von der $\land r'$-Regel entfernt wurde. Die folgende $\mathcal{M}_S$-Ableitung möge dies verdeutlichen:

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\boxed{B_0^{11}}}{\vdash p(X), p(X_1), \ldots} \quad \dfrac{\boxed{B_0^{12}}}{\vdash p(X), q(Y_1), \ldots}}{\vdash p(X), p(X_1) \land q(Y_1), \ldots}\,\land r}{\vdash p(X), \exists x.\exists y.(p(x) \land q(y))}\,\exists r \qquad \dfrac{\dfrac{\dfrac{\boxed{B_0^{21}}}{\vdash q(Y), p(X_2), \ldots} \quad \dfrac{\boxed{B_0^{22}}}{\vdash q(Y), q(Y_2), \ldots}}{\vdash q(Y), p(X_2) \land q(Y_2), \ldots}\,\land r}{\vdash q(Y), \exists x.\exists y.(p(x) \land q(y))}\,\exists r}{\vdash p(X) \land q(Y), \exists x.\exists y.(p(x) \land q(y))}\,\land r}{\vdash \exists x.\exists y.(p(x) \land q(y))}\,\exists r$$

Die Formel $\exists x.\exists y.(p(x) \land q(y))$ wird auf jedem Ast zweimal instantiiert. Bei der Übersetzung in einen $\mathcal{U}_S$-Beweis werden die $\exists r$-Regel-Anwendungen ignoriert. Von den $\land r$-Anwendungen wird nur die erste beibehalten. Man erhält den Beweis

$$\dfrac{\dfrac{\boxed{B^{11}}}{\vdash \exists x.p(x)} \quad \dfrac{\boxed{B^{22}}}{\vdash \exists y.q(y)}}{\vdash \exists x.\exists y.(p(x) \land q(y))}\,\land r'$$

Die Formeln $\exists x.p(x)$ und $\exists y.q(y)$ sind Universalisierungen für die Instantiierungen $p(X)$ und $p(X_1)$, bzw. $q(Y)$ und $q(Y_2)$ auf den beiden äußeren Ästen. Die Teilbeweise zu den beiden inneren Ästen brauchen nicht transformiert zu werden.

7.5. Verhältnis zum Verfahren von Beckert

Die Entwicklung des Kalküls $\mathcal{U}_E$ mit universellen Variablen und Constraints zielte darauf hin, die Möglichkeit zur Einbettung von bestehenden Verfahren zur automatischen Gleichheitsbehandlung zu bieten. Es ist noch zu zeigen, daß der Kalkül dies tatsächlich leistet. Grundlage ist das von Beckert in [4] vorgestellte Verfahren zur automatischen Gleichheitsbehandlung in Tableaubeweisern.

Im Gegensatz zum hier verfolgten Ansatz ist jenes Verfahren zunächst nicht auf Kalkülebene in den Tableaubeweiser eingebunden. Letzterer behandelt die Gleichheit vielmehr als gewöhnliches Prädikat. Das Verfahren wird beim Astabschluß verwendet, um zu ermitteln, ob es komplementäre Formeln auf den Ästen gibt, die unter Verwendung der vorhandenen Gleichungen unifizierbar sind. Es handelt sich also um ein sog. *E-Unifikations-Problem*: Zwei Formeln müssen bezüglich der Theorie unifiziert werden, welche durch die auf dem Ast vorhandenen Gleichungen gegebenen ist.

Einfachere Verfahren berücksichtigen universelle Variablen nicht. Für jede freie Variable einer Gleichung darf nur eine Instanz eingesetzt werden. Man spricht von *rigiden* Variablen, bzw. rigider E-Unifikation. Da *eine* Substitution für die rigiden Variablen gesucht werden muß, die *alle* Äste abschließt, also mehrere Paare von Formeln unifiziert, redet man von *simultaner* E-Unifikation.

Um etwa aus $\forall x, y, z.\,(x \cdot y) \cdot z \doteq x \cdot (y \cdot z)$ zu folgern, daß $((a \cdot b) \cdot c) \cdot d \doteq a \cdot (b \cdot (c \cdot d))$, müssen im Tableau durch Anwendung der γ-Regel zwei Instanzen der Assoziativität erzeugt werden. Das Verfahren zur rigiden E-Unifikation erhält dann die Gleichungen

$$(X \cdot Y) \cdot Z \doteq X \cdot (Y \cdot Z)$$
$$(X_1 \cdot Y_1) \cdot Z_1 \doteq X_1 \cdot (Y_1 \cdot Z_1)$$

und kann die Terme $((a \cdot b) \cdot c) \cdot d$ und $a \cdot (b \cdot (c \cdot d))$ etwa mit der Instantiierung

$$[X/a \cdot b, Y/c, Z/d, X_1/a, Y_1/b, Z_1/c \cdot d]$$

unifizieren. Das Verfahren von Beckert berücksichtigt universelle Variablen: Variablen, bezüglich derer die Gleichungen universell sind, dürfen für die Unifikation mehrfach instantiiert werden. Effektiv arbeitet das Verfahren also nicht nur mit Gleichungen, sondern mit universellen Gleichungen. Da nicht alle Variablen einer Gleichung universell sein brauchen, sondern weiterhin einige rigide darunter sein könne, redet man von *gemischter E-Unifikation*. In obigem Beispiel bräuchte der Tableaubeweiser nur eine Instanz der Assoziativität erzeugen, die der Unifikations-Algorithmus dann beliebig oft anwenden kann.

Der klassische Knuth-Bendix-Algorithmus und der „unfailing“ Knuth-Bendix-Algorithmus beziehen sich auf den Spezialfall, daß nur universelle Variablen vorkommen, also sog. *universelle E*-Unifikations-Probleme. Diese Verfahren sind für die Unifikations-Probleme, die beim Abschluß von Tableaubeweisen auftreten, nicht geeignet, da jedenfalls rigide Variablen berücksichtigt werden müssen. Das Verfahren von Beckert kann als Synthese von constraintbasierten Verfahren zur rigiden E-Unifikation und dem vervollständigungsbasierten „unfailing“ Knuth-Bendix-Algorithmus verstanden werden.

Die Effizienzsteigerung, die durch die Berücksichtung universeller Variablen bei der Gleichheitsbehandlung möglich wird, ist der Hauptgrund für die Verwendung universeller Variablen, bzw. universeller Präfixe im hier vorgestellten Kalkül.

7.5.1. Überblick über das Verfahren von Beckert

Das Verfahren von Beckert setzt sich aus zwei Systemen von Transformationsregeln zusammen. Eines stellt eine Vervollständigungsprozedur dar, die aus den gegebenen Gleichungen eine (in einem bestimmten Sinne vollständige) Menge von Rewrite-Regeln mit Constraints erzeugt. Das andere verwendet diese Rewrite-Regeln zur Normalisierung der zu unifizierenden Terme. Da im allgemeinen kein endliches vollständiges Rewrite-System existiert, die Vervollständigung also nicht terminieren muß, laufen die beiden Prozesse gleichzeitig ab: Die zu unifizierenden Terme werden mittels einer Näherung des Rewrite-Systems normalisiert, danach wird die nächste Näherung berechnet, und so weiter.

Dadurch werden mitunter überflüssige Normalisierungen durchgeführt, nämlich dann, wenn sich in einer späteren Iteration herausstellt, daß eine Rewrite-Regel gelöscht werden kann, da sie für die Vollständigkeit nicht nötig ist. Dies läßt sich jedoch nicht vermeiden, da nicht entscheidbar ist, welche Rewrite-Regeln persistent sind. Da die Regeln in jeder Iteration korrekt sind, wird nur die Effizienz und nicht die Korrektheit des Verfahrens beeinträchtigt. Dieses Phänomen wird natürlich auch bei der Übertragung des Verfahrens auf den Kalkül $\mathcal{U}_E$ auftreten.

Für die zu unifizierenden Terme werden im allgemeinen *mehrere* Normalisierungen erzeugt, die mit unterschiedlichen Constraints versehen sind. Für zu unifizierende Terme s und t werden alle Paare von Normalisierungen $s' \ll C_1$ und $t' \ll C_2$ betrachtet, die *syntaktisch* unifizierbar sind, also $\mu(s) = \mu(t)$. Dann ist jedes $\sigma \in \mathrm{Sat}(C_1 \sqcup C_2 \sqcup \mathrm{Con}(\mu))$[9] ein E-Unifikator von s und t. Wenn $C_1 \sqcup C_2 \sqcup \mathrm{Con}(\mu)$ erfüllbar ist, wird also eine Substitution ν mit $\mathrm{Con}(\nu) \equiv C_1 \sqcup C_2 \sqcup \mathrm{Con}(\mu)$ ermittelt, und an den Tableaubeweiser weitergegeben.

[9] Die Notation wurde an die in dieser Arbeit verwendete angepaßt.

Leider ist das Verfahren in dieser Form für die simultane gemischte E-Unifikation noch nicht vollständig: In einigen Fällen ist es nötig, Rewrite-Regeln *in der umgekehrten Richtung* auf die gewonnenen Unifikatoren anzuwenden, um eine vollständige Menge von Unifikatoren zu erhalten. Dies bedeutet im Prinzip, daß die von den Constraints vorgegebene Richtung der Gleichungsanwendung ignoriert wird, und damit der Suchraum genauso groß wird, wie er ohne Ordnungsconstraints war.

Nun ist aber die Vollständigkeit der E-Unifikations-Prozedur von untergeordneter Bedeutung: Wichtig ist, daß der Tableaubeweiser *zusammen* mit der verwendeten Unifikations-Prozedur vollständig ist. Alle *bisher bekannten* Beweisaufgaben, für die die Prozedur ohne Rückwärtsanwendung von Regeln unvollständig ist, lassen sich vom Gesamtsystem dennoch beweisen, und zwar, indem von einigen Gleichungen durch Anwendung der γ-Regel weitere Instanzen erzeugt werden. Mit der so erweiterten Gleichungsmenge findet die Unifikationsprozedur schließlich doch einen Unifikator. Dadurch tauscht man die Ineffizienz der ungerichteten Gleichheitsanwendung gegen die Ineffizienz, die mit einem größeren Beweis und zusätzlichen Instantiierungen einhergeht, ein. Die Praxis zeigt, daß dies im allgemeinen ein guter Tausch ist.

Es sind keine Beispiele bekannt, für welche die Kombination von Tableaubeweiser und Unifikationsprozedur ohne Rückwärtsanwendung unvollständig wäre. Die Vollständigkeit wurde allerdings auch noch nicht bewiesen. In [8] wird jedoch die Vollständigkeit eines vergleichbaren Verfahrens gezeigt, so daß anzunehmen ist, daß die Rückwärtsanwendung für die Vollständigkeit tatsächlich nicht nötig ist.

Die effiziente Einbindung des Verfahrens von Beckert in einen Tableaubeweiser ist nicht ganz einfach. Zum einen ist die simultane gemischte E-Unifikation nicht entscheidbar. Das Verfahren kann also höchstens eine vollständige Menge von Unifikatoren *aufzählen*. Da auch der Tableaubeweiser nur mögliche Tableaux aufzählen kann, müssen die Aufzählungen durch *dovetailing* ineinander „verzahnt" werden. Praktisch wird die Unifikationsprozedur mit einer Suchtiefe oder Zeitgrenze versehen, die dann bei einer Beweissuche durch inkrementelle Tiefensuche parallel zur Suchtiefe hochgesetzt werden kann.

Eine weitere Schwierigkeit tritt auf, wenn das Verfahren (etwa innerhalb der gegebenen Zeitschranken) keinen Unifikator findet. Dann wird die Tableauprozedur den Beweisbaum erweitern, und erneut nach einem Unifikator fragen. Nun ist die Menge der Gleichungen und potentiell komplementären Formelpaare seit der ersten Anfrage im allgemeinen gewachsen, aber die alten Gleichungen sind immer noch vorhanden. Auf diesen wird wieder die Vervollständigung vorgenommen, was sehr rechenaufwendig sein kann. Darum merkt sich die $_3T^AP$-Implementierung in solchen Fällen das ermittelte Rewrite-System von der ersten Anfrage, um es bei der zweiten zu ergänzen. Man spricht hier von *inkrementeller* Vervollständigung. Die Buchhaltung über die jeweils berechneten Rewrite-Regeln ist natürlich nicht ganz einfach: Die Gleichungsmengen können ja von Ast zu Ast verschieden sein, Backtracking muß berücksichtigt werden, etc.

Es soll nun erläutert werden, wie das Verfahren für ein Beweissystem adaptiert werden kann, das den Kalkül $\mathcal{U}_E$ verwendet.

7.5.2. Einbettung des Verfahrens in $\mathcal{U}_E$

Für diesen Abschnitt wird aus Platzgründen vorausgesetzt, daß der Leser mit dem Verfahren und der Notation von [4] vertraut ist. Diese Notation soll für Rewrite-Regeln und dergleichen verwendet werden. Was Gleichungen, Substitutionen, etc. angeht, werden wir bei unserer bisherigen Notation bleiben.

Sowohl die Vervollständigung als auch die Normalisierung sollen innerhalb des Kalküls $\mathcal{U}_E$ dargestellt werden. In [4] beginnt die Vervollständigung mit einem System von Rewrite-Regeln $\mathcal{R}_0$, welches zu jeder universellen Gleichung $\forall \bar{u}.s \doteq t$ des Problems zwei Regeln $(\forall \bar{u})(s \rightsquigarrow t \ll \langle \mathrm{id}, t \prec s \rangle)$ und $(\forall \bar{u})(t \rightsquigarrow s \ll \langle \mathrm{id}, s \prec t \rangle)$ enthält. Hieraus werden durch die Vervollständigungsprozedur neue Rewrite-Regeln erzeugt. Constraints in [4] haben die Form $\langle \nu, O \rangle$, wobei ν eine idempotente Substitution ist, und O eine Konjunktion von Ordnungsconstraints $s \prec t$. Sind in unserer Constraintsprache Ordnungsconstraints enthalten, entspricht dies einem Constraint $\mathrm{Con}(\nu) \sqcup O$ in unserer Notation.

Es sollen mehrere Möglichkeiten vorgestellt werden, solche Rewrite-Regeln in unsere Logik aufzunehmen, die allerdings recht ähnlich sind. Wir beginnen mit einer Beschreibung der einfachsten Variante, um dann einige Alternativen zu diskutieren.

Der naheliegendste Weg besteht darin, $\rightsquigarrow$ in die Syntax aufzunehmen, und mit der gleichen Semantik wie $\doteq$ zu versehen, wobei der automatische Beweiser die Gleichung nur in Pfeilrichtung anwendet.[10] Der Mensch kann solche gerichteten Gleichungen weiterhin in beiden Richtungen verwenden. Die Constraintsprache wird um Ordnungsconstraints erweitert. Um das System $\mathcal{R}_0$ zu erzeugen, wird eine zusätzliche Regel eingeführt, die eine universelle Gleichung in zwei gerichtete Gleichungen umformt. Die Transformationsregeln (V), (KP) und (Ko) von Beckert können durch Anwendung der Gleichheitsregeln von $\mathcal{U}_E$ zwischen gerichteten Gleichungen, sowie durch die *weak*-Regeln simuliert werden, wie man leicht überprüft. Die Regeln (L), (Ä) und (S) können sämtlich mit Hilfe der *weak*-Regeln ausgeführt werden.[11] Auch die Normalisierungsschritte können durch die Gleichheitsregeln simuliert werden, und zwar nun, indem gerichtete Gleichungen auf andere Formeln der Sequenz angewandt werden.

In seiner ursprünglichen Form betrifft die Normalisierung nur Paare von potentiell komplementären Formeln, bzw. Termpaare aus potentiell ungültigen Ungleichungen. Es scheint jedoch sinnvoll, die Anwendung von gerichteten Gleichungen auf die ganze Sequenz auszudehnen. Damit werden nämlich alle vorkommenden Formeln von vornherein bezüglich einer bestimmten Termordnung normalisiert. Wenn der Benutzer diese Termordnung vorgibt, bedeutet dies im allgemeinen eine Vereinfachung der Formeln. Wenn viele gerichtete Gleichungen vorhanden sind, werden die Sequenzen durch dieses Vorgehen jedoch sicherlich zu lang, da auch für komplexe Formeln die verschiedensten Normalisierungen erzeugt werden. Die beste Alternative besteht wohl darin, gerichtete Gleichungen auf komplexe Formeln dann anzuwenden, wenn der entstehende Constraint vom Benutzer-Constraint subsumiert wird.

Zumindest müssen jedenfalls die universellen Atome normalisiert werden, da diese für den Beweisabschluß relevant sind. Damit erübrigt sich die Ermittlung einer unifizierenden Substitution: Die Sequenzen werden mit sämtlichen Normalisierungen der universellen Atome versehen. Liefert das Verfahren in der Formulierung bei Beckert einen E-Unifikator, der den Beweis abschließt, so folgt daraus in unserer Variante, daß jedes Blatt des Beweisbaums unter diesem Unifikator (bzw. unter jeder Grundsubstitution, die

[10] Implementiert man das Verfahren mit Rückwärtsanwendung, müssen gerichtete Gleichungen natürlich manchmal doch in der anderen Richtung angewandt werden.

[11] Im letzten Abschnitt wurden die *weak*-Regeln aus dem Kalkül $\mathcal{U}_S$ ausgeschlossen, aber behauptet, die Vollständigkeit des Verfahrens von Beckert übertrage sich auf den Kalkül $\mathcal{U}_S$ mit Gleichheit. Man sieht allerdings leicht, daß die *weak*-Regeln nicht für die Vollständigkeit, sondern nur für die Effizienz wichtig sind.

diesen Unifikator subsumiert) ein $\mathcal{U}_E$-Axiom ist. Die Kombination der Constraints, die zu den beiden Termen eines Unifikations-Problems gehören, findet ihre Entsprechung in der Charakterisierung von Satz 7.5.

An dieser Stelle soll noch einmal auf Abschnitt 6.7 verwiesen werden, wo erklärt wurde, wie bei der automatischen Gleichheitsbehandlung die durch den Benutzer-Constraint gegebenen Suchraum-Einschränkungen zu berücksichtigen sind.

Etwas unschön an dieser Einbettung von Rewrite-Regeln ist die Erzeugung von *zwei* Regeln pro Gleichung. Es ist möglich, dieses Problem auf der Präsentationsebene zu behandeln: zwei gerichtete Gleichungen der Form $(\forall \bar{u})(s \rightsquigarrow t \ll \langle \mathrm{id}, t \prec s \rangle)$ und $(\forall \bar{u})(t \rightsquigarrow s \ll \langle \mathrm{id}, s \prec t \rangle)$ werden einfach als $s \doteq t$ *angezeigt*.

Eine etwas weiter reichende Variante beruht auf der Feststellung, daß der Substitutionsteil der Constraints im Verfahren von Beckert bei zwei Rewrite Regeln $(\forall \bar{u})(s \rightsquigarrow t \ll \langle \nu_1, O_1 \rangle)$ und $(\forall \bar{u})(t \rightsquigarrow s \ll \langle \nu_2, O_2 \rangle)$ immer der gleiche ist, also $\nu_1 = \nu_2$. Dem Substitutionsteil entsprechen in unserer Constraintsprache die syntaktischen Gleichungen. Das System könnte diese korrektheits-relevanten Gleichungen und die korrektheits-irrelevanten Ordnungsconstraints getrennt halten, und ein Paar von gerichteten Gleichungen $\forall \bar{u}.(s \rightsquigarrow t) \ll C \sqcup O_1$ und $\forall \bar{u}.(t \rightsquigarrow s) \ll C \sqcup O_2$ als $\forall \bar{u}.(s \doteq t) \ll C$ anzeigen. Dadurch bekäme der Benutzer gerade den k.r. Teil des Constraints zu sehen.

In Abschnitt 6.3 wurde auf die Möglichkeit hingewiesen, die k.i. Constraints von den k.r. Constraints völlig getrennt zu halten, und auf Kalkülebene nur die k.r. Constraints beizubehalten. Die k.i. Constraints würden dann – wie andere Information zur Steuerung des automatischen Beweisers – im Hintergrund verwaltet werden. Bei dieser Variante macht es natürlich keinen Sinn mehr, die Gerichtetheit der Gleichungen syntaktisch zu reflektieren. Es werden einfach zu jeder universellen Gleichung *zwei* zusätzliche Ordnungsconstraints verwaltet, nämlich einer für jede Richtung.

Welche von diesen Varianten verfolgt wird, ist letztlich eine Frage der Interpretation der Begriffe „integriert" und „Kalkül". Wer sich für die Arbeitsweise der automatischen Gleichheitsbehandlung interessiert, wird die letzte Variante ablehnen, da in ihr die für das Verfahren ganz wesentlichen Ordnungsconstraints zur Hintergrundinformation „degradiert" werden. Freunde einer klaren Semantik werden sich hingegen bei den ersten Varianten daran stören, daß die Symbole $\doteq$ und $\rightsquigarrow$ sich nur durch ihre *operationale* Bedeutung und nicht durch eine Modellsemantik unterscheiden lassen. Ebensogut, werden sie argumentieren, könnte die maximale Anzahl von automatischen Quantor-Instantiierungen oder die Reihenfolge, in der Äste abgeschlossen werden, in den Kalkül aufgenommen werden. Der interaktiv beweisende Benutzer wird keine gerichteten Gleichungen sehen wollen, da Richtungseinschränkungen und k.i. Constraints für ihn nicht bindend sind. Bevor eine Implementierung in Angriff genommen wird, sollte diese Frage jedenfalls noch gründlich diskutiert werden. Der vorgestellte Rahmen bietet die Möglichkeit, verschiedene Ansätze mit geringem Aufwand zu einzubeziehen.

7.5.3. Vor- und Nachteile der Integration

Durch die Integration der Gleichheitsbehandlung in den Kalkül ergeben sich einige Vorteile gegenüber dem ursprünglichen Ansatz eines Hintergrundbeweisers. Der größte Vorteil geht natürlich vom Prinzip der Integration selber aus: Alle für einen Beweis relevante Information ist in derselben Sprache gehalten. Es ist keine Übersetzung nötig.

Aber auch für die „Verzahnung" der Suche von Beweisbaum und Gleichheitsbeweis können sich Vorteile ergeben: Beliebige Heuristiken können abwechselnd Gleichheits- und andere Regeln abwechselnd *im selben Beweis* anwenden. Die Beschreibung und Implementierung solcher Heuristiken wird dadurch vereinfacht.

Die Probleme eines inkrementellen Algorithmus und der damit verbundenen Buchhaltung entfallen größtenteils, da der derzeitige Stand der Vervollständigung in Form der generierten (gerichteten) Gleichungen Teil der Sequenzen ist.

Als Nachteil könnte sich der Verlust an Modularität herausstellen. Die $_3T^AP$-Implementierung des Verfahrens von Beckert stellt ein relativ eigenständiges Modul dar, das über eine generische Schnittstelle mit dem Tableaubeweiser verbunden ist. Laut [3] konnte diese Schnittstelle von einem früher verwendeten Verfahren ohne Änderungen übernommen werden. Bei einer direkten Einbettung der Gleichheitsbehandlung in den Kalkül werden die Komponenten sicherlich viel enger zusammenhängen, wodurch der Umstieg auf ein deutlich unterschiedliches Verfahren in der Implementierung wahrscheinlich mehr Probleme als in der theoretischen Behandlung bereitet. Auf die Frage, wie hier möglichst viel Modularität gewahrt werden kann, sollte bei der Planung einer Implementierung großer Wert gelegt werden.

7.6. Zusammenfassung

Es wurde die Verwendung von universellen Variablen in Tableaukalkülen erläutert, und vorgeschlagen, diese durch universelle Präfixe zu simulieren, um eine explizitere Darstellung zu erhalten. Davon ausgehend wurde der Kalkül $\mathcal{U}_E$ konstruiert, der aus dem Kalkül $\mathcal{M}_E$ durch Hinzufügen von universellen Präfixen entsteht. Auch für diesen Kalkül wurde eine Charakterisierung der Abgeschlossenheit durch Constraints geliefert. Es wurde gezeigt, daß der Kalkül $\mathcal{U}_E$ korrekt ist, und dessen für automatische Beweiser verwendbare Einschränkung $\mathcal{U}_S$ vollständig. Schließlich wurde diskutiert, wie das Verfahren von Beckert [4] in einem Beweissystem verwendet werden kann, das auf dem Kalkül $\mathcal{U}_E$ basiert, und auf einige Freiheitsgrade bei dieser Einbettung hingewiesen.

8. Hypergraphenkalkül

Die verschiedenen in dieser Arbeit bisher vorgestellten Kalküle sind Sequenzenkalküle: Beweise in diesen Kalkülen bestehen aus Bäumen, deren Knoten mit Sequenzen markiert sind. Roland Preiß schlägt in seiner Dissertation [17] Hypergraphen zur Darstellung von Beweisen vor. In diesem Kapitel soll erörtert werden, wie sich die vorgestellten Ansätze auf einen Hypergraphen-Kalkül übertragen lassen. In dieser Arbeit muß aus Platzgründen auf eine Einführung in den Hypergraphen-Kalkül verzichtet werden. Es wird daher in diesem Kapitel eine gewisse Vertrautheit mit dem in [17] beschriebenen Kalkül vorausgesetzt.

Die Diskussion eines Hypergraphen-Kalküls für ein integriertes Beweissystem gestaltet sich etwas schwierig, da bislang überhaupt nicht untersucht wurde ob, und in welcher Weise Hypergraphen für ein *automatisches* Beweissystem verwendet werden könnten. Die Untersuchungen von Preiß beziehen sich im wesentlichen auf eine redundanzfreie Darstellung von bestehenden Beweisen, und die Verwendung für eine Mischung aus formalen und informalen Beweisen. Das von Preiß implementierte System `groovy` erlaubt immerhin das interaktive Beweisen mit Hypergraphen, doch konnten wegen der Neuheit des Konzepts noch nicht so viele praktische Erfahrungen damit gesammelt werden wie mit anderen Kalkülen.

Es gibt jedoch Aspekte, die dafür sprechen, daß Hypergraphen gewinnbringend auch für ein automatisches System verwendet werden könnten: Insbesondere werden Redundanzen in der Reihenfolge von Regelanwendungen reduziert. Außerdem werden mehrfach auftretende Teilbeweise identifiziert. Diese Eigenschaften könnten zu einer bedeutenden Verkleinerung des Suchraums für einen automatischen Beweiser führen.

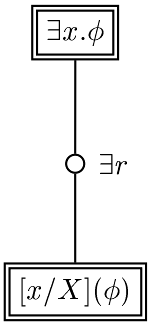
Für ein integriertes System bieten Hypergraphen die vielversprechende Möglichkeit, weitgehend unabhängig von der Reihenfolge von Regelanwendungen Teilbeweise zu löschen. Damit könnte der Benutzer an einem maschinell gefundenen Teilbeweis viel feinkörnigere Korrekturen vornehmen, als durch das Rücksetzen einer Baumstruktur erlaubt wird.

Preiß führt in seiner Diskussion ein ausgefeiltes Konzept zur Behandlung von Quantoren ein, das auf einer Abhängigkeitsrelation zwischen freien Variablen beruht. Dieses Verfahren ist natürlich nicht kompatibel mit den hier vorgestellten Konzepten von Metavariablen und ϵ-Termen. Wir beschränken uns daher darauf, die *aussagenlogische* Struktur der Hypergraphen zu nutzen, halten aber an Metavariablen und ϵ-Termen fest. Es soll nun für die verschiedenen in dieser Arbeit besprochenen Konzepte skizziert werden, wie diese in einem Hypergraphen-Kalkül verwendet werden können.

8.1. Metavariablen

Metavariablen können in einem Hypergraphen-Kalkül mit derselben Bedeutung eingeführt werden, wie es in dieser Arbeit beim Sequenzenkalkül geschehen ist: Metavariablen sind Platzhalter für Grundterme. Ein Hypergraphen-Beweis mit Metavariablen wird durch Anwendung einer Grundsubstitution für die Metavariablen zum Grund-Beweis.

Die $\exists r$-Regel von $\mathcal{M}_E$ nimmt also folgende Form an:



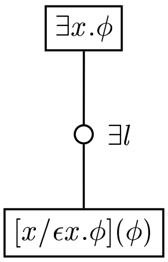
Etwas problematisch sind die Abschlußregeln, die im Grundkalkül folgende Form haben:

Die Anwendung dieser Regeln bereitet Probleme, wenn Metavariablen im Spiel sind: Der Abschluß ist zum Beispiel zwischen einem positiven Knoten $p(X)$ und einem negativen Knoten $p(a)$ erlaubt, falls X mit a instantiiert wird, und sonst nicht. Eine mögliche Lösung des Problems besteht darin, den Abschluß im Hypergraph nicht anzuwenden, wie ja auch Axiome in unseren Sequenzenkalkülen nicht angewandt werden. Der Abschlußtest, bzw. die Schnitteigenschaft müßten dann so erweitert werden, daß potentiell komplementäre Formeln identifiziert werden, und für diese ein bedingter Abschluß vorgenommen wird. Bei der Einbettung von Constraints wird sich allerdings ein eleganterer Weg anbieten, der den Abschlußcharakterisierungen von Satz 6.7, bzw. 7.5 entspricht, und die explizite Anwendung von Abschlußregeln erlaubt.

Für die Beweispräsentation können die Überlegungen in Abschnitt 6.6 offenbar übertragen werden: Die Darstellung von Metavariablen im Graphen kann vom aktuellen Benutzer-Constraint abhängig gemacht werden.

8.2. ϵ-Terme

Was ϵ-Terme angeht, ergeben sich überhaupt keine Schwierigkeiten. Die Hypergraphen-Version der $\exists l$-Regel lautet:



8.3. Constraints

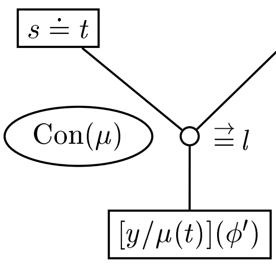
Der interessanteste Aspekt der Übertragung unserer Kalküle auf Hypergraphen betrifft sicherlich die Constraints. In den vorgestellten Sequenzenkalkülen wurden Constraints an Formeln geheftet. Man erinnere sich, daß in 6.1 kurz die Möglichkeit erwogen wurde, ganze Äste mit Constraints zu versehen. Diese Möglichkeit wurde verworfen, da sie zu grobkörnig ist: ist eine Regelanwendung eines Astes unter einer bestimmten Grundsubstitution für die Metavariablen ungültig, wird dadurch der ganze Ast ungültig. Daher wurden Constraints an Formeln geheftet.

Nun ist jedoch zu bemerken, daß Constraints eigentlich Einschränkungen an die Zulässigkeit von *Regelanwendungen* sind. Da Regeln auf der Beweisbaum-Ebene ganze Sequenzen betreffen, *mußten* wir die Constraints an die Formeln heften, um die ausgedrückten Einschränkungen lokal zu halten.

Bei einem Hypergraphen-Kalkül ist die Situation ganz anders: Es wäre zwar möglich, Constraints an die Formelknoten zu heften. Es würde jedoch einige Umstände bereiten, diese richtig zu verwalten: Ein Aspekt von Hypergraphen ist ja, daß Formelknoten, die aus unterschiedlichen Regelanwendungen stammen, aber dieselbe Formel enthalten, identifiziert werden. Die Formeln müßten dann mit jeweils einem Constraint pro Position in der Teilungsstruktur des Hypergraphen versehen werden.

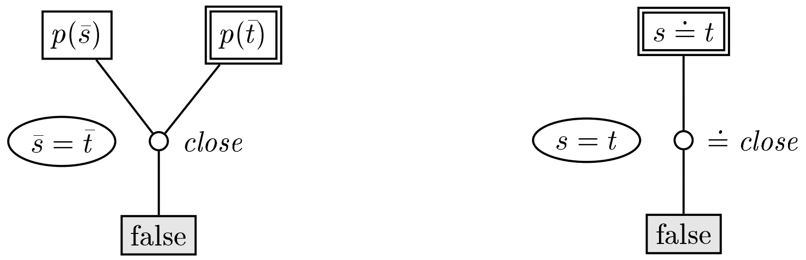
Andererseits ist es bei Hypergraphen aber möglich, Regelanwendungen mit Constraints zu versehen! Hypergraphen-Regeln betreffen nämlich nur noch einzelne Formeln. Unter einer gegebenen Grundsubstitution der Metavariablen werden dann nicht mehr die *Formeln* beibehalten, deren Constraint erfüllt ist, sondern die *Hyperkanten*.

Die Gleichheitsregeln (ohne universelle Variablen) haben damit folgende Form:



mit denselben Bezeichnungen wie in $\mathcal{M}_E$, siehe Abb. 6.1. Man bemerke, daß hier nur der zu μ gehörige Constraint benötigt wird: Die Constraints zu den beiden anderen Formeln stehen an den Regeln, aus denen diese gewonnen wurden.

Es lassen sich auch Constraint-Versionen der Abschlußregeln angeben, nämlich:



Diese Regeln können für beliebige atomare Formelknoten angewandt werden. Sie werden beim Abschlußtest nur für Substitutionen berücksichtigt, unter denen die Abschlüsse auch tatsächlich gültig sind. Man definiert:

> Ein Hypergraphen-Beweis mit Metavariablen und Constraints ist mit einer Substitution $\sigma \in \mathbf{Sub}^0$ geschlossen, wenn derjenige Graph die Schnitteigenschaft erfüllt, der entsteht, wenn nur Hyperkanten mit Constraints beibehalten werden, die von σ erfüllt werden.

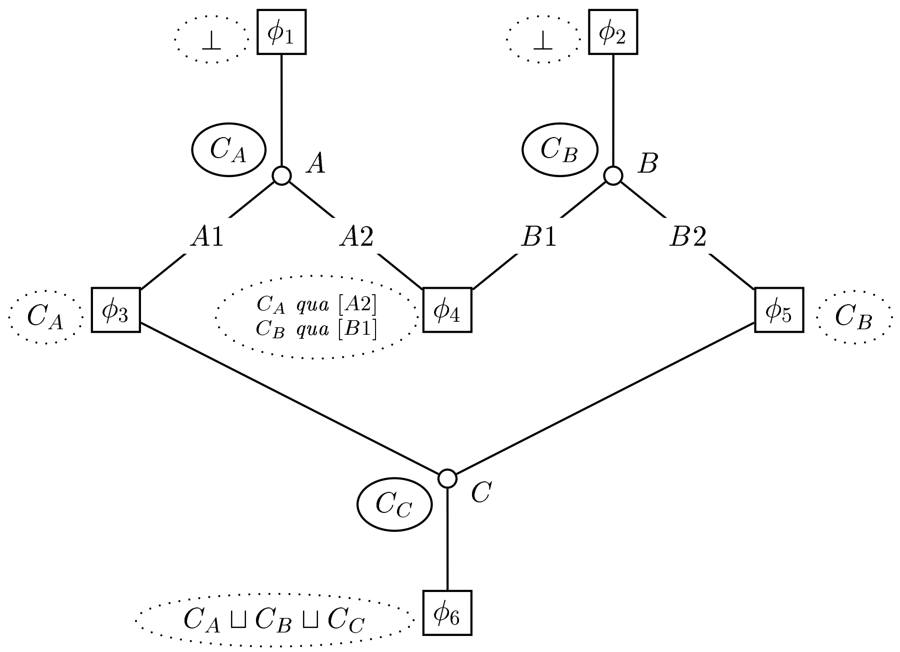
Es ist natürlich noch zu untersuchen, wie die Existenz eines σ, das diese Eigenschaft erfüllt, algorithmisch zu testen ist. Ein möglicher Ansatz besteht darin, den Formelknoten *akkumulierte Constraints* zuzuordnen: Man faßt für jede Position eines Knotens in der Teilungsstruktur konjunktiv die Constraints der Regelanwendungen zusammen, die von den Eingabeknoten zu ihm führen. Damit erhält letztlich jeder false-Knoten für jede seiner Positionen einen Constraint. Man bemerke, daß die *Positionen* von false-Knoten gerade *Ästen* in einem Sequenzenbeweis entsprechen, und die akkumulierten Constraints der false-Knoten gerade abschließende Constraints dieser Äste gemäß Def. 6.6 sind.

Der Abschlußtest eines Grund-Hypergraphen-Kalküls sucht nach einer Überdeckung der Teilungsstruktur, so daß in jedem Segment der Überdeckung ein false-Knoten liegt, ähnlich wie der Abschlußtest eines Grund-Sequenzen-Beweisers überprüfen muß, ob jeder Ast des Beweises abgeschlossen ist. Die Constraint-Version des Abschlußtests muß überprüfen, *ob es eine Überdeckung gibt, so daß in jedem Segment ein* false-*Knoten liegt, und die Konjunktion der akkumulierten Constraints dieser* false-*Knoten erfüllbar ist,* was genau dem auf S. 97 beschriebenen Algorithmus entspricht.

In Abb. 8.3 ist ein Beispiel für die Ermittlung von akkumulierten Constraints gegeben. Auf die eigentlichen Formeln und Regeln wird dabei kein Wert gelegt. ϕ_1 und ϕ_2 mögen Eingabeformeln sein, sie erhalten den akkumulierten Constraint $\perp$. Die Constraints C_A, resp. C_B der beiden oberen Regelanwendungen vererben sich auf die Formeln ϕ_3, resp. ϕ_5, die an den Positionen $[A1]$, resp. $[B2]$ liegen. Die Formel ϕ_4 hat *zwei* Positionen, nämlich $[A2]$ und $[B1]$, und demnach auch zwei akkumulierte Constraints. ϕ_6 hat schließlich die Position $[A1B2]$ und erhält damit die akkumulierten Constraints C_A und C_B der beiden Vorgängerformeln, sowie C_C von der Regelanwendung C.

Der akkumulierte Constraint einer Formel an einer bestimmten Position entspricht offenbar dem an diese Formel auf dem entsprechenden Ast in einem Sequenzenbeweis gehefteten Constraint. Der Unterschied besteht darin, daß es sich beim akkumulierten um eine *berechnete, abhängige* Information handelt. Die primitive Information liegt in den Constraints, die zu den Regeln gehören, und Abschlußeigenschaften, etc. lassen

Abbildung 8.1.: Hypergraphen-Beweis mit akkumulierten Constraints



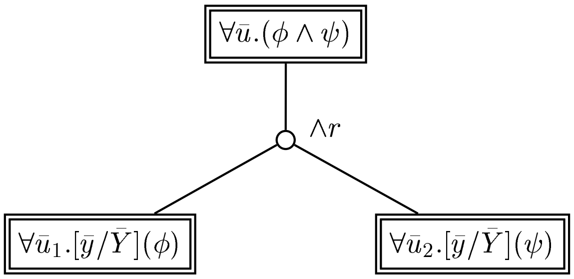
sich ohne die akkumulierten Constraints formulieren. Sie sind lediglich Teil einer Implementierung, die diese Eigenschaft testen kann. Die akkumulierten Constraints können im Hintergrund zusammen mit den Formeln verwaltet werden, und bei Regelanwendungen inkrementell aktualisiert werden. Die akkumulierten Constraints eines Formelknotens können sich nachträglich ändern, wenn etwa der Knoten durch eine spätere Formelanwendung eine zusätzliche Position erhält, oder durch Löschen einer Regelanwendung verliert. Die Constraints an einer Regelanwendung ändern sich nicht.

Außer für den Abschluß können die akkumulierten Constraints noch an einer weiteren Stelle gebraucht werden: Man entsinne sich, daß ein automatischer Beweiser in unseren Sequenzenkalkülen sinnvollerweise nur Regeln anwenden wird, die zu Constraints führen, die vom Benutzer-Constraint subsumiert werden, da die entstehenden Formeln sonst außerhalb des vom Benutzer vorgegebenen Suchraums liegen. Insbesondere werden nur Formeln mit erfüllbaren Constraints erzeugt, da andere nie für einen Abschluß verwendet werden können.

Die gleiche Taktik kann bei einem Hypergraphen-Beweiser verwendet werden, indem die akkumulierten Constraints betrachtet werden: Ebenso wie ein automatischer Beweiser keine Regel auf Formeln aus unterschiedlichen Zweigen einer Fallunterscheidung anwenden würde, da der resultierende Formelknoten dann keine legale Position hätte, würde er keine Regel anwenden, die eine Formel erzeugt, die in allen Positionen unerfüllbare Constraints hat. Der Benutzer-Constraint kann zur Suchraumeinschränkung verwendet werden, indem der Automat nur Formeln erzeugt, deren akkumulierter Constraint vom Benutzer-Constraint subsumiert wird.

8.4. Universelle Präfixe

Die Verwendung von universellen Präfixen in Hypergraphen bereitet ebensowenig Schwierigkeiten wie die ϵ-Terme. Die $\wedge r$-Regel erhält beispielsweise folgende Form:



wobei $\bar{u}_1 := \bar{u} \cap (\mathrm{fv}(\phi) \setminus \mathrm{fv}(\psi))$,
$\bar{u}_2 := \bar{u} \cap (\mathrm{fv}(\psi) \setminus \mathrm{fv}(\phi))$,
$\bar{y} := \bar{u} \cap (\mathrm{fv}(\phi) \cap \mathrm{fv}(\psi))$ und
$\bar{Y}$ neue Metavariablen sind.

8.5. Zusammenfassung

Es wurde gezeigt, wie die vorgeschlagenen Konzepte für einen Kalkül zum integrierten automatischen und interaktiven Beweisen in einem Hypergraphen-Kalkül verwendet werden können. Durch die Einführung von akkumulierten Constraints wurde ein Weg gewiesen, um algorithmische Probleme beim Abschlußtest zu lösen, sowie gewisse Fragestellungen zur automatischen Benutzung des Kalküls zu beantworten.

Es bleibt zu untersuchen, ob Hypergraphen-Kalküle überhaupt für automatische Beweissysteme geeignet sind. Dazu müßte insbesondere ein inkrementeller Abschlußtest erarbeitet werden, der dem Ast-für-Ast Abschluß von Tableaubeweisern entspricht. Den Abschlußtest für den gesamten Beweis nach jeder Regelanwendung aufzurufen wäre sicherlich zu ineffizient. Auch die Brauchbarkeit der akkumulierten Constraints wurde noch nicht nachgewiesen: es muß etwa noch festgestellt werden, wie groß diese in der Praxis werden.

9. Fazit

In dieser Arbeit wurden Fragestellungen untersucht, die sich beim Entwurf eines integrierten automatischen und interaktiven Beweisystems auf Basis eines gemeinsamen Kalküls ergeben. Zentrales Thema war dabei die Gestaltung jenes Kalküls in Hinblick auf die unterschiedlichen Anforderungen, die sich aus dessen interaktivem und automatischem Gebrauch ergeben.

Nachdem in Kapitel 2 diese Anforderungen identifiziert wurden, wurde eine Strukturierung der Arbeit gerechtfertig, die sich an denjenigen Aspekten eines Kalküls für *automatische* Beweissysteme orientiert, die für ein *interaktives* System unerwünscht sind. Im Hauptteil der Arbeit wurden Verfahren vorgeschlagen, um in diesen Punkten den Anforderungen beider Seiten gerechtzuwerden. Im einzelnen wurden folgende Vorschläge zum Entwurf eines gemeinsamen Kalküls vorgebracht, und eingehend diskutiert:

- **Metavariablen** ersetzen die freien Variablen des Tableaukalküls. Während die Semantik der freien Variablen auf Belegungen basiert, sind Metavariablen Platzhalter für Grundterme. Für die Beweispräsentation können sie durch vom Benutzer angegebene intendierte Instantiierungen dargestellt werden.

- **ϵ-Terme** stellen eine Alternative zu den verschiedenen beim automatischen Beweisen verwendeten Skolemisierungsverfahren dar. Sie ermöglichen kürzere Beweise und haben eine intuitivere Semantik als automatisch eingeführte Skolemfunktionen mit willkürlichen Namen.

- **Constraints** werden von bestehenden Verfahren zur automatischen Gleichheitsbehandlung übernommen. Es wurde erläutert, wie der zusätzliche formale Ballast, der durch ihre Verwendung auf der Kalkülebene entsteht, auf der Präsentationsebene eliminiert werden kann.

- **Universelle Präfixe** machen die Behandlung von universellen Variablen explizit. Sie sind nicht mehr Hintergrundinformation, sondern syntaktischer Bestandteil des Kalküls.

Es wurden jeweils konkrete Kalküle vorgestellt, in denen die vorgeschlagenen Techniken verwendet werden. In den zuletzt definierten Kalkülen $\mathcal{U}_S$, bzw. $\mathcal{U}_E$ sind alle vier Vorschläge verwirklicht.

Die *Motivationen* der einzelnen Konzepte sind jedoch, wie die Konzepte selber, weitgehend orthogonal: Sollten sich zum Beispiel schwerwiegende Argumente *gegen* die Verwendung von ϵ-Termen ergeben, können die anderen Konzepte dennoch beibehalten werden. Andererseits könnten ϵ-Terme und die Handhabung universeller Präfixe statt universeller Variablen auch für ein rein interaktives System Vorteile haben.

Die Übertragbarkeit der erarbeiteten Methoden auf einen Nicht-Sequenzen-Kalkül wurde anhand einer Diskussion über deren Verwendung in einem Hypergraphen-Kalkül im Sinne von [17] demonstriert.

Abgerundet wird die Arbeit durch eine kurze Erörterung zur Implementierung von ϵ-Logiken, sowie einen Anhang zur Einführung von ϵ-Termen durch Hilbert in [13].

Durch die Untersuchung der Kalkülebene wurde ein Fundament zur Entwicklung eines integrierten Beweissystems mit gemeinsamem Kalkül entworfen. Bevor jedoch eine Implementierung auf dieses Fundament gebaut werden kann, sind noch weitere „architektonische“ Untersuchungen nötig:

- Eine Benutzerschnittstelle ist zu entwerfen, die die flexible Handhabung des Benutzer-Constraints und des Beweisbaums ermöglicht. Sie muß Mechanismen zur Verfügung stellen, die es dem Benutzer ermöglichen, bequem zwischen verschiedenen Sichtweisen auf den Kalkül zu wechseln.
- Es muß ein Modell für die Interaktion zwischen menschlichem und automatischem Beweiser geschaffen werden. Welche Möglichkeiten hat der Mensch, um dem Automaten bei der Arbeit zu helfen? Wie können vom Automaten gefundene Teilbeweise vom Menschen weiterverwendet werden? Sollen Mensch und Automat abwechselnd an einem Beweis arbeiten, oder ist ein nebenläufiges System denkbar, in dem Mensch und Maschine gleichzeitig arbeiten, und die Maschine im Hintergrund die Validierung der vom Menschen durchgeführten Schritte übernimmt?
- Ein auf dem Hypergraphen-Kalkül aufbauendes System könnte gegenüber den üblichen, auf baumorientierten Kalkülen aufbauenden Systemen erhebliche Vorteile haben. Es ist jedoch noch zu untersuchen, ob und wie der Hypergraphen-Kalkül überhaupt für ein automatisches Beweissystem verwendet werden kann.
- Die vorliegende Untersuchung beschränkt sich auf die Prädikatenlogik erster Stufe mit Gleichheit und ϵ-Termen. Auch für andere Logiken, wie etwa Modal- und Temporallogiken, Logiken höherer Stufe, etc. existieren jedoch automatische und interaktive Systeme. Eine Integration auch für solche Logiken zu unternehmen, ist eine große Herausforderung.

Es ist zu hoffen, daß für die verbleibenden Probleme Lösungen gefunden werden, die zur Implementierung eines integrierten Beweissystems führen, in dem die Stärken von automatischen und interaktiven Systemen zusammenfließen.

A. ι und η: Die Vorfahren der ϵ-Terme

Da ϵ-Terme in der vorliegenden Arbeit eine recht zentrale Rolle spielen, soll in diesem Anhang kurz auf ihre Herkunft eingegangen werden.

In §8 des ersten Bands von [13], – „Der Begriff *derjenige, welcher* und seine Eliminierbarkeit“ – führt Hilbert die ι-Terme ein. Deren Bezeichnung ist anscheinend von Russel und Whitehead übernommen. Die ι-Terme haben die Form[1]

$$\iota x.\phi(x),$$

zu lesen als „dasjenige Ding x, für welches ϕ besteht.“ In dieser Lesart ist schon verankert, daß erstens überhaupt ein solches existiert, also

$$\exists x.\phi(x), \tag{E}$$

und ferner, daß *nur* eines existiert, also

$$\forall x, y.(\phi(x) \wedge \phi(y) \to x = y). \tag{I}$$

Somit werden solche Terme nur zugelassen, wenn diese *Unitätsformeln* (E) und (I) bereits bewiesen sind. Diese Forderung wird im Rahmen eines Hilbert-Kalküls verständlich: Dort werden aus Axiomen mit Hilfe von Regeln sukzessive wahre Formeln abgeleitet. Wenn erst gezeigt wurde, daß (E) und (I) für eine Formel ϕ und eine Variable x gelten, dann darf im Rest des Beweises der Term $\iota x.\phi(x)$ verwendet werden, und ferner die Aussage $\phi(\iota x.\phi(x))$. Man erhält also eine Beweisregel

$$\frac{\begin{array}{c}\exists x.\phi(x)\\ \forall x, y.(\phi(x) \wedge \phi(y) \to x = y)\end{array}}{\phi(\iota x.\phi(x))}\ \iota$$

Verfolgt man, wie Hilbert, einen axiomatischen Ansatz, ist eine solche Definition natürlich problemlos möglich. Die heute übliche Trennung von Syntax und Modellsemantik muß jedoch an den ι-Termen scheitern: Ein ι-Term darf überhaupt nur *syntaktisch* gebildet werden, wenn die zugehörige Formel gewisse *semantische* Eigenschaften hat. Ohne eine Modellsemantik, die beliebigen Termen unter *jeder* Interpretation einen Wert zuordnen muß, kann die Semantik allerdings nur anhand der Ableitbarkeit definiert werden. Korrektheit und Vollständigkeit sind damit gratis, aber die Logik trifft nur noch Aussagen über sich selbst.

[1] Die Schreibweise ist hier etwas modernisiert. Bei Hilbert heißt es $\iota_x\mathfrak{A}(x)$, was etwas ungewohnt zu lesen ist.

Für Hilberts Arbeit wäre eine Modellsemantik nicht angemessen gewesen, da er die Grundlagen der Mathematik erörterte. Für ihn waren die logischen Schlüsse tatsächlich primitiv, es existierte keine Meta-Ebene, auf der über Modelle hätte geredet werden können.[2]

Ein Beispiel für eine Anwendung der ι-Terme bietet folgende Axiomatisierung einer Gruppe:

$$\begin{array}{ll} (a) & \forall x, y, z. x(yz) = (xy)z \\ (e\&i) & \exists e.(\quad \forall x.(ex = x \ \wedge \ xe = x) \\ & \qquad \wedge \forall x. \exists i.(xi = e \ \wedge \ ix = e)) \end{array}$$

Daraus lassen sich die Formeln

$$\begin{array}{l} \exists e. \forall x.(ex = x \ \wedge \ xe = x) \\ \forall y, z.((\forall x.(yx = x \ \wedge \ xy = x)) \wedge (\forall x.(zx = x \ \wedge \ xz = x)) \ \rightarrow \ y = z) \end{array}$$

ableiten. Diese sind die Unitätsformeln für den Term $\iota x. \forall x.(ex = x \ \wedge \ xe = x)$, welcher das eindeutig definierte neutrale Element der Gruppe bezeichnet. Kürzt man diesen Term durch E ab, kommt man auf ähnliche Weise zu dem Term $\iota i.(xi = E \ \wedge \ ix = E)$, der zu jedem x das eindeutig bestimmte inverse Element bezeichnet. Man kann für diesen Term zum Beispiel $I(x)$ schreiben, und hat damit effektiv eine Funktion eingeführt. Mit diesen Symbolen erhält man die übliche Axiomatisierung von Gruppen:

$$\begin{array}{ll} (A) & \forall x, y, z. x(yz) = (xy)z \\ (E) & \forall x.(Ex = x \ \wedge \ xE = x) \\ (I) & \forall x.(xI(x) = E \ \wedge \ I(x)x = E) \end{array}$$

Wie man sieht, wurden sämtliche Existenzquantoren aus der Axiomatisierung eliminiert. Faßt man freie Variablen als allquantifiziert auf, können auch die Allquantoren weggelassen werden, und man erhält durch die Einführung der neuen, durch ι-Terme definierten, Konstanten- und Funktionssymbole eine quantorfreie Formulierung.

Gerade dies war nun Hilberts Absicht, denn er wollte die Widerspruchsfreiheit seines Formalismus auf die Widerspruchsfreiheit eines Formalismus ohne gebundene Variablen zurückführen. Doch dazu reichen die ι-Terme nicht, denn nicht zu jeder Existenz-Formel gilt die zugehörige andere Unitätsformel (I).

Hilbert geht also in Band II, §1.2 von der „symbolischen Auflösung“ von Existenzquantoren aus: Wenn eine Formel $\forall u. \exists x. \phi(u, x)$ gültig ist, so liefert das Auswahlaxiom eine Funktion, die zu jedem u ein x liefert, so daß $\phi(u, x)$ wahr wird. Man kann nun dieser Funktion einen neuen Namen f geben, und erhält so die Formel $\forall u. \phi(u, f(u))$, aus der der Existenzquantor eliminiert ist. Diese Einführung einer neuen Funktion wird nun mittels des η-Operators ausgedrückt: Der Term

$$\eta x. \phi(x)$$

darf verwendet werden, sobald die Formel

$$\exists x. \phi(x), \tag{E}$$

bewiesen wurde. Die zugehörige Regel lautet:

$$\frac{\exists x. \phi(x)}{\phi(\eta x. \phi(x))} \ \eta$$

[2]Man bemerke, wie sich das $=$-Zeichen, statt dem $\doteq$ in (I) eingeschlichen hat!

Diese η-Terme sind schon recht nahe an unseren ϵ-Termen. Insbesondere stellt sich hier bereits die Frage, in welchem Verhältnis η-Terme zu unterschiedlichen Formeln stehen. Erhält man etwa $\eta x.p(x,a)$, wenn man a für y in $\eta y.p(x,y)$ einsetzt? Mit anderen Worten: sind η-Terme substitutiv? Hilbert macht auf diese Frage aufmerksam, die Antwort geht aus dem Text allerdings nicht klar hervor, aber es ist anzunehmen daß Hilbert an eine substitutive Semantik dachte. Man bedenke, daß Hilbert mangels einer Modellsemantik nicht von Belegungen, und daher schon gar nicht von Substitutivität sprechen konnte. Es ist eine Frage der mathematischen Intuition und nicht Substitution, daß man $f(a)$ erhält, wenn man a für x in $f(x)$ einsetzt. Explizit verlangt wird jedoch die Stabilität gegenüber gebundener Umbenennung.

Die η-Terme haben immer noch einen Nachteil: Sie dürfen nicht verwendet werden, bevor die Formel (E) bewiesen ist. Dieser Nachteil kann jedoch durch einen kleinen Kunstgriff umgangen werden: Man definiert neue Symbole $\epsilon x.\phi(x)$ durch die Gleichung

$$\epsilon x.\phi(x) \;=\; \eta x.(\exists y.\phi(y) \rightarrow \phi(x)).$$

Man[3] überzeugt sich leicht, daß die zu dem η-Term gehörige Formel

$$\exists x.(\exists y.\phi(y) \rightarrow \phi(x))$$

unabhängig von der Formel ϕ gültig ist, und daher die Verwendung des η-Terms immer erlaubt ist. Man erhält

$$\dfrac{\exists y.\phi(y) \qquad \dfrac{\overline{\exists x.(\exists y.\phi(y) \rightarrow \phi(x))}}{\exists y.\phi(y) \rightarrow \phi(\eta x.(\exists y.\phi(y) \rightarrow \phi(x)))}\,\eta}{\phi(\epsilon x.\phi(x))}\,\rightarrow$$

Daraus ergibt sich, wie erwünscht, durch Variablenumbenennung die abgeleitete Regel

$$\dfrac{\exists x.\phi(x)}{\phi(\epsilon x.\phi(x))}\,\epsilon$$

Aber im Gegensatz zu den η-Termen können ϵ-Terme immer verwendet werden, auch ohne vorher die Bedingung (E) nachzuweisen.

In den meisten Lehrbüchern der Logik – [14] ist hier eine Ausnahme – werden ϵ-Terme (wenn überhaupt) nur im Zusammenhang mit Hilberts Widerspruchsfreiheits-Beweisen erwähnt. Die angenehme Eigenschaft, auch ohne den Nachweis von Unitätsformeln verwendbar zu sein, macht es möglich, für die ϵ-Terme eine moderne Modellsemantik anzugeben, die die Grundlage für Kapitel 5 dieser Arbeit bildet.

[3] Der Leser kann dazu etwa einen der in dieser Arbeit eingeführten ϵ-Kalküle verwenden :-) Es ist sehr erhellend, sich klar zu machen, *welches* x es ist, das hier existiert.

B. Zur Implementierung der ϵ-Logik

Bei der Einführung der ϵ-Terme ist eine wichtige Frage unbeantwortet geblieben, nämlich ob es eine Möglichkeit gibt, diese effizient zu implementieren. In diesem Abschnitt soll gezeigt werden, daß auf einer einfachen, auf Graphen basierenden Repräsentation von Termen Regelanwendungen genauso effizient möglich sind, wie wenn Skolemfunktionen verwendet werden.

B.1. Repräsentation durch Bäume und Graphen

Die einfachste Repräsentation für Formeln und Terme sind verzeigerte Strukturen. Variablen, Junktoren, Prädikate, etc. werden durch „Zellen“ repräsentiert, die den Typ der Entität und Verweise auf eventuelle Teilterme enthalten. Wir stellen solche Strukturen als Baum dar: Die Formel $\forall x.p(f(b,x),g(x,a))$ wird zum Beispiel wie folgt repräsentiert:

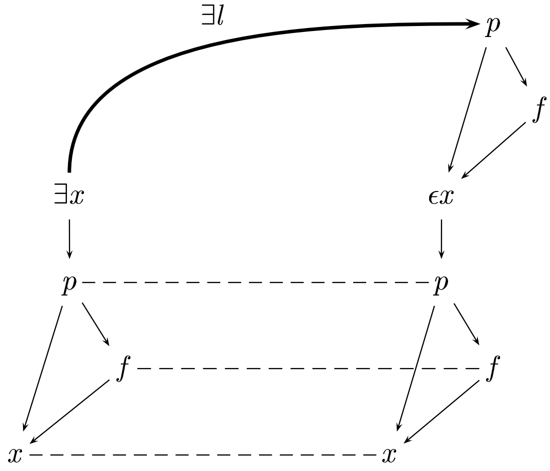
$$
\begin{tikzcd}
& & & \forall x \arrow[d] & & & \\
& & & p \arrow[dll] \arrow[drr] & & & \\
& f \arrow[dl] \arrow[dr] & & & & g \arrow[dl] \arrow[dr] & \\
b & & x & & x & & a
\end{tikzcd}
$$

Eine geschicktere Repräsentation verwendet gerichtete, azyklische Graphen. Dadurch können mehrfach (bzw. in verschiedenen repräsentierten Formeln) auftretende Teilterme identifiziert werden. Die obige Formel wird dann repräsentiert durch:

$$
\begin{tikzcd}
& & \forall x \arrow[d] & & \\
& & p \arrow[dl] \arrow[dr] & & \\
& f \arrow[dl] \arrow[dr] & & g \arrow[dl] \arrow[dr] & \\
b & & x & & a
\end{tikzcd}
$$

Durch Verwendung einer Hashtabelle kann bei dieser Darstellung erreicht werden, daß zwei Terme genau dann gleich sind, wenn sie durch dieselbe Zelle repräsentiert werden.

Wie verhalten sich nun solche Graph-Repräsentationen unter Regelanwendungen? Man betrachte etwa die Sequenz $\exists x.p(x, f(x)) \vdash$. Durch Anwendung der $\exists l$-Regel wird hieraus $p(\epsilon x.p(x, f(x)), f(\epsilon x.p(x, f(x)))) \vdash$. Das doppelte Auftreten des ϵ-Terms wird durch die Graph-Darstellung abgefangen, man erhält:



Die die Formel $p(x, f(x))$ repräsentierende Struktur kann für den ϵ-Term wiederverwendet werden. Für die neue Formel muß die Struktur kopiert werden, wobei an den entsprechenden Stellen die Referenzen auf die Variable x zu Referenzen auf den ϵ-Term werden. Der Aufwand hierfür ist im schlimmsten Falle linear in der Größe der *Repräsentation* der Formel. Die Strukturen für eventuell auftretende Teilformeln, in denen die Variable x gar nicht vorkommt, brauchen natürlich nicht repliziert zu werden.

Herkömmliche Skolemisierungsverfahren sind genauso aufwendig: auch sie müssen Formeln replizieren, um Variablen durch Skolemsymbole zu ersetzen. Bei der $_3T^AP$-Implementierung der δ^{++}-Regel geschieht sogar etwas sehr ähnliches, wie hier für ϵ-Terme vorgeschlagen wird: Das Skolemsymbol $f_{[\phi]}$ für eine Formel ϕ wird intern tatsächlich mittels eines Zeigers auf die Repräsentation der Formel ϕ benannt. Der einzige Unterschied zur jetzigen Situation besteht darin, daß diese referierte Formel nun tatsächlich syntaktischer Teil des Terms, und mit einer Semantik versehen ist.

Die Anwendung der $\exists r$-Regel erfordert ebenfalls ein Durchlaufen des Graphen, um die quantifizierte Variable durch eine Metavariable zu ersetzen. Die $\wedge$- und $\neg$-Regeln können offenbar in konstanter Zeit angewandt werden. Aufwendiger ist die Anwendung der $\doteq$-Regeln: Wenn in der obigen Formel, die zwei identische ϵ-Terme enthält, in einem der ϵ-Terme eine Gleichung angewandt werden soll, muß die Struktur für diesen ϵ-Term vor der Anwendung kopiert werden, da sich die Anwendung sonst auf *alle* Vorkommnisse des ϵ-Terms bezieht, was im allgemeinen nicht erwünscht ist.

Der Aufwand für diese Regelanwendungen ist jedoch unabhängig von den ϵ-Termen, er ist der gleiche wie bei anderen Skolemisierungsverfahren.

Die Anwendung der *Regeln* wird also durch die ϵ-Terme nicht aufwendiger! Ganz umsonst bekommt man die in 6.4.3 nachgewiesene größere Mächtigkeit der ϵ-Terme allerdings nicht: Um den dort angegebenen Beweis schließen zu können, muß das System die Erfüllbarkeit des Constraints

$$X = \epsilon x.\neg p_n(x, Y, b) \ \sqcup \ Y = a \ \sqcup \ X = \epsilon x.\neg p_n(x, a, Z) \ \sqcup \ Z = b$$

feststellen. Dazu ist es unabdingbar, die beiden zu den ϵ-Termen gehörigen Strukturen zu durchwandern, und zu vergleichen. Bei δ^{+^+}-Skolemisierung stünden an Stelle der ϵ-Terme flache Skolemterme, die schnell durchlaufen sind. Aber dafür kann der Beweis bei δ^{+^+}-Skolemisierung hier gar nicht abgeschlossen werden!

Die Graph-Darstellung bietet die Möglichkeit, den Abschlußtest mit einem „Schalter“ zu versehen: Sollen Abschlüsse gefunden werden, die die Analyse von ϵ-Termen erfordern, so muß der zusätzliche Aufwand in Kauf genommen werden. Ansonsten kann der Abschlußtest so eingestellt werden, daß er nicht versucht, ϵ-Terme miteinander zu unifizieren, wenn diese auf unterschiedliche Formeln verweisen. Damit ist er genauso stark wie ein Kalkül mit δ^{+^+}-Skolemisierung.

Übrigens ist zu erwarten, daß der Aufwand zur Durchwanderung von ϵ-Termen in der Praxis recht klein ist: Erfahrungen mit bestehenden automatischen Beweissystemen zeigen, daß der Unifikationsalgorithmus meistens mit *nicht* unifizierbaren Termen aufgerufen wird. Um die Nicht-Unifizierbarkeit festzustellen, brauchen die Terme aber nur durchlaufen zu werden, bis zwei Teilterme auftreten, deren oberste Funktionssymbole verschiedenen sind. Dies geschieht meist recht schnell. Diese Situation dürfte sich auf ein System, das ϵ-Terme verwendet, übertragen. Das hieße, daß beim Erfüllbarkeitstest von abschließenden Constraints meistens nur ein kleiner Teil der ϵ-Termen durchwandert werden braucht, bevor festgestellt wird, daß der Constraint unerfüllbar ist.

B.2. Zusammenfassung

Es wurde gezeigt, daß eine Darstellung von Formeln als gerichtete Graphen möglich ist, die Regelanwendungen mit linearem Speicher- und Zeitaufwand in der Größe der Formeln erlaubt.

Ferner wurde angedeutet, wie der Aufwand beim Abschlußtest so gering gehalten werden kann wie bei der δ^{+^+}-Regel, wenn man auf die zusätzliche Mächtigkeit der ϵ-Terme verzichtet.

C. Verzeichnis der beschriebenen Kalküle

Dieser Anhang soll einen Überblick über die verschiedenen in dieser Arbeit vorgestellten Kalküle liefern.

Die Kalküle heißen $\mathcal{G}_S$, $\mathcal{G}_E$, $\mathcal{M}_S$, $\mathcal{M}_E$, $\mathcal{U}_S$, $\mathcal{U}_E$. Dabei sind die beiden $\mathcal{G}$-Kalküle Grundkalküle für die ϵ-Logik. Die $\mathcal{M}$-Kalküle sind zusätzlich mit Metavariablen und Constraints versehen. Die $\mathcal{U}$-Kalküle verwenden schließlich universelle Präfixe. Die Indizes beziehen sich auf die den ϵ-Termen zukommende Semantik. Der Index S bezeichnet die Kalküle für substitutive Semantik. Diese sind außerdem mit Einschränkungen versehen, die sie für den automatischen Beweiser geeignet machen. Der Index E wird für die Kalküle mit extensionaler Semantik verwendet, die für den interaktiven Gebrauch vorgesehen sind. Zum Verhältnis von substitutiver Semantik und extensionaler Semantik innerhalb *eines* Beweissystems, siehe die Diskussion im ersten Absatz von Abschnitt 5.6.3, S. 55.

Die folgende Tabelle gibt eine Übersicht darüber, wo welcher Kalkül definiert wird:

Kalkül	Definition	Regeln
$\mathcal{G}_S$	Def. 5.17, S. 47	Abb. 5.2 S. 47
$\mathcal{G}_E$	Def. 5.21, S. 57	Abb. 5.3 S. 57
$\mathcal{M}_S$	Satz 6.10, S. 85	siehe $\mathcal{M}_E$
$\mathcal{M}_E$	Def. 6.5, S. 77	Abb. 6.1 S. 78
$\mathcal{U}_S$	Satz 7.7, S. 117	siehe $\mathcal{U}_E$
$\mathcal{U}_E$	Def. 7.3, S. 114	Abb. 7.1 S. 113

Literaturverzeichnis

[1] Ahrendt, Wolfgang, Bernhard Beckert, Reiner Hähnle, Wolfram Menzel, Wolfgang Reif, Gerhard Schellhorn und Peter H. Schmitt: *Integration of Automated and Interactive Theorem Proving.* In: Bibel, W. und P. Schmitt (Herausgeber): *Automated Deduction — A Basis for Applications*, Band II, 4, Seiten 97–116. Kluwer, 1998.

[2] Bachmair, Leo, Nachum Dershowitz und Jieh Hsiang: *Orderings for Equational Proofs.* In: *Proceedings, Symposium on Logic in Computer Science (LICS), Boston, USA*, Seiten 346–357, 1986.

[3] Beckert, Bernhard: *Ein vervollständigungsbasiertes Verfahren zur Behandlung von Gleichheit im Tableaukalkül mit freien Variablen.* Diplomarbeit, Fakultät für Informatik, Universität Karlsruhe, Juli 1993.

[4] Beckert, Bernhard: *A Completion-Based Method for Mixed Universal and Rigid E-Unification.* In: Bundy, A. (Herausgeber): *Proceedings, 12th International Conference on Automated Deduction (CADE), Nancy, France*, LNCS 814, Seiten 678–692. Springer, 1994.

[5] Beckert, Bernhard und Reiner Hähnle: *Analytic Tableaux.* In: Bibel, Wolfgang und Peter H. Schmitt (Herausgeber): *Automated Deduction — A Basis for Applications*, Band I: Foundations, Seiten 11–41. Kluwer, Dordrecht, 1998.

[6] Beckert, Bernhard, Reiner Hähnle, Peter Oel und Martin Sulzmann: *The Tableau-based Theorem Prover $_3T^AP$, Version 4.0.* In: *Proceedings, 13th International Conference on Automated Deduction (CADE), New Brunswick, NJ, USA*, LNCS 1104, Seiten 303–307. Springer, 1996.

[7] Beckert, Bernhard, Reiner Hähnle und Peter H. Schmitt: *The Even More Liberalized δ-Rule in Free Variable Semantic Tableaux.* In: Gottlob, G., A. Leitsch und D. Mundici (Herausgeber): *Proceedings, 3rd Kurt Gödel Colloquium (KGC), Brno, Czech Republic*, LNCS 713, Seiten 108–119. Springer, 1993.

[8] Degtyarev, A. und A. Voronkov: *What you always wanted to know about rigid E-unification.* Journal of Automated Reasoning, 20(1):47–80, 1998.

[9] Fine, K.: *Reasoning With Arbitrary Objects.* Blackwell, Oxford, 1985.

[10] Fitting, Melvin C.: *First-Order Logic and Automated Theorem Proving.* Springer, 1990.

[11] Hähnle, Reiner und Peter H. Schmitt: *The Liberalized δ-Rule in Free Variable Semantic Tableaux.* Journal of Automated Reasoning, 13(2):211–222, 1994.

[12] Heisel, M., W. Reif und W. Stephan: *Tactical theorem proving in program verification.* In: *Proceedings of the 10th International Conference on Automated Deduction*, Band 449 der Reihe *Lecture Notes in Computer Science*, Seiten 117–131. Springer Verlag, 1990.

[13] Hilbert, D. und P. Bernays: *Grundlagen der Mathematik.* Springer Verlag, 1939. Zweite Auflage 1970.

[14] Leisenring, A.C.: *Mathematical Logic and Hilbert's ϵ-Symbol.* MacDonald, London, 1969.

[15] Meyer Viol, W.P.M.: *Instantial Logic. An Investigation into Reasoning with Instances.* Nummer DS–1995–11 in *ILLC Dissertation Series.* Universiteit van Amsterdam, 1995.

[16] Nieuwenhuis, Robert und Albert Rubio: *AC-superposition with constraints: no AC-unifiers needed.* In: Bundy, A. (Herausgeber): *Proceedings of the 12th International Conference on Automated Deduction*, Band 814 der Reihe *Lecture Notes in Computer Science*, Seiten 545–559. Springer Verlag, 1994.

[17] Preiss, Roland: *Beweisvisualisierung und -analyse mit Hypergraphen.* Doktorarbeit, Universität Karlsruhe, 1998.

[18] Reif, W.: *The KIV-Approach to Software Verification.* In: Broy, M. und S. Jähnichen (Herausgeber): *KORSO: Methods, Languages, and Tools for the Construction of Correct Software*, Band 1009 der Reihe *Lecture Notes in Computer Science*, Seiten 339–368. Springer Verlag, 1995.

[19] Stenz, Gernot: *Beweistransformation in Gentzenkalkülen.* Diplomarbeit, Fakultät für Informatik, Universität Karlsruhe, 1997.

[20] Vigneron, Laurent: *Associative-Commutative Deduction with Constraints.* In: Bundy, A. (Herausgeber): *Proceedings of the 12th International Conference on Automated Deduction*, Band 814 der Reihe *Lecture Notes in Computer Science*, Seiten 530–544. Springer Verlag, 1994.