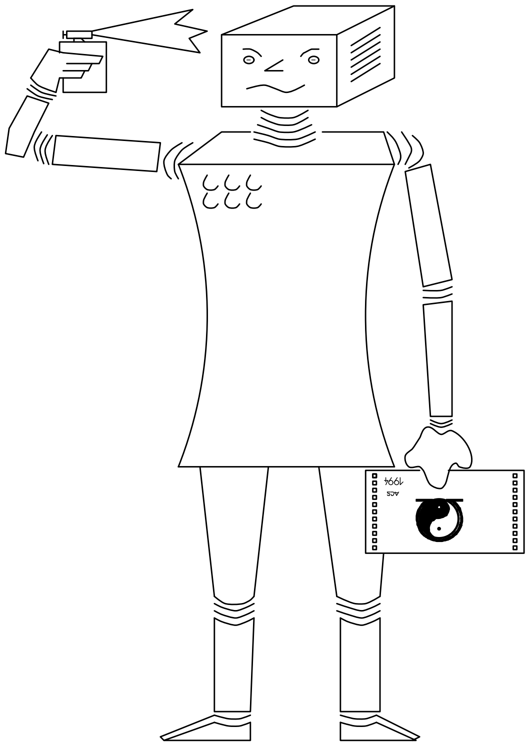


Sortenlogiken und ihre Automatisierung

Manfred Kerber

Published as: Script of a lecture, Summer 1994, Fachbereich Informatik, Universität des Saarlandes, Saarbrücken, Germany

Sortenlogiken und ihre Automatisierung

– Skript zur Vorlesung –

Sommersemester 1994

Manfred Kerber

Fachbereich Informatik

Universität des Saarlandes

Inhaltsverzeichnis

Kapitel 1

Unsortierte Prädikatenlogik erster Stufe

Als Ausgangspunkt unserer Betrachtung wollen wir in diesem Kapitel die Standardprädikatenlogik erster Stufe mit Syntax, Semantik und Resolutionskalkül kurz wiederholen.

1.1 Syntax

Die logischen Zeichen der Prädikatenlogik sind:

- eine unendliche Menge von *Variablen* x, y und z (x_i),
- der einstellige *Junktor* $\neg$ („nicht"), die zweistelligen *Junktoren* $\wedge$ („und"), $\vee$ („oder"), $\Rightarrow$ („impliziert") und $\Leftrightarrow$ („äquivalent"),
- die *Quantoren* $\forall$ („für alle") und $\exists$ („es gibt ein") sowie
- die *Hilfssymbole*), (und ,.

Die Signatur $\mathcal{S}$ besteht für jede natürliche Zahl n aus einer (möglicherweise leeren) Menge von n-stelligen *Relationssymbolen*, auch *Prädikatssymbole* genannt, die wir in der Regel mit P, Q und R bezeichnen, sowie einer (möglicherweise leeren) Menge von n-stelligen *Funktionssymbolen*, die wir in der Regel mit f, g und h bezeichnen. Nullstellige Funktionssymbole heißen *Konstanten* und werden mit c, d und e bezeichnet. Bevor wir den Begriff einer Formel definieren können, brauchen wir den Begriff „Term". Terme werden allgemein induktiv definiert.

1.1 Definition (Terme): *Terme* zu einer Signature $\mathcal{S}$ sind genau die Ausdrücke, die sich durch endlichmalige Anwendung der folgenden Regeln erhalten lassen:

T1 Jede Variable ist ein Term.

T2 Jede Konstante ist ein Term.

T3 Sind $t_1, \dots, t_n$ Terme und ist f ein n-stelliges Funktionssymbol, so ist auch $f(t_1, \dots, t_n)$ ein Term. ■

1.2 Definition (Formeln der Prädikatenlogik erster Stufe): *Formeln* der Prädikatenlogik erster Stufe zu einer Signature $\mathcal{S}$ sind genau die Ausdrücke, die sich durch endlichmalige Anwendung der folgenden Regeln erhalten lassen:

F1 Sind $t_1, \dots, t_n$ Terme und ist P ein n-stelliges Relationsssymbol, dann ist der Ausdruck $P(t_1, \dots, t_n)$ eine Formel. So gebildete Formeln heißen *atomar*.

F2 Ist φ eine Formel, so ist $(\neg\varphi)$ eine Formel.

F3 Sind φ und ψ Formeln, so sind auch $(\varphi \wedge \psi)$, $(\varphi \vee \psi)$, $(\varphi \Rightarrow \psi)$ und $(\varphi \Leftrightarrow \psi)$ Formeln.

F4 Ist φ eine Formel und x eine Variable, so sind auch $(\forall x\varphi)$ und $(\exists x\varphi)$ Formeln[1].

F5 In einer Gleichheitslogik ist das zweistellige Prädikatssymbol $\doteq$ vordefiniert, das heißt für zwei Terme t_1 und t_2 ist $t_1 \doteq t_2$ eine Formel. ■

Wir vereinbaren, daß die Bindungsstärke der Junktoren $\neg$, $\wedge$, $\vee$, $\Rightarrow$ und $\Leftrightarrow$ in dieser Reihenfolge abnimmt und können somit viele überflüssige Klammern weglassen. Anstatt $(\mathsf{Q}x\varphi)$ mit $\mathsf{Q} \in \{\forall, \exists\}$ schreiben wir auch $\mathsf{Q}x.\, \varphi$.

1.3 Beispiel: Formeln der Prädikatenlogik sind zum Beispiel „$\forall x.\,$Mensch$(x) \Rightarrow$ sterblich(x)“, „Mensch(Sokrates)“ oder „sterblich(Sokrates)“. ■

1.2 Semantik

Die Bedeutung von Formeln wird dadurch erklärt, daß man Formeln die Wahrheitswerte `falsch` oder `wahr` zuweist.

1.4 Definition (Interpretation):
Eine Interpretation für Formeln der Prädikatenlogik ist ein Paar $\langle\mathcal{U}, \mathcal{I}\rangle$, bestehend aus einer nicht-leeren Menge $\mathcal{U}$, dem *Universum* und einer Abbildung $\mathcal{I}$, die jeder Konstanten aus $\mathcal{S}$ ein Element aus $\mathcal{U}$ zuordnet, jedem n-stelligen Funktionssymbol aus $\mathcal{S}$ eine Funktion von $\mathcal{U} \times \cdots \times \mathcal{U} \to \mathcal{U}$ und jedem n-stelligen Relationssymbol eine n-stellige Relation über $\mathcal{U}$. Eine Belegung ξ in eine Interpretation ist eine Abbildung, die jeder Variable x ein Element aus $\mathcal{U}$ zuordnet. Eine Interpretation und eine Belegung ergeben zusammen die Bedeutung für Terme durch folgende Abbildungen:

- $\mathcal{I}_\xi(x) = \xi(x)$ für jede Variable x
- $\mathcal{I}_\xi(c) = \mathcal{I}(c)$ für jede Konstante c

[1] Im Gegensatz zur Logik höherer Stufe darf in der Logik erster Stufe nur über Objektvariablen quantifiziert werden.

– für n-stelliges Funktionssymbol f und Terme $t_1, \ldots, t_n$ gilt $\mathcal{I}_\xi(f(t_1, \ldots, t_n)) = \mathcal{I}(f)(\mathcal{I}_\xi(t_1), \ldots, \mathcal{I}_\xi(t_n))$

Für Formeln wird definiert[2]:

– $\mathcal{I}_\xi \models P(t_1, \ldots, t_n))$ gdw $\mathcal{I}(P)(\mathcal{I}_\xi(t_1), \ldots, \mathcal{I}_\xi(t_n))$.

– $\mathcal{I}_\xi \models (\neg\varphi)$ gdw nicht $\mathcal{I}_\xi \models \varphi$

– $\mathcal{I}_\xi \models (\varphi \wedge \psi)$ gdw $\mathcal{I}_\xi \models \varphi$ und $\mathcal{I}_\xi \models \psi$

– $\mathcal{I}_\xi \models (\varphi \vee \psi)$ gdw $\mathcal{I}_\xi \models \varphi$ oder $\mathcal{I}_\xi \models \psi$

– $\mathcal{I}_\xi \models (\varphi \Rightarrow \psi)$ gdw wenn $\mathcal{I}_\xi \models \varphi$ dann $\mathcal{I}_\xi \models \psi$

– $\mathcal{I}_\xi \models (\varphi \Leftrightarrow \psi)$ gdw $\mathcal{I}_\xi \models \varphi$ genau dann, wenn $\mathcal{I}_\xi \models \psi$

– $\mathcal{I}_\xi \models (\forall x\varphi)$ gdw für alle $a \in \mathcal{U}$ gilt $\mathcal{I}_{\xi[x \leftarrow a]} \models \varphi$

– $\mathcal{I}_\xi \models (\exists x\varphi)$ gdw es gibt ein $a \in \mathcal{U}$ mit $\mathcal{I}_{\xi[x \leftarrow a]} \models \varphi$

– Hat man eine Logik mit Gleichheit, so wird zusätzlich definiert: $\mathcal{I}_\xi \models (t_1 \doteq t_2)$ gdw $\mathcal{I}_\xi(t_1) = \mathcal{I}_\xi(t_2)$.

Damit ist es nun möglich die folgenden Begriffe zu definierten:

– Eine Interpretation $\mathcal{I}$ *erfüllt* eine Formel φ genau dann, wenn für jede Belegung ξ gilt $\mathcal{I}_\xi \models \varphi$, $\mathcal{I}$ heißt *Modell* von φ. $\mathcal{I}$ heißt Modell einer Formelmenge Γ genau dann, wenn $\mathcal{I}$ Modell jeder Formel aus Γ ist.

– Eine Formel φ *folgt* aus einer Formelmenge Γ (geschrieben als $\Gamma \models \varphi$) genau dann, wenn jedes Modell von Γ auch Modell von φ ist. Eine Formel φ heißt *erfüllbar*, wenn es ein Modell von φ gibt, *unerfüllbar*, wenn es kein Modell gibt, *Tautologie* (geschrieben als $\models \varphi$), wenn jede Interpretation φ erfüllt. ■

1.5 Bemerkung: Für Sätze, das heißt Formeln ohne freie Variable, hängt der Wahrheitswert ihrer Interpretation nicht von der Belegung ab. ■

1.6 Beispiel: Betrachten wir die Formeln aus Beispiel 1.3. Eine intendierte Interpretation ist die folgende: Das Universum $\mathcal{U}$ ist die Menge aller Lebewesen (lebende oder verstorbene). Das Relationssymbol „Mensch" wird auf eine Relation abgebildet, die genau für Menschen wahr ist, entsprechend wird das Relationssymbol „sterblich" so interpretiert, daß es genau auf sterblichen Lebewesen zu wahr ausgewertet wird. Das Konstantensymbol „Sokrates" wird auf den Menschen Sokrates abgebildet. Damit

[2]gdw steht für „genau dann wenn". $\xi[x \leftarrow a]$ ist für alle Variablen außer x identisch mit ξ und $\xi[x \leftarrow a](x) = a$.

ergibt sich für die Formeln „Mensch(Sokrates)" und „sterblich(Sokrates)" als Interpretation jeweils der Wahrheitswert `wahr`. Die Formel „$\forall x$Mensch$(x) \Rightarrow$ sterblich(x)" ergibt auch `wahr`, da für jedes einzelne Element $a \in \mathcal{U}$ gilt, wenn a ein Mensch ist, dann ist a auch sterblich.
Eine andere Interpretation – die natürlich nicht intendiert ist – der Formeln ist: Das Universum $\mathcal{U} = \mathbb{N}$, $\mathcal{I}$(Sokrates) = 8, $\mathcal{I}$(Mensch) ist die Relation, die angibt, ob eine Zahl durch 4 teilbar ist und $\mathcal{I}$(sterblich) gibt an, ob eine Zahl gerade ist. Auch in diesem Fall werden alle drei Formeln zu wahr interpretiert. ∎

1.3 Wichtige Eigenschaften klassischer Logik

In diesem Abschnitt sollen die wichtigsten Eigenschaften der Standardprädikatenlogik kurz vorgestellt werden. Dazu benötigen wir noch den Begriff eines Kalküls.

1.7 Definition (Kalkül): Ein Kalkül ist eine Menge von *Schlußregel* zusammen mit einer Menge von *Axiomen*. Ausgehend von den Axiomen kann man mit den Schlußregeln neue Formeln *ableiten*. Eine Formel φ heißt ableitbar, wenn man sie durch endliche Anwendung von Schlußregeln aus den Axiomen herleiten kann. Man schreibt $\vdash \varphi$. ∎

Als Beispiel für ein Axiom diene das Axiom vom ausgeschlossenen Dritten $\varphi \vee \neg\varphi$, das besagt, daß man für jede Aussage φ sie selbst oder aber ihr Gegenteil annehmen kann. Die vielleicht berühmteste Schlußregel ist der Modus Ponens:

$$\begin{array}{l} \varphi \\ \varphi \Rightarrow \psi \\ \hline \psi \end{array}$$

Nimmt man zu den Axiomen noch eine Formelmenge Γ hinzu und kann dann mit den Schlußregeln aus den Axiomen und Γ eine Formel φ ableiten, so schreibt man $\Gamma \vdash \varphi$.

1.8 Definition (Korrektheit und Vollständigkeit): Ein Kalkül heißt *korrekt*, wenn immer $\Gamma \vdash \varphi$ gilt, dann gilt auch $\Gamma \models \varphi$. Gilt umgekehrt, wann immer $\Gamma \models \varphi$ dann gilt auch $\Gamma \vdash \varphi$, so heißt der Kalkül *vollständig*. ∎

1.9 Bemerkung: Hat man einen korrekten und vollständigen Kalkül, so ist es möglich, die Allgemeingültigkeit von Formeln, die Beziehung $\Gamma \models \varphi$ oder die Unerfüllbarkeit von Formeln zu zeigen, ohne daß es nötig ist, konkrete Modelle zu betrachten. ∎

Für die Prädikatenlogik erster Stufe gilt:

1.10 Theorem:

- *Für die Prädikatenlogik erster Stufe gibt es korrekte und vollständige Kalküle*

[Göd30]. Das heißt die Semantik läßt sich vollkommen auf die Syntax zurückführen.

- *Die Prädikatenlogik erster Stufe ist* nicht *entscheidbar, das heißt es gibt keinen Algorithmus, der für jede Formel entscheidet, ob es sich um eine Tautologie handelt oder nicht [Chu36, Tur37].*

- *Die Tautologien der Prädikatenlogik erster Stufe sind rekursiv aufzählbar (man sagt auch die Prädikatenlogik erster Stufe ist semi-entscheidbar), das heißt es gibt einen Algorithmus, der für jede Tautologie nach endlicher Zeit erkennt, daß es sich um eine Tautologie handelt, für andere Formeln aber nicht zu terminieren braucht.* ∎

Für die Prädikatenlogik erster Stufe sind eine Reihe korrekter und vollständiger Kalküle entwickelt worden, zum Beispiel der Resolutionskalkül, der im automatischen Beweisen eine wichtige Rolle spielt. Zum vertieften Studium gibt es eine Vielzahl von einführenden Logikbüchern wie [Ric78, EFT86, Men87]. Einen Überblick über das ganze Gebiet der Logik wird in den Handbüchern [Bar77, GG89] gegeben. Wer sich für automatisches Beweisen interessiert, sei auf eines der zahlreichen Lehrbücher verwiesen [Bib82, BB87, BM79, CL73, Duf91, Fit90, Lov78]. Zum Abschluß wollen wir kurz den Resolutionskalkül vorstellen.

1.4 Resolution für klassische Prädikatenlogik

Als großen Durchbruch im automatischen Beweisen kann man die Entwicklung des *Resolutionskalkül* 1965 durch J.A. Robinson ansehen [Rob65], der das Problem der Instantiierung von Variablen gelöst hat. Die Resolution löst zwar bei weitem nicht alle Probleme des automatischen Beweisens, aber viele Sätze können mit dieser Methode vollautomatisch bewiesen werden. Der Kalkül wurde im Laufe der Zeit etwas verfeinert und erweitert und auch heute noch basieren die stärksten existierenden Beweisern auf der Resolution (erweitert um Methoden zur Behandlung von Gleichheitsproblemen). Für viele Sortenlogiken ist auch ein Resolutionskalkül entwickelt worden. In diesem Abschnitt wird nun der Standardresolutionskalkül vorgestellt.

Resolution ist ein negatives Testverfahren, das bei seiner Erfindung einen großen Fortschritt darstellte, da die Instantiierungen für allquantifizierte Formeln nicht geraten werden müssen. Der Resolutionskalkül setzt eine Normalform, die *Klauselnormalform*, voraus, von der ausgehend weitere *Klauseln* hergeleitet werden. Kann man die *leere Klausel*, das einzige Axiom für eine falsche Formel, herleiten, so war die ursprüngliche Klauselmenge und damit die Ausgangsformelmenge unerfüllbar.

Die Klauselnormalform ist eine Konjunktion von Disjunktionen, das heißt, eine Formel der Gestalt:

$$(L_{11} \vee \ldots \vee L_{1n_1}) \wedge \ldots \wedge (L_{m1} \vee \ldots \vee L_{mn_m})$$

wobei die einzelnen L_{ij}, die sogenannten *Literale*, atomare Formeln oder deren Negation sind. $(L_{i1} \vee \ldots \vee L_{in_i})$ heißt eine *Klausel*. Üblicherweise notiert man die Klauselnormalform als eine Liste von Klauseln, die implizit mit "$\wedge$" verknüpft sind. Die einzelnen Klauseln werden dabei durchnummeriert. In der Klauselnormalform sind alle Variablen implizit allquantifiziert. Existenzquantoren werden durch die *Skolemisierung* eliminiert, zum Beispiel erhält man aus $\forall x\ \exists y\ P(x, y)$ die Formel $\forall x\ P(x, f(x))$, wobei f ein neues Funktionssymbol ist. Zu jeder Formelmenge kann man eine Klauselnormalform berechnen, die genau dann unerfüllbar ist, wenn es die Ursprungsformelmenge war.

Die Klauselnormalform wird dadurch erzeugt, daß man die Ausgangsformelmenge mit Hilfe der folgenden Äquivalenzen umformt, wobei noch eine Variablenumbenennungsregel benutzt wird, die es einem erlaubt die Voraussetzungen für Regeln, mit denen man einen Quantor über Disjunktionen und Konjunktionen ziehen kann, zu schaffen. Dadurch läßt sich eine Formel in eine äquivalente Pränexform überführen. [EO87, S.27f]:

$$
\begin{array}{lll}
\varphi \Leftrightarrow \psi & \rightsquigarrow & (\varphi \Rightarrow \psi) \wedge (\psi \Rightarrow \varphi) \\
\varphi \Rightarrow \psi & \rightsquigarrow & \neg\varphi \vee \psi \\
\neg(\varphi \wedge \psi) & \rightsquigarrow & \neg\varphi \vee \neg\psi \\
\neg(\varphi \vee \psi) & \rightsquigarrow & \neg\varphi \wedge \neg\psi \\
\neg\forall x\varphi & \rightsquigarrow & \exists x\neg\varphi \\
\neg\exists x\varphi & \rightsquigarrow & \forall x\neg\varphi \\
(\forall x\varphi) \wedge \psi & \rightsquigarrow & \forall x(\varphi \wedge \psi) \qquad \text{sofern } x \text{ nicht frei in } \psi \\
(\forall x\varphi) \vee \psi & \rightsquigarrow & \forall x(\varphi \vee \psi) \qquad \text{sofern } x \text{ nicht frei in } \psi \\
(\exists x\varphi) \wedge \psi & \rightsquigarrow & \exists x(\varphi \wedge \psi) \qquad \text{sofern } x \text{ nicht frei in } \psi \\
(\exists x\varphi) \vee \psi & \rightsquigarrow & \exists x(\varphi \vee \psi) \qquad \text{sofern } x \text{ nicht frei in } \psi \\
(\forall x\varphi) \wedge (\forall x\psi) & \rightsquigarrow & \forall x(\varphi \wedge \psi) \\
(\exists x\varphi) \vee (\exists x\psi) & \rightsquigarrow & \exists x(\varphi \vee \psi) \\
\varphi \vee (\psi \wedge \chi) & \rightsquigarrow & (\varphi \vee \psi) \wedge (\varphi \vee \chi) \\
\varphi \wedge (\psi \vee \chi) & \rightsquigarrow & (\varphi \wedge \psi) \vee (\varphi \wedge \chi)
\end{array}
$$

Eine Formel $\forall x_1 \cdots \forall x_m \exists y \varphi$ wird durch *Skolemisierung*[3] in eine Formel ohne Existenzquantor dadurch übergeführt, daß alle freien Vorkommen von y in φ durch den Term $f_y(x_1, \ldots, x_m)$ ersetzt werden, wobei die *Skolemfunktion* f_y durch ein neues Funktionssymbol dargestellt wird. Nachdem man alle Existenzquantoren durch Skolemisierung beseitigt hat, hat man (wenn man nicht von beliebigen Formeln, sondern von Sätzen ausgeht) nur noch allquantifizierte Variablen in den Formeln, also kann man die Quantoren weglassen und sich merken, daß alle Variablen allquantifiziert sind.

Zwar sind die Ausgangsformelmenge Γ und die so erhaltene Klauselnormalform $\mathcal{C}_\Gamma$ nicht mehr logisch äquivalent, aber es gilt der folgende Satz.

[3] Die Skolemisierung wird nach Thoralf Skolem benannt [Sko19].

1.11 THEOREM: *Eine Formelmenge Γ ist genau dann erfüllbar (unerfüllbar), wenn die zugehörige Klauselnormalform $\mathcal{C}_\Gamma$ erfüllbar (unerfüllbar) ist.* ■

1.12 BEISPIEL: Sei zum die folgende unerfüllbare Formelmenge Γ gegeben:

(1) $\mathsf{mensch}(\mathsf{max})$
(2) $\forall x\,(\mathsf{mensch}(x) \Rightarrow \mathsf{mensch}(\mathsf{vater}(x)))$
(3) $\neg\mathsf{mensch}(\mathsf{vater}(\mathsf{max}))$.

Die zugehörige Klauselnormalform $\mathcal{C}_\Gamma$ ist dann:

(1) $\mathsf{mensch}(\mathsf{max})$
(2) $\neg\mathsf{mensch}(x) \vee \mathsf{mensch}(\mathsf{vater}(x))$
(3) $\neg\mathsf{mensch}(\mathsf{vater}(\mathsf{max}))$ ■

Auf der Klauselnormalform kann man nun den Resolutionskalkül anwenden. Mit den beiden Kalkülregeln **Res** und **Fak** werden ausgehend von einer Klauselmenge $\mathcal{C}$ neue Klauseln erzeugt und zu der aktuellen Klauselmenge hinzugefügt bis das einzige Axiom (für Falschheit), die leere Klausel $\Box$ abgeleitet ist. Hat man dies abgeleitet, so hat man gezeigt, daß die Ausgangsklauselmenge unerfüllbar ist.
Die allgemeine Resolutionsregel hat die Gestalt:

Res
$$\frac{\begin{array}{ll} L \vee K_1 \vee \ldots \vee K_n & \\ \neg L' \vee M_1 \vee \ldots \vee M_m & \sigma(L) = \sigma(L') \end{array}}{\sigma(K_1) \vee \ldots \vee \sigma(K_n) \vee \sigma(M_1) \vee \ldots \vee \sigma(M_m)}$$

Dabei ist σ die allgemeinste Einsetzung die L und L' gleich macht, der *allgemeinste Unifikator*. Im obigen Beispiel ist σ gleich $(x \leftarrow \mathsf{max})$. Die Faktorisierungsregel ist:

Fak
$$\frac{L \vee L' \vee K_1 \vee \ldots \vee K_n \qquad \sigma(L) = \sigma(L')}{\sigma(L) \vee \sigma(K_1) \vee \ldots \vee \sigma(K_n)}$$

Die Resolutionsregel angewendet auf (1)1 und (2)1 – die zweite Zahl soll dabei das entsprechende Literal angeben – liefert im obigen Beispiel die *Resolvente*

(4) $\mathsf{mensch}(\mathsf{vater}(\mathsf{max}))$

Diese wiederum resolviert mit Klausel (3)1, man erhält

(5) $\Box$

Also war die Ausgangsklauselmenge $\mathcal{C}_\Gamma$ unerfüllbar und damit auch die Formelmenge Γ.
Es gilt der folgende Satz:

1.13 THEOREM: *Der Resolutionskalkül bestehend aus der Resolutionsregel* **Res** *und der Faktorregel* **Fak** *zusammen mit der leeren Klausel $\Box$ als Axiom für eine widersprüchliche Formel ergibt einen widerlegungskorrekten und widerlegungsvollständigen Kalkül.* ■

Das bedeutet, man kann die leere Klausel nur ableiten, wenn die Ausgangsklauselmenge unerfüllbar ist (Korrektheit), und umgekehrt, wenn die Ausgangsklauselmenge unerfüllbar ist, so läßt sich auch die leere Klausel ableiten (Vollständigkeit).

Eine wichtige Eigenschaft der Resolution ist, daß weder ganze Formeln geraten werden müssen, die in die Axiome mit der Substitutionsregel eingesetzt werden müssen, noch ist es wegen der Berechnung der allgemeinsten Unifikatoren notwendig, Einsetzungen für universell quantifizierte Variablen zu raten.

1.14 BEMERKUNG: Aus den beiden Sätzen zusammen ergibt sich also das Resolutionsverfahren. Will man zeigen, daß aus einer Formelmenge Γ eine Formel φ folgt (also $\Gamma \models \varphi$), so bedeutet das (in klassischer Logik) ja gerade, daß $\Gamma \cup \{\neg\varphi\}$ unerfüllbar ist. Diese Formelmenge wird in die zugehörige Klauselnormalform übersetzt, die genau dann unerfüllbar ist, wenn es die Ausgangsformelmenge war. Nun werden solange die Resolutions- und die Faktorregel angewandt bis die leere Klausel abgeleitet ist. In diesem Fall ist ein Widerspruch hergeleitet, somit war die Ausgangsklauselmenge unerfüllbar, also folgt φ tatsächlich aus Γ. Ist keine der Regeln mehr anwendbar und die leere Klausel ist nicht abgeleitet (was in praktischen Fällen selten vorkommt), so war die Ausgangsklauselmenge erfüllbar, φ folgt also nicht aus Γ. Sind aber noch Regeln anwendbar und die leere Klausel ist noch nicht abgeleitet, so weiß man nicht, ob sie noch abgeleitet werden kann oder nicht (was der Unentscheidbarkeit der Prädikatenlogik entspricht). ■

Kapitel 2

Grundlagen sortierter Prädikatenlogik

Während in klassischer Prädikatenlogik sowohl in Syntax als auch in der Semantik alle Individuen unstrukturiert gegenüberstehen, wird der Diskursbereich in den Sortenlogiken strukturiert. In diesem Kapitel wollen wir uns dazu einige Beispiele ansehen und die grundlegenden Begriffe von Sortenlogiken einführen.

2.1 Historie, Motivation und Beispiele

In der mathematischen Praxis betrachtet man anders als in der formalen klassischen Logik erster Stufe keine unstrukturierten Objekte, sondern legt ganz im Gegenteil sehr viel Wert auf eine klare Strukturierung, was sich zum Beispiel in einer sorgfältig gewählten Nomenklatur niederschlägt. Dies trifft zwar teilweise auf die Prädikatenlogik erster Stufe auch zu, es werden ja Variable (mit x, y oder z bezeichnet), Konstante (c, d), Funktionen (f, g oder h) und Prädikate (P, Q oder R) unterschieden, aber der Individuenbereich (Variable und Konstante) ist nicht weiter untergliedert, wie zum Beispiel in die Zahlbereiche natürliche Zahlen, ganze Zahlen, rationale Zahlen, reelle Zahlen und komplexe Zahlen. Für einige dieser Bereiche werden allerdings in der Mathematik die Platzhalter nicht nur für eine Problemstellung festgelegt, sondern stehen übergreifend fest, wie n oder m für natürliche Zahlen.
Entsprechend hat man in der Logik dann auch schon früh Systeme untersucht, die auch eine Strukturierung des Individuenbereiches erlauben, dies geschieht in den sogenannten Sortenlogiken [Sch38, Obe62]. In diesen Logiken ist eine Menge von Sorten **SORT** gegeben, und man kann den Termen der Logik Sorten aus dieser Menge zuordnen. Bei Variablen geschieht dies dadurch, daß man die Sorte in einem Index angibt, für andere Terme werden die Sorten durch sogenannte Termdeklarationen festgelegt. Sehen wir uns nun einige Beispiele an:

- $\forall x_{\mathbb{Z}}.\ x^2 \geq 0$
- $\forall x_{\mathbb{N}}.\ \forall y_{\mathbb{N}}.\ \mathsf{gerade}(x) \wedge \mathsf{gerade}(y) \Rightarrow \mathsf{gerade}(x+y)$

- $\mathsf{gerade}(0)$
- $\exists x_{\mathsf{lebewesen}}.\ \mathsf{sterblich}(x)$
- $\mathsf{sterblich}(\mathsf{sokrates})$
- $\forall x_{\mathbb{Z}}.\ x^2 \geq 0 \Rightarrow 1^2 > 0$

Was sollte eine Sortenlogik für Konzepte enthalten, die es erlauben solche Ausdrücke zu benutzen. Zuerst braucht man natürlich die Menge **SORT** der Sortensymbole, in diesem Fall also **SORT** $= \{\mathbb{Z}, \mathbb{N}, \mathsf{lebewesen}, \mathsf{mensch}\}$. Darüber hinaus soll in dieser Menge eine Untersortenbeziehung zwischen den Sortensymbolen möglich sein, wie $\mathbb{N} \sqsubseteq \mathbb{Z}$ und $\mathsf{mensch} \sqsubseteq \mathsf{lebewesen}$. Weiterhin wollen wir ausdrücken können, daß bestimmte Konstante zu einer Sorte gehören, wie $0 \lessdot \mathbb{Z}$, $1 \lessdot \mathbb{N}$ und $\mathsf{sokrates} \lessdot \mathsf{mensch}$. Für auftretende Funktionssymbole wie $+$ wollen wir sagen können, daß die Summe zweier natürlichen Zahlen wieder eine natürliche Zahl ist, also $+ : \mathbb{N} \times \mathbb{N} \to \mathbb{N}$. Dies geschieht mit Hilfe von Termdeklaration, die allerdings auch noch andere Fälle umfassen können, wie zum Beispiel $x_{\mathbb{Z}} * x_{\mathbb{Z}} \lessdot \mathbb{N}$. Zuguterletzt wollen wir auch noch sagen können, daß gewisse Prädikate nur auf bestimmten Sorten Sinn machen, zum Beispiel $\mathsf{sterblich}$ macht nur für Lebewesen Sinn, nicht aber für Zahlen, und gerade macht nur für ganze Zahlen Sinn, nicht aber für Lebewesen.
Was ist nun durch solche eine Erweiterung gewonnen? Kann man mehr ausdrücken als vorher? Die Antwort darauf ist: Nein. Denn man kann die obigen Formeln alle auch in unsortierter Logik ausdrücken, nämlich durch:

- $\forall x.\ \mathbb{Z}(x) \Rightarrow x^2 \geq 0$
- $\forall x.\ \mathbb{N}(x) \Rightarrow (\forall y.\ \mathbb{N}(y) \Rightarrow (\mathsf{gerade}(x) \wedge \mathsf{gerade}(y) \Rightarrow \mathsf{gerade}(x+y)))$
- $\mathsf{gerade}(0)$
- $\exists x.\ \mathsf{lebewesen}(x) \wedge \mathsf{sterblich}(x)$
- $\mathsf{sterblich}(\mathsf{sokrates})$
- $\forall x.\ \mathbb{Z}(x) \Rightarrow (x^2 \geq 0 \Rightarrow 1^2 > 0)$

Zusätzlich muß man noch die Beziehungen, die in den Untersorten und Termdeklarationen aufgeführt sind, in unsortierte Prädikatenlogik übersetzen. Dies ergibt im obigen Fall Beziehungen wie:

- $\forall x.\ \mathbb{N}(x) \Rightarrow \mathbb{Z}(x)$
- $\forall x.\ \mathbb{N}(x) \Rightarrow \forall y.\ \mathbb{N}(y) \Rightarrow \mathbb{N}(x+y)$
- $\forall x.\ \mathsf{mensch}(x) \Rightarrow \mathsf{lebewesen}(x)$
- $\forall x.\ \mathbb{Z}(x) \Rightarrow \mathbb{N}(x * x)$
- $\mathbb{Z}(0)$, $\mathbb{N}(1)$ und $\mathsf{mensch}(\mathsf{sokrates})$

Was ist dann also der Vorteil der sortierten Formulierung?

Im wesentlichen kann man drei Vorteile unterscheiden:
Erstens kann man in Sortenlogik gewisse unsinnige Formeln auf der syntaktischen Ebene abfangen, die in unsortierter Logik hingeschrieben werden können. Zum Beispiel kann ein Ausdruck wie $\mathsf{mensch}(1) \Rightarrow \mathsf{sterblich}(1)$ in sortierter Logik zurückgewiesen werden, ebenso Ausdrücke in denen Terme wie $\mathsf{sokrates} + \mathsf{platon}$ vorkommen. Diese können auch in unsortierter Logik nicht zu sinnvollen Deduktionsschritten verwendet werden, zum Beispiel kann man mit der Formel $\mathsf{mensch}(1) \Rightarrow \mathsf{sterblich}(1)$ nie den Modus Ponens anwenden, da man nie $\mathsf{mensch}(1)$ in einer sinnvollen Axiomatisierung ableiten können wird. Das heißt solche unsinnigen Formeln vergrößern zwar nicht die Menge der ableitbaren Formeln, aber sie können sehr wohl den Suchraum vergrößern. (Zudem haben Sorten in der Logik eine ähnlich heilsame Wirkung gegen Tippfehler wie Typen in Programmiersprachen wie PASCAL.)
Zweitens ist die Darstellung in sortierter Logik einfacher als in unsortierter Logik. Man sieht dies an den obigen Beispielen: während die sortierte Formel $\forall x_{\mathbb{Z}}.\, x^2 \geq 0$ ein Atom enthält, enthält die entsprechende unsortierte Formel $\forall x.\, \mathbb{Z}(x) \Rightarrow x^2 \geq 0$ zwei Atome. Dies vergrößert bei einer Beweissuche den Suchraum entsprechend.
Der vielleicht entscheidende Vorteil einer sortierten Formulierung gegenüber einer unsortierten besteht drittens darin, daß die möglichen Einsetzungen für Variablen in Quantifizierungen wie in $\forall x_{\mathbb{Z}}.\, x^2 \geq 0$ gegenüber unsortierten Formulierungen eingeschränkt ist. In dieser Formel darf man für x nur Terme der Sorte $\mathbb{Z}$ einsetzten, also zum Beispiel 0 oder 1, nicht aber $\mathsf{sokrates}$ oder platon. In der entsprechenden unsortierten Formel $\forall x.\, \mathbb{Z}(x) \Rightarrow x^2 \geq 0$ kann man hingegen für x geliebe Terme einsetzen, also auch $\mathsf{sokrates}$, womit man erhält $\mathbb{Z}(\mathsf{sokrates}) \Rightarrow \mathsf{sokrates}^2 \geq 0$. Diese Ableitung ist aber unsinnig, da man sie nicht weiter verwenden kann: Die Formel ist deshalb wahr, weil die Prämisse $\mathbb{Z}(\mathsf{sokrates})$ falsch ist, also nie hergeleitet werden kann.

2.2 Syntax sortierter Logik erster Stufe

Wir wollen uns nun anschauen, wie man formal die Syntax einer sortierten Logik erster Stufe angegeben kann. Dafür folgen wir im wesentlichen der Darstellung in [SS89].
Wir wollen für das folgende einige Notationen festlegen:

2.1 NOTATION: Wir bezeichnen die unsortierte Signatur im folgenden mit $\bar{\Sigma}$, die Menge der Funktionssymbole mit $\mathbf{F}_{\bar{\Sigma}}$ und die der Prädikatssymbole mit $\mathbf{P}_{\bar{\Sigma}}$. Die Menge der Variablen wir mit $\mathbf{V}$ und die der Terme mit $\mathbf{T}$ (abkürzend für $\mathbf{T}(\bar{\Sigma})$) bezeichnet. ∎

2.2 DEFINITION: Sei **SORT** eine nichtleere Menge von Sortensymbolen.

- Eine *Termdeklaration* ist ein Paar (t, S) bestehend aus einem Term und einem Sortensymbol, geschrieben als $t \lessdot S$, mit $t \in \mathbf{T} \setminus \mathbf{V}$. Wenn t die Gestalt $f(x_1, \ldots, x_n)$ mit paarweise verschiedenen Variablen hat, dann sagen wir $t \lessdot S$

ist eine *Funktionsdeklaration*, im Falle von $c \lessdot S$ reden wir von einer *Konstantendeklaration* und andernfalls von einer *eigentlichen Termdeklaration*.

– Eine *Untersortendeklaration* hat die Form $R \sqsubseteq S$ mit $R, S \in \mathbf{SORT}$.

– Eine *Prädikatsdeklaration* hat die Form $P \lessdot S_1 \times \cdots \times S_n$ mit $S_i \in \mathbf{SORT}$. ■

Die logischen Zeichen der sortierten Prädikatenlogik sind genau die der unsortierten Logik (vergleiche Abschnitt 1.1). Nun sind wir in der Lage eine sortierte Signatur zu definieren:

2.3 DEFINITION (SORTIERTE SIGNATUR): Eine *sortierte Signatur* Σ besteht aus:

1. einer unsortierten Signatur $\bar{\Sigma}$,
2. einer Menge **SORT** von Sorten,
3. einer Funktion $\mathfrak{s} : \mathbf{V} \to \mathbf{SORT}$, so daß für jede Sorte $S \in \mathbf{SORT}$ eine abzählbar unendliche Menge von Variablen $x \in \mathbf{V}$ gibt mit $\mathfrak{s}(x) = S$ sowie
4. je einer Menge von Termdeklarationen (mit $\mathbf{TD}_\Sigma$ bezeichnet), Untersortendeklarationen ($\mathbf{SD}_\Sigma$) und Prädikatsdeklarationen ($\mathbf{PD}_\Sigma$).

In einer sortierten Logik mit Gleichheit ist das Gleichheitsprädikat $=_\Sigma$ in $\mathbf{P}_\Sigma$ und für alle Sorten R und S ist $=_\Sigma \lessdot R \times S$ in $\mathbf{PD}_\Sigma$. ■

Durch die Funktion $\mathfrak{s}$ werden die Menge der Variablen in verschiedene Mengen $\mathbf{V}_S$ partitioniert. Wir schreiben eine Variable x mit $\mathfrak{s}(x) = S$ auch kurz als x_S. In gleicher Weise kürzen wir für Konstanten auch $c \lessdot S$ mit c_S ab. Im folgenden bezeichnen wir mit $\sqsubseteq_\Sigma$ (Wenn keine Verwechslungen zu befürchten sind, lassen wir in diesem wie in anderen Fällen den Subskript Σ einfach weg.) den reflexiven und transitiven Abschluß der in **SD** angegebenen Relation.
Die hier erlaubten Sortenstrukturen sind relativ allgemein, was für die Kalkülisierung natürlich Schwierigkeiten bereiten kann. Deshab werden die folgenden Begriffe eingeführt, die einfachere Sortenlogiken als Spezialfälle des allgemeinen Falles klassifizieren.

2.4 DEFINITION: Eine Signatur heißt

– *einsortig*, wenn **SORT** einelementig ist.

– *flachsortiert*, wenn es mehr als ein Sortensymbol gibt, aber keine Untersortendeklarationen.

– *ordnungssortiert*, wenn es mehr als ein Sortensymbol gibt und Untersortendeklarationen vorhanden sind.

– *linear*, wenn alle Terme in **TD** linear sind, das heißt jede Variable höchstens einmal vorkommt.

– *elementar*, wenn alle Termdeklarationen Funktionsdeklarationen sind.

– *einfach*, wenn sie elementar ist und für jedes Funktionssymbol genau eine Funktionsdeklaration vorkommt.

– *baumartig*, wenn durch die Untersortenbeziehung ein Baum definiert wird. ■

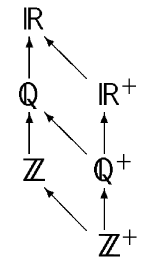
Schauen wir uns nun für all diese Fälle Beispiele an:

2.5 Beispiel:

– *einsortig*: $\mathbf{SORT} = \{\mathbb{Z}\}$. In diesem Fall bildet $\mathfrak{s}$ alle Variablen auf $\mathbb{Z}$ ab und alle Terme können nur die Sorte $\mathbb{Z}$ erhalten.

– *flachsortiert*: $\mathbf{SORT} = \{\mathbb{Z}, \mathsf{mensch}\}$. Es gibt Variablen, die ganze Zahlen und welche die Menschen denotieren, Termdeklarationen können beide Sorten umfassen, wie in $p(x_{\mathbb{Z}}) < \mathbb{Z}$ und $p(x_{\mathsf{mensch}}) < \mathsf{mensch}$. Allerdings ist keine Untersortendeklaration wie $\mathbb{Z} \sqsubset \mathsf{mensch}$ möglich.

– *ordnungssortiert*: $\mathbf{SORT} = \left\{\mathbb{Z}, \mathbb{Q}, \mathbb{R}, \mathbb{Z}^+, \mathbb{Q}^+, \mathbb{R}^+\right\}$ mit den Untersortenbeziehungen $\mathbb{Z} \sqsubseteq \mathbb{Q} \sqsubseteq \mathbb{R}$, $\mathbb{Z}^+ \sqsubseteq \mathbb{Z}$, $\mathbb{Z}^+ \sqsubseteq \mathbb{Q}^+$; $\mathbb{Q}^+ \sqsubseteq \mathbb{Q}$, $\mathbb{Q}^+ \sqsubseteq \mathbb{R}^+$; $\mathbb{R}^+ \sqsubseteq \mathbb{R}$, das heißt der Struktur:

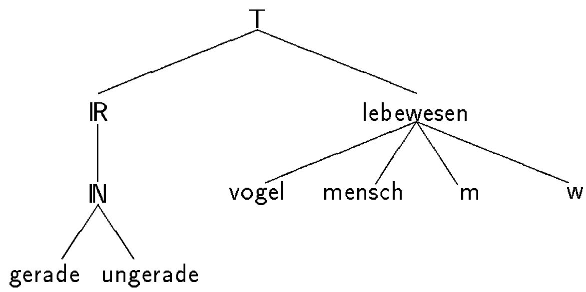


– *linear*: Die Termdeklarationen haben die Gestalt $+(*(x_{\mathbb{N}}, y_{\mathbb{Z}}), z_{\mathbb{N}}) < \mathbb{Z}$, das heißt Termdeklarationen der Form $*(x_{\mathbb{Z}}, x_{\mathbb{Z}}) < \mathbb{N}$ sind ausgeschlossen.

– *elementar*: Die Termdeklarationen sind von der Form $+(x_{\mathbb{N}}, y_{\mathbb{N}}) < \mathbb{N}$. Zu dieser Termdeklaration ist gleichzeitig die Termdeklaration $+(u_{\mathbb{Z}}, v_{\mathbb{Z}}) < \mathbb{Z}$ möglich. Generell nicht möglich sind Termdeklarationen der Form des Beispiels für lineare Termdeklarationen.

– *einfach*: Die obige Termdeklaration $+(x_{\mathbb{N}}, y_{\mathbb{N}}) < \mathbb{N}$ ist möglich, aber dann darf die zweite, das heißt $+(u_{\mathbb{Z}}, v_{\mathbb{Z}})$, nicht mehr auftreten.

– *baumartig*: $\mathbf{SORT} = \{\mathsf{T}, \mathbb{R}, \mathbb{N}, \mathsf{gerade}, \mathsf{ungerade}, \mathsf{lebewesen}, \mathsf{vogel}, \mathsf{mensch}, \mathsf{m}, \mathsf{w}\}$



■

2.6 DEFINITION (SUBSTITUTION): Man definiert eine *Substitution* σ als einen Homomorphismus auf der Menge der Terme – also eine Abbildung mit $\sigma(f(t_1,\ldots,t_n)) = f(\sigma(t_1),\ldots,\sigma(t_n))$ –, so daß die Menge $\{x \in \mathbf{V} \mid \sigma(x) \neq x\}$ endlich ist. Wir bezeichnen mit $t[x \leftarrow r]$ den Term, der aus t entsteht, wenn man die Substitution $[x \leftarrow r]$, die alle Vorkommen von x durch r ersetzt, auf t anwendet, ■

Nun kommen wir zur Definition von Termen in einer sortierten Logik. Sei also eine Signatur Σ und eine Sorte S in dieser Signatur gegeben.

2.7 DEFINITION (SORTIERTE TERME): Die Menge der *wohl-sortierten Terme* $\mathbf{T}_{\Sigma,S}$ der Sorte S bezüglich einer Signatur Σ werden rekursiv durch die folgenden drei Regeln erzeugt:

- $x \in \mathbf{T}_{\Sigma,S}$ für alle Variablen x mit $\mathfrak{s}(x) \sqsubseteq S$
- $t \in \mathbf{T}_{\Sigma,S}$ wenn $(t{<}R) \in \Sigma$ und $R \sqsubseteq S$
- $t[x \leftarrow r] \in \mathbf{T}_{\Sigma,S}$ wenn $t \in \mathbf{T}_{\Sigma,S}$, $r \in \mathbf{T}_{\Sigma,R}$ und $x \in \mathbf{V}_\Sigma$ mit $R \sqsubseteq \mathfrak{s}(x)$. ■

2.8 BEISPIEL: Sei eine Signatur mit $\mathbf{SORT} = \{\mathsf{gerade}, \mathrm{I\!N}\}$, $\mathbf{SD} = \{\mathsf{gerade} \sqsubseteq \mathrm{I\!N}\}$
$\mathbf{TD} = \{0{<}\mathsf{gerade}, s{<}\mathrm{I\!N} \to \mathrm{I\!N}, s(s(x_{\mathsf{gerade}})){<}\mathsf{gerade},$
$+{<}\mathrm{I\!N} \times \mathrm{I\!N} \to \mathrm{I\!N}, +{<}\mathsf{gerade} \times \mathsf{gerade} \to \mathsf{gerade}, y_{\mathrm{I\!N}} + y_{\mathrm{I\!N}}{<}\mathsf{gerade}\}$
Wie sehen nun die Sorten einiger Terme aus: Der Term $y_{\mathrm{I\!N}}{+}y_{\mathrm{I\!N}}$ ist in $\mathbf{T}_{\Sigma,\mathsf{gerade}}$ (gemäß der zweiten Regel), folglich ist auch der Term $s(0)+s(0)$ in $\mathbf{T}_{\Sigma,\mathsf{gerade}}$, $s(0)+0$ jedoch nicht, dieser Term ist wie die beiden anderen in $\mathbf{T}_{\Sigma,\mathrm{I\!N}}$ ■

2.9 BEMERKUNG: Eine äquivalente Möglichkeit, Terme zu definieren, besteht darin, daß man auf Untersortenbeziehungen ganz verzichtet und statt einer Untersortenbeziehung $S \sqsubseteq T$ die Termdeklaration $x_S{<}T$ zuläßt. In diesem Fall muß man natürlich die drei Regeln anpassen. ■

Es gilt das folgende Lemma:

2.10 LEMMA:

- *Für alle Sorten* $R, S \in \mathbf{SORT}_\Sigma$ *mit* $R \sqsubseteq S$ *gilt* $\mathbf{T}_{\Sigma,R} \subseteq \mathbf{T}_{\Sigma,S}$.
- *Für alle Variablen gilt:* $x \in \mathbf{T}_{\Sigma,S} \Leftrightarrow \mathfrak{s}(x) \sqsubseteq S$. ■

Die Menge aller wohlsortierten Terme $\mathbf{T}_\Sigma$ wird definiert als die Vereinigung aller wohlsortierten Terme aller Sorten, das heißt $\mathbf{T}_\Sigma = \bigcup_{S \in \mathbf{SORT}} \mathbf{T}_{\Sigma,S}$. Wir bezeichnen für einen Term t mit $\mathbf{SORT}_\Sigma(t)$ die Menge aller Sorten, die t hat, also $\mathbf{SORT}_\Sigma(t) = \{S \in \mathbf{SORT} \mid t \in \mathbf{T}_{\Sigma,S}\}$.
Wir sagen eine Signatur Σ und entsprechend die Menge der Terme $\mathbf{T}_\Sigma$ ist *untertermabgeschlossen*, wenn jeder Unterterm eines wohlsortierten Terms selbst wieder wohlsortiert ist.

2.11 Beispiel: Sei Σ eine Signatur mit den Funktionssymbolen f und a, $\mathbf{SORT} = \{S\}$ und der Termdeklaration $f(a){<}S$. Die Menge der wohlsortierten Terme besteht dann aus $\mathbf{T}_\Sigma = \{f(a)\} \cup \mathbf{V}$. Insbesondere ist $a \notin \mathbf{T}_\Sigma$, also ist Σ nicht untertermabgeschlossen. ■

2.12 Definition (Grundterm): Ein *Grundterm* ist ein Term, der keine Variablen enthält. Die Menge aller Grundterme einer Sorte S wird mit $\mathbf{T}_{\Sigma,S,\mathrm{gr}}$ bezeichnet, die Menge aller wohlsortierten Grundterme mit $\mathbf{T}_{\Sigma,\mathrm{gr}}$. ■

Für das folgende machen wir die beiden Annahmen:

2.13 Annahme:

1. Alle Signaturen sind untertermabgeschlossen.

2. Für jede Sorte S aus $\mathbf{SORT}_\Sigma$ gibt es einen Grundterm t_{gr} aus $\mathbf{T}_{\Sigma,\mathrm{gr}}$ mit $S \in \mathbf{SORT}_\Sigma(t)$ ■

Die erste Annahme besagt also, daß es keine unsinnigen Terme als Unterterme in wohlgeformten Termen geben kann. Die zweite Annahme besagt mehr oder minder, daß Sorten nicht-leer sind, also daß es zu jeder Sorte tatsächlich einen Bewohner gibt, den man auch konkret (in Form eines Grundterms) angeben kann. Es ist allerdings auch möglich Sortensysteme mit leeren Sorten zuzulassen. Dies bereitet allerdings in der Deduktion, daß man gewisse zusätzliche Behauptungen zeigen muß. Konkret, wenn man in einem Resolutionsschritt aus den Klauseln $P(x_S)$ und $\neg P(x_S)$ die leere Klausel herleiten kann, so ist noch kein Widerspruch hergeleitet, sondern man hat noch die Randbedingung zu zeigen, daß die Sorte S nicht-leer ist.

2.14 Definition (Formeln): Eine *wohlsortierte Formel* wird wieder rekursiv definiert:

F1 Für jedes Prädikatssymbol mit einer Prädikatsdeklaration $P{<}S_1 \times \cdots \times S_n$ und $S_i \in \mathbf{SORT}$, so daß für alle Terme $t_i \in \mathbf{T}_{\mathbf{SORT},S_i}$, ist $P(t_1, \ldots, t_n)$ eine wohlsortierte (atomare) Formel.

F2 Ist φ eine Formel, so ist $(\neg\varphi)$ eine Formel.

F3 Sind φ und ψ Formeln, so sind auch $(\varphi \wedge \psi)$, $(\varphi \vee \psi)$, $(\varphi \Rightarrow \psi)$ und $(\varphi \Leftrightarrow \psi)$ Formeln.

F4 Ist φ eine Formel und x eine Variable mit $\mathfrak{s}(x) = S$, so sind auch $(\forall x_S.\, \varphi)$ und $(\exists x_S.\, \varphi)$ Formeln.

F5 In einer Gleichheitslogik ist das zweistellige Prädikatssymbol $\doteq$ vordefiniert, das heißt für zwei Terme t_1 und t_2 ist $t_1 \doteq t_2$ eine Formel. ■

2.3 Semantik sortierter Logik erster Stufe

Wie für die unsortierte Logik geben wir nun eine Semantik für die sortierte Logik an. Im Gegensatz zur unsortierten Logik können wir uns die Semantik so vorstellen, daß nicht mehr ein einziges Universum existiert, sondern für jede Sorte $S \in \mathbf{SORT}$ ein Universum $\mathcal{U}_S$. Allerdings müssen wir nicht nur für Konstante angeben, wie sie interpretiert werden sollen, sondern auch für Funktionen, dort können aber kompliziertere Situationen auftreten, im Beispiel 2.8 etwa haben wir die beiden Funktionsdeklarationen $+ < \mathbb{N} \times \mathbb{N} \to \mathbb{N}$ und $+ < \mathsf{gerade} \times \mathsf{gerade} \to \mathsf{gerade}$. Deshalb führen wir die Semantik mithilfe von partiellen Funktionen ein. Eine partielle Funktion ist eine Funktion, die zwar eindeutig sein muß, durchaus aber Definitionslücken haben kann, das heißt eine rechteindeutige Relation (sie muß im Gegensatz zu einer totalen Funktion nicht linkstotal sein).

2.15 Definition (Interpretation): Eine Σ-Interpretation für Formeln der sortierten Prädikatenlogik mit der Signatur Σ ist ein Paar $\langle \mathcal{U}, \mathcal{I} \rangle$, bestehend aus einer nicht-leeren Menge $\mathcal{U}$, dem *Universum* und einer Abbildung $\mathcal{I}$. Für jede Sorte $S \in \mathbf{SORT}$ haben wir eine nicht-leere Menge $\mathcal{U}_S \subseteq \mathcal{U}$ mit $\mathcal{U} = \bigcup_{S \in \mathbf{SORT}} \mathcal{U}_S$. Die Abbildung $\mathcal{I}$ ordnet jeder Konstanten ein Element aus $\mathcal{U}$ zu, jedem n-stelligen Funktionssymbol aus Σ eine *partielle* Funktion von $\mathcal{U} \times \cdots \times \mathcal{U} \to \mathcal{U}$ und jedem n-stelligen Relationssymbol eine n-stellige Relation über $\mathcal{U}$. Eine Belegung ξ in eine Interpretation ist eine Abbildung, die jeder Variable x der Sorte S ein Element aus $\mathcal{U}_S$ zuordnet. Eine Interpretation und eine Belegung ergeben zusammen die Bedeutung für Terme durch folgende Abbildungen:

- $\mathcal{I}_\xi(x) = \xi(x)$ für jede Variable x
- $\mathcal{I}_\xi(c) = \mathcal{I}(c)$ für jede Konstante c
- für n-stelliges Funktionssymbol f und Terme $t_1, \ldots, t_n$ gilt $\mathcal{I}_\xi(f(t_1, \ldots, t_n)) = \mathcal{I}(f)(\mathcal{I}_\xi(t_1), \ldots, \mathcal{I}_\xi(t_n))$

Zusätzlich müssen die folgenden Bedingungen erfüllt sein:

- Wenn $S \sqsubseteq T$, dann gilt $\mathcal{U}_S \subseteq \mathcal{U}_T$.
- Hat ein Term t die Sorte S, so gilt $\mathcal{I}_\xi(t) \in \mathcal{U}_S$.
- Für jeden wohlsortierten Term $f(t_1, \ldots, t_n)$ ist $(\mathcal{I}_\xi(t_1), \ldots, \mathcal{I}_\xi(t_n))$ im Definitionsbereich von $\mathcal{I}(f)$.

Analog zum unsortierten Fall gilt:

- $\mathcal{I}_\xi \models P(t_1, \ldots, t_n))$ gdw $\mathcal{I}(P)(\mathcal{I}_\xi(t_1), \ldots, \mathcal{I}_\xi(t_n))$.
- $\mathcal{I}_\xi \models (\neg\varphi)$ gdw nicht $\mathcal{I}_\xi \models \varphi$

- $\mathcal{I}_\xi \models (\varphi \wedge \psi)$ gdw $\mathcal{I}_\xi \models \varphi$ und $\mathcal{I}_\xi \models \psi$
- $\mathcal{I}_\xi \models (\varphi \vee \psi)$ gdw $\mathcal{I}_\xi \models \varphi$ oder $\mathcal{I}_\xi \models \psi$
- $\mathcal{I}_\xi \models (\varphi \Rightarrow \psi)$ gdw wenn $\mathcal{I}_\xi \models \varphi$ dann $\mathcal{I}_\xi \models \psi$
- $\mathcal{I}_\xi \models (\varphi \Leftrightarrow \psi)$ gdw $\mathcal{I}_\xi \models \varphi$ genau dann, wenn $\mathcal{I}_\xi \models \psi$
- $\mathcal{I}_\xi \models (\forall x_S.\, \varphi)$ gdw für alle $a \in \mathcal{U}_S$ gilt $\mathcal{I}_{\xi[x \leftarrow a]} \models \varphi$
- $\mathcal{I}_\xi \models (\exists x_S.\, \varphi)$ gdw es gibt ein $a \in \mathcal{U}_S$ mit $\mathcal{I}_{\xi[x \leftarrow a]} \models \varphi$
- Hat man eine Logik mit Gleichheit, so wird zusätzlich definiert: $\mathcal{I}_\xi \models (t_1 \doteq t_2)$ gdw $\mathcal{I}_\xi(t_1) = \mathcal{I}_\xi(t_2)$. ■

Da $\mathcal{I}(f)$ eine partielle Funktion ist, macht der Ausdruck $\mathcal{I}(f)(\mathcal{I}_\xi(t_1), \ldots, \mathcal{I}_\xi(t_n))$ nur Sinn, wenn $f(t_1, \ldots, t_n)$ wohlsortiert ist, da dann $\mathcal{I}(f)$ nur auf den entsprechenden Termen definiert sein muß.
Die Begriffe „Erfüllbarkeit“, „Modell“, „Folgerung“, „erfüllbar“, „unerfüllbar“, „Tautologie“, „Satz“ sind wie im unsortierten Fall definiert.

2.16 Beispiel: Sei $\mathbf{SORT} = \{\mathsf{gerade}, \mathsf{ungerade}, \mathbb{N}\}$,
$\mathbf{SD} = \{\mathsf{gerade} \sqsubseteq \mathbb{N}, \mathsf{ungerade} \sqsubseteq \mathbb{N}\}$,
$\mathbf{TD} = \{1 {<} \mathbb{N}, + {<} \mathbb{N} \times \mathbb{N} \rightarrow \mathbb{N}, + {<} \mathsf{gerade} \times \mathsf{gerade} \rightarrow \mathsf{gerade},$
$x_{\mathbb{N}} + x_{\mathbb{N}} {<} \mathsf{gerade}, x_{\mathbb{N}} + s(x_{\mathbb{N}}) {<} \mathsf{ungerade}, a_2 {<} \mathsf{gerade}, a_4 {<} \mathsf{gerade}, a_8 {<} \mathsf{gerade}\}$
$\mathbf{PD} = \{\mathsf{teilt} {<} \mathbb{N} \times \mathbb{N}\}$
$\Gamma = \{\forall n_{\mathbb{N}}.\, \forall m_{\mathbb{N}}.\, \forall k_{\mathbb{N}}.\, \mathsf{teilt}(n, m) \wedge \mathsf{teilt}(m, k) \Rightarrow \mathsf{teilt}(n, k)$
$\mathsf{teilt}(a_2, a_4), \mathsf{teilt}(a_4, a_8)\}$
$\varphi = \mathsf{teilt}(a_2, a_8)$
Ein mögliches Σ-Modell für Γ besteht in der folgenden Struktur:
$\mathcal{U} = \mathcal{U}_{\mathbb{N}} = \{1, 2, 3, 4, 5, 6, 7, \ldots\}$, mit den Teilmengen $\mathcal{U}_{\mathsf{gerade}} = \{2, 4, 6, 8, 10, 12, \ldots\}$, $\mathcal{U}_{\mathsf{ungerade}} = \{1, 3, 5, 7, 9, 11, \ldots\}$, $+$ wird auf die übliche Addition abgebildet und $\mathcal{I}(\mathsf{teilt})$ ist die übliche Teilbarkeitsrelation, die Konstanten a_2, a_4, a_8 werden auf $2, 4, 8$ abgebildet. Dieses Modell erfüllt auch φ. ■

2.17 Definition (Wohlsortierte Substitution): Eine *Substitution* σ heißt wohlsortiert, wenn $S_\Sigma(\sigma(x)) \supseteq S_\Sigma(x)$. Die Menge aller wohlsortierten Substitutionen wird mit $\mathbf{SUB}_\Sigma$ bezeichnet, entsprechend die der (unsortierten) Substitutionen mit $\mathbf{SUB}_{\bar{\Sigma}}$. ■

2.18 Definition (Vergessensfunktor): Abschließend definieren wir einen Operator „quer“ $\bar{}$ als eine Operation auf Termen und Substitutionen, der sich als Vergessensfunktor verhält: $\bar{} : \Sigma \rightarrow \bar{\Sigma}$, $\bar{} : \mathbf{T}_\Sigma \rightarrow \bar{\mathbf{T}}_\Sigma$ und $\bar{} : \mathbf{SUB}_\Sigma \rightarrow \mathbf{SUB}_{\bar{\Sigma}}$. Der Vergessensfunktor ist injektiv. ■

Kapitel 3

Relativierung

In diesem Abschnitt wollen wir uns näher ansehen, wie die sortierten Logiken mit den unsortierten Logiken zusammenhängen. Am Beispiel von Seite 10 und des Problems von Aufgabe Übungsblatt 1, Aufgabe 2 und Übungsblatt 2, Aufgabe 1 haben wir gesehen, daß die Ausdrucksstärke von sortierter und unsortierter Logik gleich groß sind. Wir wollen uns diesen Zusammenhang nun formal ansehen. Wir gehen von einer sortierten Logik mit Signatur Σ und einer Formelmenge Γ aus und wollen eine unsortierte Logik mit Signatur $\Re(\Sigma)$ und Formelmenge $\Re(\Gamma)$ angeben, so daß $\Re(\Gamma)$ unerfüllbar ist genau dann, wenn Γ unerfüllbar ist. Zuerst geben wir eine allgemeine Formelrelativierung an, mit der beliebige sortierte Formeln in unsortierte Formeln übersetzt werden können, und dann eine spezielle Termrelativierung, die nur für einfache, baumartige Signaturen funktioniert, aber im Deduktionsverhalten de facto gleich effizient ist wie die sortierte Formulierung.

3.1 Die Formelrelativierung

3.1 DEFINITION (RELATIVIERUNG): Wir definieren die *Formelrelativierung*, kurz *Relativierung*, $\Re$ als Funktor:

- Der Signatur Σ einer sortierten Logik wird die Signatur $\Re(\Sigma)$ einer unsortierten Logik zugeordnet mit $\Re(\Sigma) = \bar{\Sigma} \cup \Re(\mathbf{SORT})$ wobei $\Re$ jedem Sortensymbol S aus **SORT** ein einstelliges Prädikatssymbol $\Re(S) = \hat{\Re}(S)$ zuordnet, das wir der Einfachheit halber oft wieder mit S bezeichnen. Es gilt $\Re(\mathbf{SORT}) = \{\Re(S) \,|\, S \in \mathbf{SORT}\}$
- Für Formelmengen Γ definieren wir die Relativierung als eine Abbildung $\Re(\Gamma) = \left\{\hat{\Re}(\varphi) \,|\, \varphi \in \Gamma\right\} \cup \mathbf{AX}_\Sigma$

Dabei wird $\hat{\Re}$ definiert als:

- $\hat{\Re}$ ist der Vergessensfunktor auf Variablen, Konstanten und zusammengesetzen Termen sowie atomaren Formeln.

- Sind φ und ψ zwei sortierte Formeln so gilt $\hat{\Re}(\varphi * \psi) = \hat{\Re}(\varphi) * \hat{\Re}(\psi)$ mit $* \in \{\wedge, \vee, \Rightarrow, \Leftrightarrow\}$ und $\hat{\Re}(\neg\varphi) = \neg\hat{\Re}(\varphi)$.
- Für quantifizierte sortierte Formeln gilt:
  $\hat{\Re}(\forall x_S\text{.}\, \varphi) = \forall x\text{.}\, S(x) \Rightarrow \hat{\Re}(\varphi)$ und
  $\hat{\Re}(\exists x_S\text{.}\, \varphi) = \exists x\text{.}\, S(x) \wedge \hat{\Re}(\varphi)$

Sei $\Sigma = \langle \bar{\Sigma}, \mathbf{SORT}, \mathfrak{s}, \mathbf{TD}, \mathbf{SD}, \mathbf{PD} \rangle$ eine sortierte Signatur. Die Signaturaxiome $\mathbf{AX}_\Sigma$ zu dieser Signatur besteht aus der folgenden Formelmenge:

$$\begin{array}{ll} \{\exists x\text{.}\, \Re(S)(x) \mid S \in \mathbf{SORT}\} & \cup \\ \left\{\overline{\Re(S)(t)} \mid (t{<}S) \in \mathbf{TD}_\Sigma\right\} & \cup \\ \{\forall x\text{.}\, \Re(S)(x) \Rightarrow \Re(T)(x) \mid (S \sqsubseteq T) \in \mathbf{SD}_\Sigma\} & \end{array}$$

Dabei bezeichnen wir mit $\overline{\varphi}$ den universellen Abschluß der Formel φ, das heißt die Formel, die aus φ entsteht, wenn man über alle in φ vorkommenden Variablen universell quantifiziert. Zum Beispiel ist der universelle Abschluß der Formel $\forall x\text{.}\, P(x, y)$ die Formel $\forall y\text{.}\, \forall x\text{.}\, P(x, y)$. ■

3.2 Beispiel: Sei eine Signatur Σ gegeben mit

$$\begin{array}{l} \mathbf{SORT} = \{\mathsf{gerade}, \mathbb{N}\}, \\ \mathbf{TD} = \{0{<}\mathsf{gerade}, s{<}\mathbb{N} \to \mathbb{N}, s(s(x_{\mathsf{gerade}})){<}\mathsf{gerade}\} \\ \mathbf{SD} = \{\mathsf{gerade} \sqsubseteq \mathbb{N}\}, \\ \mathbf{PD} = \{> \,{<}\mathbb{N} \times \mathbb{N}\} \end{array}$$

Weiterhin haben wir die Formelmenge

$$\Gamma = \{\forall n_{\mathbb{N}}\text{.}\, \exists m_{\mathsf{gerade}}\text{.}\, m > n, \neg\exists n_{\mathsf{gerade}}\text{.}\, \forall m_{\mathbb{N}}\text{.}\, n > m\}$$

Dann ergibt sich $\hat{\Re}(\Gamma)$ zu[1]

$$\begin{array}{l} \{\forall n\text{.}\, \mathbb{N}(n) \Rightarrow \exists m\text{.}\, \mathsf{gerade}(m) \wedge m > n, \\ \ \neg\exists n\text{.}\, \mathsf{gerade}(n) \wedge \forall m\text{.}\, \mathbb{N}(m) \Rightarrow n > m\} \end{array}$$

Die Signaturaxiome $\mathbf{AX}_\Sigma$ ergeben sich zu:

$$\begin{array}{ll} \{\exists x\text{.}\, \mathbb{N}(x), \exists x\text{.}\, \mathsf{gerade}(x)\} & \cup \\ \{\mathsf{gerade}(0), \forall x\text{.}\, \mathbb{N}(x) \Rightarrow \mathbb{N}(s(x)), \forall x\text{.}\, \mathsf{gerade}(x) \Rightarrow \mathsf{gerade}(s(s(x)))\} & \cup \\ \{\forall x\text{.}\, \mathsf{gerade}(x) \Rightarrow \mathbb{N}(x)\} & \end{array}$$

Die Relativierung $\Re(\Gamma)$ ergibt sich als Vereinigung von $\hat{\Re}(\Gamma)$ und $\mathbf{AX}_\Sigma$. ■

Es gilt der folgende wichtige Zusammenhang, das sogenannte Sortentheorem, das die sortierte und die unsortierte Formulierung in Zusammenhang setzt:

[1] Wir schreiben statt $\hat{\Re}(\mathbb{N})$ und $\hat{\Re}(\mathsf{gerade})$ wieder kurz $\mathbb{N}$ und gerade.

3.3 THEOREM: *Sei* Γ *eine sortierte Formelmenge bezüglich einer sortierten Signatur* Σ*, dann ist* Γ *genau dann erfüllbar, wenn* $\Re(\Gamma)$ *in* $\Re(\Sigma)$ *unsortiert erfüllbar ist.*

Beweis:

„$\Rightarrow$“: Sei Γ eine eine erfüllbare sortierte Formelmenge bezüglich der sortierten Signatur Σ, dann gibt es ein sortiertes Modell $\mathcal{M} = \langle \mathcal{U}, \mathcal{I} \rangle$ von Γ, so daß für alle Belegungen ξ gilt $\mathcal{I}_\xi \models \Gamma$. Wir konstruieren daraus ein Modell $\widetilde{\mathcal{M}} = \langle \widetilde{\mathcal{U}}, \widetilde{\mathcal{I}} \rangle$ von $\Re(\Gamma)$, so daß für alle Belegungen $\tilde{\xi}$ gilt $\widetilde{\mathcal{I}}_{\tilde{\xi}} \models \Re(\Gamma)$. Dazu definieren wir $\widetilde{\mathcal{U}} := \mathcal{U}$, $\widetilde{\mathcal{I}}(\Re(c)) = \mathcal{I}(c)$ und $\widetilde{\mathcal{I}}(\Re(f))$ ist eine totale Funktion, die mit der partiellen Funktion $\mathcal{I}(f)$ auf deren Definitionsbereich übereinstimmt und wählen zur Belegung $\tilde{\xi}$ die Belegung $\xi := \tilde{\xi}$. $\widetilde{\mathcal{I}}(\Re(P))$ wird definiert als $\mathcal{I}(P)$. Für die neuen Prädikatssymbole $\Re(S)$ definieren wir: $\widetilde{\mathcal{I}}(\Re(S))$ ist die Relation, die für $u \in \mathcal{U}$ genau dann erfüllt ist, wenn $u \in \mathcal{U}_S$. Die Behauptung ist, daß es sich dabei um ein Modell von $\Re(\Gamma)$ handelt. Der Beweis, daß $\widetilde{\mathcal{M}}$ ein Modell von $\Re(\Gamma)$ ist, zeigt man zum einen darüber, daß man induktiv über den Aufbau der Formeln zeigt, daß für alle Terme und Formeln gilt: $\widetilde{\mathcal{I}}_{\tilde{\xi}} \circ \hat{\Re} = \mathcal{I}_\xi$.[2] Die Beziehung ist für Variable, Konstanten, zusammengesetzen Termen, atomare und durch Junktoren verknüpfte Formeln trivial. Die einzig interessanten Fälle sind die Quantifizierungen.

Für Allquantoren ergibt sich:

$\widetilde{\mathcal{I}}_{\tilde{\xi}}(\hat{\Re}(\forall x_S.\, \varphi))$	gdw
$\widetilde{\mathcal{I}}_{\tilde{\xi}}(\forall x.\, S(x) \Rightarrow \hat{\Re}(\varphi))$	gdw
Für alle u in $\mathcal{U}$ gilt $\widetilde{\mathcal{I}}_{\tilde{\xi}[x\leftarrow u]}(S(x) \Rightarrow \hat{\Re}(\varphi))$	gdw
Für alle u in $\widetilde{\mathcal{U}}$ gilt $\widetilde{\mathcal{I}}(\hat{\Re}(S))(u) \Rightarrow \widetilde{\mathcal{I}}_{\tilde{\xi}[x\leftarrow u]}(\hat{\Re}(\varphi))$	gdw (nach Induktionsvor.)
Für alle u in $\widetilde{\mathcal{U}}$ gilt $u \in \mathcal{U}_S \Rightarrow \mathcal{I}_{\xi[x_S\leftarrow u]}(\varphi)$	gdw
Für alle u in $\mathcal{U}_S$ gilt $\mathcal{I}_{\xi[x_S\leftarrow u]}(\varphi)$	gdw
$\mathcal{I}_\xi(\forall x_S.\, \varphi)$.	

Für Existenzquantoren erhält man:

$\widetilde{\mathcal{I}}_{\tilde{\xi}}(\hat{\Re}(\exists x_S.\, \varphi))$	gdw
$\widetilde{\mathcal{I}}_{\tilde{\xi}}(\forall x.\, S(x) \wedge \hat{\Re}(\varphi))$	gdw
Es gibt ein u in $\mathcal{U}$ mit $\widetilde{\mathcal{I}}_{\tilde{\xi}[x\leftarrow u]}(S(x) \wedge \hat{\Re}(\varphi))$	gdw
Es gibt ein u in $\widetilde{\mathcal{U}}$ mit $\widetilde{\mathcal{I}}(\hat{\Re}(S))(u) \wedge \widetilde{\mathcal{I}}_{\tilde{\xi}[x\leftarrow u]}(\hat{\Re}(\varphi))$	gdw (nach Induktionsvor.)
Es gibt ein u in $\widetilde{\mathcal{U}}$ mit $u \in \mathcal{U}_S \wedge \mathcal{I}_{\xi[x_S\leftarrow u]}(\varphi)$	gdw
Es gibt ein u in $\mathcal{U}_S$ mit $\mathcal{I}_{\xi[x_S\leftarrow u]}(\varphi)$	gdw
$\mathcal{I}_\xi(\exists x_S.\, \varphi)$.	

Zusätzlich muß man nun noch zeigen, daß die Signaturaxiome in $\widetilde{\mathcal{M}}$ erfüllt sind: Die Formeln $\exists x.\, \Re(S)(x)$ sind für alle Sorten S aus **SORT** erfüllt, da $\mathcal{U}_S$ nicht-leer

[2] Wobei wir für Formeln „=“ als Äquivalenz auffassen.

ist, es also ein $u_S \in \mathcal{U}_S$ gibt, so daß $\mathcal{I}_{\xi[x \leftarrow u_S]}(\Re(S)(x))$ erfüllt ist.
Die Formeln $\overline{\Re(S)(t)}$ für $(t \lessdot S) \in \mathbf{TD}_\Sigma$ sind in $\widetilde{\mathcal{M}}$ erfüllt, da $\widetilde{\mathcal{I}}_{\widetilde{\xi}}(\Re(S)(t))$ gdw $\widetilde{\mathcal{I}}_{\widetilde{\xi}}(t) \in \mathcal{U}_S$ gdw $\mathcal{I}_\xi(t) \in \mathcal{U}_S$ für alle ξ.
Die Formeln $\forall x.\, \Re(S)(x) \Rightarrow \Re(T)(x)$ sind erfüllt für $(S \sqsubseteq T) \in \mathbf{SD}_\Sigma$ wegen $\mathcal{U}_S \subseteq \mathcal{U}_T$ und der Definition von Teilmenge.

„$\Leftarrow$“: Nun nehmen wir an, daß wir ein Model $\mathcal{M}$ von $\Re(\Gamma)$ haben. Wir definieren ein Model $\widetilde{\mathcal{M}}$ von Γ durch:

- $\widetilde{\mathcal{U}}_S := \left\{u \in \mathcal{U} \,|\, \mathcal{I}(\hat{\Re}(S))(u)\right\}$ und $\widetilde{\mathcal{U}} = \mathcal{U}$
- $\widetilde{\mathcal{I}}(c) := \mathcal{I}(\hat{\Re}(c))$
- $\widetilde{\mathcal{I}}(f) := \mathcal{I}(\hat{\Re}(f))$

Der Beweis, daß es sich dabei um ein sortiertes Modell handelt erfolgt wieder über vollständige Induktion über den Term- und Formelaufbau: $\widetilde{\mathcal{I}}_{\widetilde{\xi}} = \mathcal{I}_\xi \circ \hat{\Re}$. Es ist wieder nur die Quantifikation interessant. Wir zeigen es nur für die Allquantifizierung.

$\widetilde{\mathcal{I}}_{\widetilde{\xi}}(\forall x_S.\, \varphi)$	gdw
Für alle u in $\widetilde{\mathcal{U}}_S$ gilt $\widetilde{\mathcal{I}}_{\widetilde{\xi}[x \leftarrow u]}(\varphi)$	gdw (nach Induktionsvor.)
Für alle u in $\mathcal{U}$ gilt $\mathcal{I}(\hat{\Re}(S))(u) \Rightarrow \mathcal{I}_{\xi[x \leftarrow u]}(\hat{\Re}(\varphi))$	gdw
Für alle u in $\mathcal{U}$ gilt $\mathcal{I}_{\xi[x \leftarrow u]}(\hat{\Re}(S)(x) \Rightarrow \hat{\Re}(\varphi))$	gdw
$\mathcal{I}_\xi(\forall x.\, \hat{\Re}(S)(x) \Rightarrow \hat{\Re}(\varphi))$	gdw
$\mathcal{I}_\xi(\hat{\Re}(\forall x_S.\, \varphi))$	

Die Bedingungen an Untersorten und Termdeklarationen ergeben sich aus den Signaturaxiomen. Aus der ersten Klasse von Signaturaxiomen ergibt sich, daß die $\mathcal{U}_S$ nicht-leer sind. ■

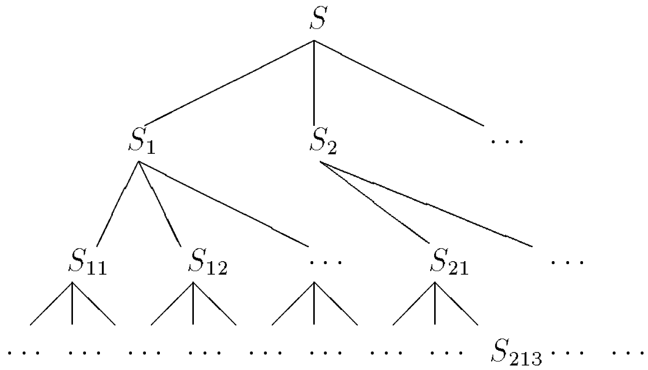
3.4 KOROLLAR: *Sei* Γ *eine sortierte Formelmenge bezüglich einer sortierten Signatur* Σ*, dann ist* Γ *genau dann unerfüllbar (allgemeingültig), wenn* $\Re(\Gamma)$ *in* $\Re(\Sigma)$ *unsortiert unerfüllbar (allgemeingültig) ist.*

Beweis: Die Äquivalenz der Unerfüllbarkeit ergibt sich direkt aus Theorem 3.3. Die Äquivalenz der Allgemeingültigkeit erhält man aus der Eineindeutigkeit der Modellbeziehung. ■

3.2 Die Termrelativierung

Durch die Formelrelativierung erhalten wir eine unsortierte Formulierung, die die Nachteile hat, daß verglichen mit der sortierten Formulierung die Formeln meist umständlich sind, und von daher in der Suche sich der Suchraum auch wesentlich vergrößert. In diesem Kapitel werden wir nun für einfache, baumartige Signaturen eine Termrelativierung angeben, also eine Übersetzung in unsortierte Logik, die diese Nachteile nicht hat.

3.5 Definition (Termrelativierung): Sei Σ eine einfache, baumartige Signatur (vergleiche Definition 2.4) mit folgender Sortenstruktur:



Das heißt wir verwenden Multiindizes $I = (i_1 i_2 \cdots i_n)$ so daß $S_{i_1 i_2 \cdots i_k i_{k+1}}$ Untersorte von $S_{i_1 i_2 \cdots i_k}$ ist. Die Topsorte wird ohne Index mit S bezeichnet, was dem leeren Multiindex entspricht. Da in einfachen Signaturen jedes Konstantensymbol und jedes Funktionssymbol eine eindeutig bestimmte Sorte haben, geben wir diese Sorte als Index am Symbol an. Wir definieren die *Termrelativierung*, $\Re^{\mathrm{T}}$ als Funktor:

- Der Signatur Σ einer sortierten Logik wird die Signatur $\Re^{\mathrm{T}}(\Sigma)$ einer unsortierten Logik zugeordnet mit $\Re^{\mathrm{T}}(\Sigma) = \overline{\Sigma} \cup \Re^{\mathrm{T}}(\mathbf{SORT})$ wobei $\Re^{\mathrm{T}}$ jedem Sortensymbol S aus **SORT** ein einstelliges Funktionsssymbol $\Re^{\mathrm{T}}(S)$ zuordnet, das wir der Einfachheit halber oft wieder mit S bezeichnen. Es gilt $\Re^{\mathrm{T}}(\mathbf{SORT}) = \left\{\Re^{\mathrm{T}}(S) \,|\, S \in \mathbf{SORT}\right\}$.
- Zu einer Sorte S_I mit Multiindex $I = (i_1 i_2 \cdots i_n)$ definieren wir $\overline{S_I}$ als die Funktion $\lambda(x)[S(S_{i_1}(S_{i_1 i_2}(\cdots(S_I(x))\cdots)))]$, d.h. $\overline{S_I} = S \circ S_{i_1} \circ S_{i_1 i_2} \circ \cdots \circ S_I$.
- Für Formelmengen Γ definieren wir die Relativierung als eine Abbildung $\Re^{\mathrm{T}}(\Gamma) = \left\{\Re^{\mathrm{T}}(\varphi) \,|\, \varphi \in \Gamma\right\}$

Dabei wird $\Re^{\mathrm{T}}$ auf Termen und Formeln definiert als:

- $\Re^{\mathrm{T}}(c_{S_I}) = \overline{S_I}(c)$
- $\Re^{\mathrm{T}}(x_{S_I}) = \overline{S_I}(x)$

– $\Re^{\mathrm{T}}(f_{(S_{I_1}\times\cdots\times S_{I_n}\to S_J)}(t_1,\ldots,t_n)) = \overline{S_J}(f(\Re^{\mathrm{T}}(t_1),\ldots,\Re^{\mathrm{T}}(t_n)))$

– $\Re^{\mathrm{T}}(P(t_1,\ldots,t_n)) = P(\Re^{\mathrm{T}}(t_1),\ldots,\Re^{\mathrm{T}}(t_n))$

– Sind φ und ψ zwei sortierte Formeln so gilt $\Re^{\mathrm{T}}(\varphi * \psi) = \Re^{\mathrm{T}}(\varphi) * \Re^{\mathrm{T}}(\psi)$ mit $* \in \{\wedge, \vee, \Rightarrow, \Leftrightarrow\}$ und $\Re^{\mathrm{T}}(\neg\varphi) = \neg\Re^{\mathrm{T}}(\varphi)$.

– Für quantifizierte sortierte Formeln gilt:
$\Re^{\mathrm{T}}(\forall x_{S_I}.\, \varphi) = \forall x.\, \Re^{\mathrm{T}}(\varphi)$ und $\Re^{\mathrm{T}}(\exists x_{S_I}.\, \varphi) = \exists x.\, \Re^{\mathrm{T}}(\varphi)$ ■

3.6 Beispiel: Sei eine Signatur Σ gegeben mit
SORT $= \{\mathsf{gerade}, \mathbb{N}\}$, **TD** $= \{0{<}\mathsf{gerade}, s{<}\mathbb{N} \to \mathbb{N}\}$,
SD $= \{\mathsf{gerade} \sqsubseteq \mathbb{N}\}$, **PD** $= \{> {<}\mathbb{N} \times \mathbb{N}\}$
Weiterhin haben wir die Formelmenge

$$\Gamma = \{\forall n_{\mathbb{N}}.\, \exists m_{\mathsf{gerade}}.\, m > n, \neg\exists n_{\mathsf{gerade}}.\, \forall m_{\mathbb{N}}.\, n > m\}$$

Dann ergibt sich $\Re^{\mathrm{T}}(\Gamma)$ zu[3]

$$\{\forall n.\, \exists m.\, \mathbb{N}(\mathsf{gerade}(m)) > \mathbb{N}(n), \neg\exists n.\, \forall m.\, \mathbb{N}(\mathsf{gerade}(n)) > \mathbb{N}(m)\}$$ ■

Es gilt der folgende wichtige Zusammenhang, das sogenannte Sortentheorem, das die sortierte und die unsortierte Formulierung in Zusammenhang setzt:

3.7 Theorem: *Sei Γ eine sortierte Formelmenge bezüglich einer sortierten einfachen, baumartigen Signatur Σ, dann ist Γ genau dann erfüllbar, wenn $\Re^{\mathrm{T}}(\Gamma)$ in $\Re^{\mathrm{T}}(\Sigma)$ unsortiert erfüllbar ist.*

Beweis: Der Beweis läuft analog zu dem der Formelrelativierung. Er ist im folgenden skizziert.
„$\Rightarrow$“: Sei Γ eine eine erfüllbare sortierte Formelmenge bezüglich der sortierten Signatur Σ, dann gibt es ein sortiertes Modell $\mathcal{M} = \langle \mathcal{U}, \mathcal{I} \rangle$ von Γ, so daß für alle Belegungen ξ gilt $\mathcal{I}_\xi \models \Gamma$. Wir konstruieren daraus ein Modell $\widetilde{\mathcal{M}} = \langle \widetilde{\mathcal{U}}, \widetilde{\mathcal{I}} \rangle$ von $\Re^{\mathrm{T}}(\Gamma)$, so daß für alle Belegungen $\tilde{\xi}$ gilt $\widetilde{\mathcal{I}}_{\tilde{\xi}} \models \Re^{\mathrm{T}}(\Gamma)$. Dazu definieren wir $\widetilde{\mathcal{U}} := \mathcal{U}$, $\widetilde{\mathcal{I}}(\Re^{\mathrm{T}}(c)) = \mathcal{I}(c)$ und für $u_i \in \mathcal{U}$ gilt mit $S_I = (S_1 \times \cdots \times S_n \to T)$

$$\widetilde{\mathcal{I}}(\Re^{\mathrm{T}}(f_{S_I})(u_1,\ldots,u_n)) = \begin{cases} \mathcal{I}(f_{S_I})(u_1,\ldots,u_n) & \text{falls alle } u_i \in \mathcal{U}_{S_i} \\ \text{beliebiges Element aus } \mathcal{U}_T & \text{sonst} \end{cases}$$

Nun müssen noch Interpretationen für die neuen Funktionssymbole $\Re^{\mathrm{T}}(S_I)$ angegeben werden. Dafür definieren wir $\widetilde{\mathcal{I}}(\Re^{\mathrm{T}}(S_I)) : \widetilde{\mathcal{U}} \stackrel{\mathrm{pr}}{\twoheadrightarrow} \mathcal{U}_{S_I}$ das heißt $\widetilde{\mathcal{I}}(\Re^{\mathrm{T}}(S_I)) \mid_{\mathcal{U}_{S_I}} = \mathrm{id}_{\mathcal{U}_{S_I}}$ und sonst beliebiges Element in $\mathcal{U}_{S_I}$.
Man kann dann induktiv zeigen, daß gilt:

$$\widetilde{\mathcal{I}}_{\tilde{\xi}} \circ \Re^{\mathrm{T}} = \mathcal{I}_\xi$$

„$\Leftarrow$“: Die Vollständigkeit ergibt sich durch die Definition von $\mathcal{U}_{S_I} := \mathrm{Bild}(\mathcal{I}(\overline{S_I})) \subseteq \mathcal{U}$ und eine entsprechende Modellkonstruktion. ■

[3]Wir schreiben statt $\Re^{\mathrm{T}}(\mathbb{N})$ und $\Re^{\mathrm{T}}(\mathsf{gerade})$ wieder kurz $\mathbb{N}$ und gerade.

Gleichheitsbehandlung und Relativierung

Bislang haben wir uns nicht speziell um das Gleichheitsprädikat gekümmert. Es läßt sich zwar nicht direkt in Logik erster Stufe definieren, aber es läßt sich eine widerlegungsäquivalente Formulierung finden, in der die Gleichheit umschrieben wird durch die Reflexivität: $\forall x.\, x \doteq x$, Symmetrie: $\forall x, y.\, x \doteq y \Rightarrow y \doteq x$, Transitivität $\forall x, y, z.\, x \doteq y \wedge y \doteq z \Rightarrow x \doteq z$, sowie die Substitutionseigenschaft, das heißt Formeln der Gestalt $\forall x_1, \ldots, x_i, \ldots, x_n.\, \forall y.\, x_i \doteq y \Rightarrow f(x_1, \ldots, x_i, \ldots, x_n) \doteq f(x_1, \ldots, y, \ldots, x_n)$ und $\forall x_1, \ldots, x_i, \ldots, x_n.\, \forall y.\, x_i \doteq y \Rightarrow P(x_1, \ldots, x_i, \ldots, x_n) \Leftrightarrow P(x_1, \ldots, y, \ldots, x_n)$ für alle vorkommenden Funktionssymbole f und Prädikatssymbole P und alle Stellen i. Dieses Verfahren ist allerdings ziemlich ineffizient, weswegen man die Paramodulation erfunden hat – diese ist zwar um Größenordnungen besser, aber selbst noch stark ansteuerungsbedürftig – und als Regel integriert hat. Sie besagt grob, daß man Gleiches durch Gleiches ersetzen darf und lautet:

$$\frac{\begin{array}{ll} t \doteq s \vee K_1 \vee \ldots \vee K_n & \\ L[\ldots t' \ldots] \vee M_1 \vee \ldots \vee M_m & \sigma(t') = \sigma(t) \end{array}}{\sigma(L[\ldots s \ldots]) \vee \sigma(M_1) \vee \ldots \vee \sigma(M_m) \vee \sigma(K_1) \vee \ldots \vee \sigma(K_n)}$$

Wir haben ja bislang noch keinen Resolutionskalkül für sortierte Logik angegeben. Deshalb sei ohne Beweis angemerkt, daß die Komplexität eines Resolutionsbeweises eines sortierten Theorems in sortierter Logik (mit sortierter Unifikation) gleich der Komplexität eines unsortierten Resolutionsbeweises der Termrelativierung des sortierten Theorems in unsortierter Logik (mit unsortierter Unifikation) ist. Für eine Gleichheitslogik, also in Gegenwart von Paramodulation ist dies allerdings nicht mehr direkt der Fall, wenn man in Variable paramoduliert. Sei zum Beispiel eine einfache, baumartige Signatur Σ gegeben mit $\mathbf{SORT} = \{S, T\}$, $\mathbf{TD} = \{a{<}S, f{<}S \to T\}$. Die Formelmenge $\Gamma = \{\forall x_S.\, x \doteq a; \forall y_T.\, f(a) \doteq y\}$ hat dann die Termrelativierung $\Re^{\mathrm{T}}(\Gamma) = \{\forall x.\, S(x) \doteq S(a), \forall y.\, T(f(S(a))) \doteq T(y)\}$. Die Klauselnormalform der sortierten Formelmenge ist $\mathcal{C}_\Gamma = \{x_S \doteq a; f(a) \doteq y_T\}$. Dort ist im wesentlichen nur der einzige (sortierte) Paramodulationsschritt a wird x_S in $f(a)$ möglich, was zur Klausel $f(x_S) \doteq y_T$ führt. Die termrelativierte Klauselnormalform lautet $\mathcal{C}_{\Re^{\mathrm{T}}(\Gamma)} = \{S(x) \doteq S(a), T(f(S(a))) \doteq T(y)\}$, hier ist es möglich x in der ersten Klausel mit $T(f(S(a)))$ zu unifizieren und durch $T(y)$ zu ersetzen. Es ergibt sich (bis auf Variablenumbenennung) die Klausel $S(T(x)) \doteq S(a)$. In dieser Klausel kann man nun wiederum x mit $T(f(S(a)))$ unifizieren und durch $T(y)$ ersetzen, was $S(T(T(x))) \doteq S(a)$ ergibt. Auf diese Weise kann man alle Klauseln der Form $S(T^n(x)) \doteq S(a)$ herleiten, wobei T^n für die n-malige Anwendung der Funktion T steht. In anderen Worten während die sortierte Formulierung einen endlichen Suchraum aufspannt, ist dies für die termrelativierte Formulierung nicht der Fall.
Dieses Problem tritt dann nicht auf, wenn man eine Paramodulation in Variablenpositionen verbietet.

Kapitel 4

Übersetzung von höherer Stufe in flachsortierte erste Stufe

In Kapitel 1 haben wir die unsortierte Prädikatenlogik erster Stufe kennengelernt, sie ist zwar für viele Probleme sehr gut geeignet, aber zur Darstellung mathematischer Sachverhalte ist es oft auch nötig, über Prädikate oder Funktionen zu quantifizieren. Beispielsweise lassen sich selbst die natürlichen Zahlen nicht endlich in der Prädikatenlogik erster Stufe ausdrücken, da man das folgende Induktionsaxiom (oder ein vergleichbares) braucht:

$$\forall P^{pred}\text{.}\, P(0) \wedge (\forall n^{\iota}\text{.}\, P(n) \Rightarrow P(s(n)) \Rightarrow \forall n^{\iota}\text{.}\, P(n)$$

Es gibt aber auch speziell für die KI wichtige Formeln höherer Stufe, so ist zum Beispiel die Circumscription-Formel (vgl. [GN87, S.134]) zweiter Stufe. Diese beiden Formeln sind echt höherer Stufe und lassen sich in ihrer vollen Allgemeinheit nicht durch Formeln erster Stufe ersetzen. Um mögliche Paradoxien wie die Russellsche zu vermeiden, ist es üblich, allen Variablen und Konstanten von Logiken höherer Stufe einen Typ zuzuordnen, so daß Formeln der Gestalt $P(P)$ ausgeschlossen werden. Typen wurden von Bertrand Russell entwickelt [Rus08] und von Alonzo Church zu einfachen Typen vereinfacht [Chu40], die wir nun verwenden.

4.1 Definition (Typen, Terme): *Typen* der Logik höheren Stufe werden induktiv aus den Basistypen der *Ordnung* 0, nämlich ι, dem Typ für Individuen und o, dem Typ für Wahrheitswerte, gebildet: Wenn $\tau_1, \ldots, \tau_m$ (mit $m \geq 1$) und σ Typen sind, dann ist $\tau = (\tau_1 \times \cdots \times \tau_m \rightarrow \sigma)$ ein Typ der Ordnung 1 + Maximum der Ordnungen von $\tau_1, \ldots, \tau_m, \sigma$. τ ist der Typ der Funktionen mit Argumenten vom Typ $\tau_1, \ldots, \tau_m$ und einem Wert vom Typ σ.[1]
In der Logik höheren Stufe hat jeder *Term* einen Typ. Man hat zu jedem Typ unendlich viele Variablen x^{τ} des entsprechenden Typs. In der Signatur werden die Typen

[1]Im Beispiel oben entspricht dann der Typ *pred* gerade dem Typ $(\iota \rightarrow o)$. Im folgenden werden wir nur Typen betrachten, bei denen die Konstruktion wie folgt eingeschränkt ist: Wenn $\tau_1, \ldots, \tau_m$ und σ Typen sind mit $\tau_i \neq o$, dann ist $(\tau_1 \times \cdots \times \tau_m \rightarrow \sigma)$ ein Typ.

der Konstantensymbole (Objektkonstanten, Funktionskonstanten und Prädikatskonstanten) festgelegt. Genauer gesagt zu jedem Typ gibt es eine möglicherweise leere Menge von Konstanten des entsprechenden Typs. Variablen und Konstanten des Typs τ sind Terme des Typs τ. Sind dann $t^{(\tau_1 \times \cdots \times \tau_m \to \sigma)}$ und $t_1^{\tau_1}, \ldots, t_n^{\tau_n}$ Terme der angegebenen Typen, so ist $t(t_1^{\tau_1}, \ldots, t_n^{\tau_n})$ ein Term des Typs σ. ∎

4.2 Definition: Eine Logik $2n$-ter Stufe ist eine Sprache, in der alle Terme nur Variablen und Konstanten der Ordnung kleiner oder gleich n enthalten. In einer Logik $(2n-1)$-ter Stufe darf darüber hinaus über keine Variable der Ordnung n quantifiziert werden. ∎

Nun gibt es allerdings keine vollständigen Kalküle für Logiken höherer Stufe bezüglich der intuitiven Standardsemantik [Göd31]. Deshalb hat Leon Henkin einen abgeschwächten Semantikbegriff eingeführt [Hen50], der die Standardsemantik approximiert und für den es möglich ist, vollständige und korrekte Kalküle anzugeben. Bezüglich dieser Semantik kann man auch Übersetzungen von Logik höherer Stufe in die erster Stufe angeben. Üblicherweise geschieht das durch die Einführung einer Mengenlehre, aber es gibt auch direktere Darstellungen, die für die üblichen Deduktionssysteme erster Stufe besser geeignet sind [Ker91]. Allerdings sind in beiden Fällen für eine große Klasse von Problemen zusätzliche Komprehensionsaxiome notwendig.

4.3 Beispiel: Betrachten wir als Beispiel die Aussage, daß alle Menschen alle Eigenschaften vom Vater oder von der Mutter erben, das heißt, daß sich jede ihrer Eigenschaften bei Vater oder Mutter wiederfinden läßt. Dies kann man in Logik zweiter Stufe durch die folgende Formel φ ausdrücken:

$$\forall P^{(\iota \to o)}.\ \forall x^{\iota}.\ \mathsf{mensch}(x) \land P(x) \Rightarrow P(\mathsf{vater}(x)) \lor P(\mathsf{mutter}(x))$$

Durch Instantiierung können wir aus φ die folgende Formel ableiten

$$\forall x^{\iota}.\ \mathsf{mensch}(x) \land \mathit{hat\_Schweißfüße}(x) \Rightarrow \mathit{hat\_Schweißfüße}(\mathsf{vater}(x)) \lor \mathit{hat\_Schweißfüße}(\mathsf{mutter}(x))$$

Dies ist noch relativ einfach zu handhaben. Allerdings will man nun für die Prädikatsvariable P auch ganze Formeln (mit einer freien Variable) einsetzen können. Zum Beispiel könnte man $P(t)$ durch $(\mathit{trinkt\_kaffee}(t) \Rightarrow \mathit{schläft\_schlecht}(t))$ ersetzen. Als Instanz von φ erhält man dann:

$$\forall x^{\iota}.\ \mathsf{mensch}(x) \land (\mathit{trinkt\_kaffee}(x) \Rightarrow \mathit{schläft\_schlecht}(x)) \Rightarrow (\mathit{trinkt\_kaffee}(\mathsf{vater}(x)) \Rightarrow \mathit{schläft\_schlecht}(\mathsf{vater}(x))) \lor (\mathit{trinkt\_kaffee}(\mathsf{mutter}(x)) \Rightarrow \mathit{schläft\_schlecht}(\mathsf{mutter}(x)))$$

Wenn wir ableiten wollen, daß eine Formelmenge bestehend aus φ, der Aussage, daß Max nach dem Genuß von Kaffee nicht schlafen kann, sowie der Aussage, daß seine Eltern beide nach Kaffee sehr gut schlafen, widersprüchlich ist, so benötigen wir bei einer Übersetzung der Aussagen in Logik erster Stufe das folgende Axiom, das es uns erlaubt die Variable P mit einer ganzen Formel gleichzusetzen:
$\exists Q^{(\iota \to o)}.\ \forall x^{\iota}.\ Q(x) \Leftrightarrow (\mathit{trinkt\_kaffee}(x) \Rightarrow \mathit{schläft\_schlecht}(x))$. Nun kann P mit Q

instantiiert werden. ■

Da die Einführung solcher Axiome – von denen es unendlich viele gibt – im allgemeinen schwer zu automatisieren ist, hat man ein spezielles Beweisverfahren für die Logik höherer Stufe entwickelt: den Lambdakalkül mit Unifikation höherer Stufe [Chu40, And71, Hue73]. Der wohl am weitesten entwickelte Beweiser für Logik höherer Stufe ist das TPS-System [AINP90]. Will man in diesem Kalkül die Unerfüllbarkeit der obigen Formelmenge zeigen, so liefert der Unifikationsalgorithmus höherer Stufe den Unifikator $\sigma = [P \leftarrow \lambda x.\, (\mathit{trinkt\_kaffee}(x) \Rightarrow \mathit{schläft\_schlecht}(x))]$, kein zusätzliches Axiom ist notwendig. Damit lassen sich viele Probleme effizient lösen. Allerdings wird die Unifikation selbst als Teilschritt im Problemlösungsverfahren unentscheidbar.
Wie sieht die Semantik nun konkret aus? Wir benutzen die folgende Notation: Seien $A_1, \ldots, A_m$ und B Mengen, dann bezeichnen wir mit $\mathcal{F}(A_1, \ldots, A_m; B)$ die Menge aller Funktionen von $A_1 \times \cdots \times A_m$ nach B.

4.4 Definition (Semantik der Logik höheren Stufe): Eine Interpretation der Logik höheren Stufe besteht aus einer Familie $\{\mathcal{U}_\tau\}_\tau$ und einer Interpretationsabbildung $\mathcal{I}$, wobei die $\mathcal{U}_\tau$ nichtleere Mengen sind, eine für jeden Typ τ, mit $\mathcal{U}_o = \{\mathtt{wahr}, \mathtt{falsch}\}$ und $\mathcal{U}_{(\tau_1 \times \cdots \times \tau_m \rightarrow \sigma)} \subseteq \mathcal{F}(\mathcal{U}_{\tau_1}, \ldots, \mathcal{U}_{\tau_m}; \mathcal{U}_\sigma)$. Die Elemente von $\mathcal{U}_o$ heißen *Wahrheitswerte* und die Elemente von $\mathcal{U}_\iota$ *Individuen*. Die Interpretationsabbildung $\mathcal{I}$ bildet Konstanten des Typs τ in die Menge $\mathcal{U}_\tau$ ab. Ebenso bilden Belegungen ξ Variablen des Typs τ in die Menge $\mathcal{U}_\tau$ ab.
Ansonsten ergibt sich nun die Semantik genau wie im Fall der Logik erster Stufe, das heißt:

- $\mathcal{I}_\xi(x) = \xi(x)$ für jede Variable x
- $\mathcal{I}_\xi(c) = \mathcal{I}(c)$ für jede Konstante c
- für zusammengesetzte Terme $t(t_1, \ldots, t_n)$ gilt:
  $\mathcal{I}_\xi(t(t_1, \ldots, t_n)) = \mathcal{I}_\xi(t)(\mathcal{I}_\xi(t_1), \ldots, \mathcal{I}_\xi(t_n))$

Für Formeln wird definiert:

- $\mathcal{I}_\xi \models P(t_1, \ldots, t_n))$ gdw $\mathcal{I}_\xi(P)(\mathcal{I}_\xi(t_1), \ldots, \mathcal{I}_\xi(t_n))$.
- $\mathcal{I}_\xi \models (\neg\varphi)$ gdw nicht $\mathcal{I}_\xi \models \varphi$
- $\mathcal{I}_\xi \models (\varphi \wedge \psi)$ gdw $\mathcal{I}_\xi \models \varphi$ und $\mathcal{I}_\xi \models \psi$ (entsprechend für die anderen Junktoren)
- $\mathcal{I}_\xi \models (\forall x^\tau.\, \varphi)$ gdw für alle $a \in \mathcal{U}_\tau$ gilt $\mathcal{I}_{\xi[x \leftarrow a]} \models \varphi$
- $\mathcal{I}_\xi \models (\exists x^\tau.\, \varphi)$ gdw es gibt ein $a \in \mathcal{U}_\tau$ mit $\mathcal{I}_{\xi[x \leftarrow a]} \models \varphi$
- Hat man eine Logik mit Gleichheit, so wird zusätzlich definiert: $\mathcal{I}_\xi \models (t_1^\tau \doteq t_2^\tau)$ gdw $\mathcal{I}_\xi(t_1) = \mathcal{I}_\xi(t_2)$ in $\mathcal{U}_\tau$. ■

Die so angegebene schwache Semantik unterscheidet sich von der Standardsemantik dadurch, daß die Funktionsuniversen $\mathcal{U}_{(\tau_1\times\cdots\times\tau_m\rightarrow\sigma)}$ nicht *alle* entsprechenden Funktionen enthalten müssen, sondern nur eine nichtleere *Teilmenge* aller Funktionen. Es gilt also statt $\mathcal{U}_{(\tau_1\times\cdots\times\tau_m\rightarrow\sigma)} = \mathcal{F}(\mathcal{U}_{\tau_1},\ldots,\mathcal{U}_{\tau_m};\mathcal{U}_\sigma)$ nur $\mathcal{U}_{(\tau_1\times\cdots\times\tau_m\rightarrow\sigma)} \subseteq \mathcal{F}(\mathcal{U}_{\tau_1},\ldots,\mathcal{U}_{\tau_m};\mathcal{U}_\sigma)$. Wir schreiben $\langle\{\mathcal{U}_\tau\}_\tau,\mathcal{I}\rangle \models \Gamma$, wenn $\langle\{\mathcal{U}_\tau\}_\tau,\mathcal{I}\rangle$ ein schwaches Modell von Γ ist (das heißt die Funktionsuniversen müssen nur die Beziehung mit „$\subseteq$" erfüllen) und $\langle\{\mathcal{U}_\tau\}_\tau,\mathcal{I}\rangle \models\!\!\equiv \Gamma$, wenn $\langle\{\mathcal{U}_\tau\}_\tau,\mathcal{I}\rangle$ ein Standardmodell von Γ ist (das heißt die Funktionsuniversen müssen nur die Beziehung mit „$=$" erfüllen).
Allerdings ist die so definierte schwache Semantik oft zu schwach, wie wir oben gesehen haben. Deshalb nimmt man eine a priori unendlich Menge von Komprehensionsaxiomen Υ an, die in der Semantik erfüllt sein müssen, das heißt man hat für ein Modell $\langle\{\mathcal{U}_\tau\}_\tau,\mathcal{I}\rangle \models \Gamma \cup \Upsilon$. Diese Semantik bezeichnen wir als stark. Das heißt wir haben drei verschiedene Semantiken: die Standardsemantik, die starke Semantik und die schwache Semantik. Entsprechend ergibt sich die Folgerungbeziehung: Sei Γ eine Formelmenge und φ eine Formel, dann gilt:

- $\Gamma \models\!\!\equiv \varphi$, wenn jedes Standardmodell von Γ auch ein Modell von φ ist.
- $\Gamma \cup \Upsilon \models \varphi$, wenn jedes starke Modell von Γ auch ein Modell von φ ist.
- $\Gamma \models \varphi$, wenn jedes schwache Modell von Γ auch ein Modell von φ ist.

Es gilt dann die einfache Beziehung:

$$\text{Wenn } \Gamma \models \varphi \text{ dann } \Gamma \cup \Upsilon \models \varphi \text{ und wenn } \Gamma \cup \Upsilon \models \varphi \text{ dann } \Gamma \models\!\!\equiv \varphi.$$

Wie sehen die Komprehensionsaxiome Υ nun aus?

4.5 DEFINITION (KOMPREHENSIONSAXIOME): Die *Komprehensionsaxiome* Υ bestehen aus den folgenden Formeln:

Υ^f Für jeden Term t des Typs $\tau \neq o$, so daß die freien Variablen von t höchstens die verschiedenen Variablen $x_1,\ldots,x_m,y_1,\ldots,y_k$ vom Typ $\tau_1,\ldots,\tau_m,\sigma_1,\ldots,\sigma_k$ sind, die Formel:
$\forall y_1\ldots\forall y_k.\, \exists f^{(\tau_1\times\cdots\times\tau_m\rightarrow\tau)}.\, \forall x_1\ldots\forall x_m.\, (f(x_1,\ldots,x_m) \doteq t)$.

Υ^p Für jede Formel φ, so daß die freien Variablen von φ höchstens die verschiedenen Variablen $x_1,\ldots,x_m,y_1,\ldots,y_k$ vom Typ $\tau_1,\ldots,\tau_m,\sigma_1,\ldots,\sigma_k$ sind, die Formel:
$\forall y_1\ldots\forall y_k.\, \exists p^{(\tau_1\times\cdots\times\tau_m\rightarrow o)}.\, \forall x_1\ldots\forall x_m.\, (p(x_1,\ldots,x_m) \Leftrightarrow \varphi)$. ■

Anschaulich gesprochen besagen die Komprehensionsaxiome also, daß für jeden Term (jede Formel) eine Variable gibt, die diesen Term zusammenfaßt. Im Beispiel 4.3 bedeutet dies für die Formel $\textit{trinkt\_kaffee}(x) \Rightarrow \textit{schläft\_schlecht}(x)$ gibt es ein einstelliges Prädikatssymbol Q mit $\forall x.\, Q(x) \Leftrightarrow \textit{trinkt\_kaffee}(x) \Rightarrow \textit{schläft\_schlecht}(x)$. Verwendet man den λ-Kalkül so ist die Existenz dieses Prädikatssymbols natürlich direkt gesichert, es ist nämlich dann gerade $Q = \lambda(x).\, \textit{trinkt\_kaffee}(x) \Rightarrow \textit{schläft\_schlecht}(x)$. Folglich sind im λ-Kalkül keine Komprehensionsaxiome erforderlich.

Wir wollen nun eine Übersetzung definieren, die Probleme der Logik höherer Stufe in Probleme einer flachsortierten, einfachen Logik erster Stufe übersetzen.

4.6 Definition (Übersetzung Θ): Sei Γ eine Formelmenge der Logik n-ter Stufe. Wir definieren die Übersetzung Θ als Funktor, der

– der Signatur Σ einer Logik höherer Stufe die Signatur

$$\Theta(\Sigma) = (\overline{\Sigma'}, \mathbf{SORT}, \mathfrak{s}, \mathbf{TD}, \mathbf{SD}, \mathbf{PD})$$

einer sortierten Logik erster Stufe mit Gleichheit zuordnet mit

$$\overline{\Sigma'} = \{\Theta(c) \,|\, c \in \Sigma\} \cup \{\alpha^\tau \,|\, \tau \text{ kommt in } \Sigma \text{ vor}\} \qquad \alpha \text{ neu}$$

wobei $\Theta(c)$ ein Konstantensymbol ist, so daß Θ injektiv ist und für jede Konstante c vom Typ τ gilt $\Theta(c)$ ist eine Konstante der Sorte $\Theta(\tau)$, die wir der Einfachheit halber wieder mit c bezeichnen,

$$\mathbf{SORT} = \{\Theta(\tau) \,|\, \tau \text{ kommt in } \Sigma \text{ vor}\}$$

wobei $\Theta(\tau)$ ein Sortensymbol ist, so daß Θ injektiv ist. Statt $\Theta(\tau)$ schreiben wir kurz $\overline{\tau}$. Für Variablen gilt $\mathfrak{s}(\Theta(x_\tau)) = \overline{\tau}$. Die Termdeklarationen bestehen aus

$$\mathbf{TD} = \{c \lessdot \overline{\tau} \,|\, c \in \Sigma \text{ mit } \mathrm{Typ}(c) = \tau\} \cup \{\alpha^\tau \lessdot (\overline{\tau} \times \overline{\tau_1} \times \cdots \times \overline{\tau_m} \to \overline{\sigma})\}$$

wobei $\tau = (\tau_1 \times \cdots \times \tau_m \to \sigma)$ mit $\sigma \neq o$ ist. Die Prädikatsdeklarationen sind

$$\mathbf{PD} = \{P \lessdot \overline{\tau} \,|\, P \in \Sigma \text{ mit } \mathrm{Typ}(P) = \tau\} \cup \{\alpha^\tau \lessdot (\overline{\tau} \times \overline{\tau_1} \times \cdots \times \overline{\tau_m})\}$$

mit $\tau = (\tau_1 \times \cdots \times \tau_m \to o)$. Die Untersortenbeziehungen sind leer, das heißt $\mathbf{SD} = \emptyset$.

– Für Formelmengen Γ definieren wir
$\Theta(\Gamma) = \left\{\hat{\Theta}(\varphi) \,|\, \varphi \in \Gamma\right\} \cup \Xi$, wobei Ξ Extensionalitätsaxiome sind.

Dabei gilt:

* Für alle Variablen x_τ gilt $\hat{\Theta}(x_\tau) = x_{\overline{\tau}}$
* Für alle Konstanten c_τ gilt $\hat{\Theta}(c_\tau) = c_{\overline{\tau}}$
* Für Terme und atomare Formeln mit m-stelligem Topterm t vom Typ τ gilt: $\hat{\Theta}(t(t_1, \ldots, t_m)) = \alpha^\tau(\hat{\Theta}(t), \hat{\Theta}(t_1), \ldots, \hat{\Theta}(t_m))$
* Für durch Junktoren verbundene Formeln gilt $\hat{\Theta}(\varphi_1 * \varphi_2) = \hat{\Theta}(\varphi_1) * \hat{\Theta}(\varphi_2)$ mit $* \in \{\wedge, \vee, \Rightarrow, \Leftrightarrow\}$ und $\hat{\Theta}(\neg\varphi) = \neg\hat{\Theta}(\varphi)$.
* Für quantifizierte Formeln gilt $\hat{\Theta}(\forall x^\tau\!\centerdot\, \varphi) = \forall x_{\overline{\tau}}\!\centerdot\, \hat{\Theta}(\varphi)$

* Ξ ist die Menge aller Formeln mit $\tau = (\tau_1 \times \cdots \times \tau_m \to \sigma)$ und $\sigma \neq o$:
$\forall f^{\overline{\tau}}\!.\, \forall g^{\overline{\tau}}\!.\, (\forall x_1^{\overline{\tau_1}}\!., \ldots, \forall x_m^{\overline{\tau_m}}\!.$
$\alpha^\tau(f, x_1, \ldots, x_m) \doteq \alpha^\tau(g, x_1, \ldots, x_m)) \Rightarrow f \doteq g$
und mit $\tau = (\tau_1 \times \cdots \times \tau_m \to o)$:
$\forall p^{\overline{\tau}}\!.\, \forall q^{\overline{\tau}}\!.\, (\forall x_1^{\overline{\tau_1}}\!., \ldots, \forall x_m^{\overline{\tau_m}}\!.$
$\alpha^\tau(p, x_1, \ldots, x_m) \Leftrightarrow \alpha^\tau(q, x_1, \ldots, x_m)) \Rightarrow p \doteq q$ ■

4.7 THEOREM: Θ *ist korrekt bezüglich der schwachen Semantik, das heißt für jede Formelmenge* Γ *gilt wenn* Γ *erfüllbar ist, dann ist auch* $\Theta(\Gamma)$ *erfüllbar.*

Beweis: Sei $\mathcal{M} = \langle \{\mathcal{U}_\tau\}_\tau, \mathcal{I}\rangle$ ein schwaches Modell einer Formelmenge Γ. Wir konstruieren aus $\mathcal{M}$ ein Modell von $\Theta(\Gamma)$ mit $\hat{\mathcal{U}}_{\overline{\tau}} := \mathcal{U}_\tau$, $\hat{\mathcal{I}}$ wird definiert durch $\hat{\mathcal{I}}(\hat{\Theta}(c)) := \mathcal{I}(c)$ für alle Konstanten c in Σ. (Wegen der Injektivität von Θ ist diese Abbildung wohldefiniert.) Für Belegungen definieren wir analog $\hat{\xi}(\hat{\Theta}(x)) := \xi(x)$. Wegen $\hat{\mathcal{U}}^{\overline{\tau}} = \mathcal{U}_\tau$ erhalten wir so alle Belegungen, denn in der sortierten Logik gibt es ja keine Funktions oder Prädikatsvariablen. Für die Funktionen α^τ mit $\tau = (\tau_1 \times \cdots \times \tau_m \to \sigma)$ definieren wir Interpretationen, so daß sie die Interpretation des ersten Arguments hernehmen, was ja eine Funktion darstellt, und das Ergebnis auf die Interpretationen der restlichen Argumente anwendet. Wie können diese Funktionen so interpretieren, da sie neu sind. Formal heißt das: für alle $f \in \hat{\mathcal{U}}_{\overline{\tau}}$ gilt für alle $x_1 \in \hat{\mathcal{U}}_{\overline{\tau_1}}, \ldots, x_m \in \hat{\mathcal{U}}_{\overline{\tau_m}}$ $\hat{\mathcal{I}}_{\hat{\xi}}(\alpha_{\overline{\tau}})(f, x_1, \ldots, x_m) := f(x_1, \ldots, x_m)$. Beachte, daß $f \in \hat{\mathcal{U}}_{\overline{\tau}} = \mathcal{U}_\tau$ also eine Funktion ist und somit anwendbar ist. Analog definieren wir die Interpretation für Prädikate. Wie im Fall der Relativierung können wir durch vollständige Induktion über den Term- und Formelaufbau zeigen, daß gilt $\hat{\mathcal{I}}_{\hat{\xi}} \circ \hat{\Theta} = \mathcal{I}_\xi$.

Ist also $\mathcal{M}$ schwaches Modell für eine Formel φ aus Γ, das heißt $\mathcal{I}_\xi(\varphi) = \texttt{wahr}$ für jede Belegung ξ, so gilt $\hat{\mathcal{I}}_{\hat{\xi}}(\hat{\Theta}(\varphi)) = \texttt{wahr}$, wobei wir alle Belegungen erhalten, also gilt $\hat{\mathcal{I}}$ ist Modell von $\hat{\Theta}(\varphi)$.

Es bleibt zu zeigen, daß die Extensionalitätseigenschaften gelten, formal:

$$\hat{\mathcal{I}}_{\hat{\xi}} \begin{pmatrix} \forall f_{\overline{\tau}} \forall g_{\overline{\tau}} (\forall x^1_{\overline{\tau_1}}, \ldots, \forall x^m_{\overline{\tau_m}} \\ \alpha^\tau(f, x_1, \ldots, x_m) \doteq \alpha^\tau(g, x_1, \ldots, x_m)) \Rightarrow f \doteq g \end{pmatrix} = \texttt{wahr}.$$

Dazu ist es nötig zu zeigen, daß für alle F, G in $\hat{\mathcal{U}}_{\overline{\tau}}$ gilt:
wenn für alle $X_1 \in \hat{\mathcal{U}}_{\overline{\tau}_1}, \ldots, X_m \in \hat{\mathcal{U}}_{\overline{\tau}_m}$, $\hat{\mathcal{I}}_{\hat{\xi}[f,g,x_1,\ldots,x_m \leftarrow F,G,X_1,\ldots,X_m]}(\alpha^\tau(f, x_1, \ldots, x_m)) = \hat{\mathcal{I}}_{\hat{\xi}[f,g,x_1,\ldots,x_m \leftarrow F,G,X_1,\ldots,X_m]}(\alpha^\tau(g, x_1, \ldots, x_m))$, dann $\hat{\mathcal{I}}_{\hat{\xi}[f,g \leftarrow F,G]}(f) = \hat{\mathcal{I}}_{\hat{\xi}[f,g \leftarrow F,G]}(g)$, das heißt, $F = G$.

Nach Definition gilt aber $\hat{\mathcal{I}}_{\hat{\xi}[f,g,x_1,\ldots,x_m \leftarrow F,G,X_1,\ldots,X_m]}$ und die Interpretation von α^τ ist die Vorbedingung äquivalent zu $F(X_1, \ldots, X_m) = G(X_1, \ldots, X_m)$. Da zwei Funktionen gleich sind, wenn sie in allen Werten übereinstimmen, ergibt sich $F = G$.
Analoges gilt für die Axiome die Prädikate betreffen. ■

4.8 THEOREM: Θ *ist im schwachen Sinne vollständig, das heißt für jede Formelmenge* Γ *gilt, wenn* $\Theta(\Gamma)$ *erfüllbar ist, dann ist auch* Γ *erfüllbar.*

Beweis: Der Beweis läuft so ähnlich wie im obigen Fall, allerdings ist die Konstruktion der Universen in diesem Fall etwas komplizierter. Sei also $\mathcal{M}$ ein beliebiges Modell

von $\Theta(\Gamma)$. Wir definieren nun induktiv die Universen über $\check{\mathcal{U}}_\iota := \mathcal{U}_{\overline{\iota}}$ und $\check{\mathcal{U}}_o := \mathcal{U}_{\overline{o}}$. Für zusammengesetzte Typen $\tau = (\tau_1 \times \cdots \times \tau_m \to \sigma)$ wird $\check{\mathcal{U}}_\tau$ als Teilmenge von $\mathcal{F}(\check{\mathcal{U}}_{\tau_1}, \ldots, \check{\mathcal{U}}_{\tau_m}; \check{\mathcal{U}}_\sigma)$ definiert. Dazu definieren wir induktive injektive Funktionen $\natural_\tau$ von $\mathcal{U}_{\overline{\tau}}$ nach $\mathcal{F}(\check{\mathcal{U}}_{\tau_1}, \ldots, \check{\mathcal{U}}_{\tau_m}; \check{\mathcal{U}}_\sigma)$ und setzen $\check{\mathcal{U}}_\tau := \natural_\tau(\mathcal{U}_{\overline{\tau}})$. Also ist $\natural_\tau$ eine bijektive Funktion von $\mathcal{U}_{\overline{\tau}}$ nach $\check{\mathcal{U}}_\tau$.[2]
Wir definieren $\natural_\tau$ als bijektive Funktionen induktiv:

1. $\natural_\iota : \mathcal{U}_{\overline{\iota}} \to \check{\mathcal{U}}_\iota$ and $\natural_o : \mathcal{U}_{\overline{o}} \to \check{\mathcal{U}}_o$ als die Indentität, also sind beide Funktionen trivialerweise bijektiv.

2. Seien $\natural_{\tau_i}$ und $\natural_\sigma$ definiert für $\mathcal{U}_{\overline{\tau}_1}, \ldots, \mathcal{U}_{\overline{\tau}_m}$ und $\mathcal{U}_{\overline{\sigma}}$. Daraus definieren wir eine Funktion $\natural_\tau$ mit $\tau = (\tau_1 \times \cdots \times \tau_m \to \sigma)$, $\sigma \neq o$, für $\mathcal{U}_{\overline{\tau}}$. Für alle $x \in \mathcal{U}_{\overline{\tau}}$ wird $\natural_\tau(x)$ definiert als $\natural_\tau(x)(\check{x}_1, \ldots, \check{x}_m) := \natural_\sigma(\mathcal{I}_\xi(\alpha^\tau)(x, \natural_{\tau_1}^{-1}(\check{x}_1), \ldots, \natural_{\tau_m}^{-1}(\check{x}_m)))$ für alle $\check{x}_1 \in \check{\mathcal{U}}_{\tau_1}, \ldots, \check{x}_m \in \check{\mathcal{U}}_{\tau_m}$

   Das folgende Diagramm zeigt die beteiligten Abbildungen auf einen Blick:

$$\begin{tikzcd}
\mathcal{I}_\xi(\alpha^\tau) : \mathcal{U}_{\overline{\tau}} \arrow[d, "!"', "\natural_\tau"] & \times & \mathcal{U}_{\overline{\tau}_1} & \times \cdots \times & \mathcal{U}_{\overline{\tau}_m} \arrow[r] & \mathcal{U}_{\overline{\sigma}} \arrow[d, "\natural_\sigma"] \\
\check{\mathcal{U}}_\tau \arrow[r, hook] & \mathcal{F}(& \check{\mathcal{U}}_{\tau_1} \arrow[u, "\natural_{\tau_1}^{-1}"] & , \ldots, & \check{\mathcal{U}}_{\tau_m} \arrow[u, "\natural_{\tau_m}^{-1}"] \ ; & \check{\mathcal{U}}_\sigma)
\end{tikzcd}$$

   Um die Injektivität von $\natural_\tau$ zu zeigen benutzen wir das Extensionalitätsaxiom

$$\forall f_{\overline{\tau}}.\ \forall g_{\overline{\tau}}.\ (\forall x^1_{\overline{\tau}_1}.\ \ldots \forall x^m_{\overline{\tau}_m}.\ \alpha^\tau(f, x^1, \ldots, x^m) \doteq \alpha^\tau(g, x^1, \ldots, x^m)) \Rightarrow f \doteq g$$

   Also gilt in einem Modell für alle x, x' in $\mathcal{U}_{\overline{\tau}}$

$$\text{für alle } y_1 \in \mathcal{U}_{\overline{\tau}_1}, \ldots, y_m \in \mathcal{U}_{\overline{\tau}_m} \text{ wenn } \mathcal{I}_\xi(\alpha^\tau)(x, y_1, \ldots, y_m) = \mathcal{I}_\xi(\alpha^\tau)(x', y_1, \ldots, y_m) \text{ dann } x = x' \qquad (*)$$

   Gelten $\natural_\tau(x) = \natural_\tau(x')$ für beliebige x und x' in $\mathcal{U}_{\overline{\tau}}$. Dann haben wir per Definition für alle $\check{x}_1 \in \check{\mathcal{U}}_{\tau_1}, \ldots, \check{x}_m \in \check{\mathcal{U}}_{\tau_m}$
   $\natural_\sigma \mathcal{I}_\xi(\alpha^\tau)(x, \natural_{\tau_1}^{-1}(\check{x}_1), \ldots, \natural_{\tau_m}^{-1}(\check{x}_m)) = \natural_\sigma \mathcal{I}_\xi(\alpha^\tau)(x', \natural_{\tau_1}^{-1}(\check{x}_1), \ldots, \natural_{\tau_m}^{-1}(\check{x}_m))$. Da die Abbildungen $\natural_{\tau_1}, \ldots, \natural_{\tau_m}, \natural_\sigma$ alle bijektiv sind, ergibt sich für alle $y_1 \in \mathcal{U}_{\overline{\tau}_1}, \ldots,$ $y_m \in \mathcal{U}_{\overline{\tau}_m}$: $\mathcal{I}_\xi(\alpha^\tau)(x, y_1, \ldots, y_m) = \mathcal{I}_\xi(\alpha^\tau)(x', y_1, \ldots, y_m)$. Wegen der Beziehung $(*)$ folgt $x = x'$, also ist die Injektivität gezeigt. Die Surjektivität ergibt sich per definitionem, also ist die Bijektivität gezeigt.

   Die Argumentation im Falle $\sigma = o$ ist analog.

Also haben wir die Universen für alle Typen τ definiert. Wir bezeichnen im folgenden mit $\natural$ die polymorphe Abbildung, die durch all die individuellen $\natural_\tau$ gebildet wird. Nun wird eine Interpretationsfunktion $\check{\mathcal{I}}$ definiert mit $\check{\mathcal{I}}(c) := \natural \circ \mathcal{I} \circ \hat{\Theta}(c)$ für alle Konstanten c.

[2] Bijektivität von $\mathcal{U}_{\overline{\tau}}$ nach $\mathcal{F}(\check{\mathcal{U}}_{\tau_1}, \ldots, \check{\mathcal{U}}_{\tau_m}; \check{\mathcal{U}}_\sigma)$ ist im allgemeinen nicht erreichbar, da dies Vollständigkeit bezüglich von Standardmodellen bedeuten würde.

Wiederum per Induktion über den Term- und Formelaufbau zeigt man $\check{\mathcal{I}}_{\check{\xi}}(t) = \natural \circ \mathcal{I}_\xi \circ \hat{\Theta}(t)$.
Zum Beispiel ergibt sich für die universelle Quantifikation:
$\check{\mathcal{I}}_{\check{\xi}}(\forall x_\tau \varphi)$ gdw für alle d in $\check{\mathcal{U}}_\tau$ gilt $\check{\mathcal{I}}_{\check{\xi}[x \leftarrow d]}(\varphi)$ gdw (nach Ind.hyp) für alle d in $\check{\mathcal{U}}_\tau$ gilt $\natural \mathcal{I}_{\xi[\hat{\Theta}(x) \leftarrow \natural^{-1}(d)]} \hat{\Theta}(\varphi)$ gdw für alle d in $\mathcal{U}_{\bar{\tau}}$ $\natural \mathcal{I}_{\xi[\hat{\Theta}(x) \leftarrow d]} \hat{\Theta}(\varphi)$ gdw $\natural \mathcal{I}_\xi \hat{\Theta}(\forall x \varphi)$.
Wir benutzen $\check{\xi} = \natural \circ \xi \circ \hat{\Theta}$ und $\check{\xi}[x_\tau \leftarrow d] = \natural \circ \xi[\hat{\Theta}(x_\tau) \leftarrow \natural^{-1}(d)] \circ \hat{\Theta}$ für alle Belegungen.
Nun zeigen wir, wenn $\mathcal{M}$ ein Modell von $\Theta(\Gamma)$ ist, dann ist $\check{M}$ ein Modell von Γ. Wenn $\mathcal{M}$ ein Modell von $\Theta(\Gamma)$ ist, dann ist $\mathcal{M}$ Modell von $\hat{\Theta}(\varphi)$ für alle $\varphi \in \Gamma$. Sei $\check{\xi}$ eine beliebige Belegung und ξ so definiert wie oben, dann haben wir $\mathcal{I}_\xi(\hat{\Theta}(\varphi)) = \texttt{wahr}$, da $\mathcal{M}$ Modell von $\hat{\Theta}(\varphi)$ ist. Also gilt $\check{\mathcal{I}}_{\check{\xi}}(\varphi) = \natural(\mathcal{I}_\xi(\hat{\Theta}(\varphi))) = \texttt{wahr}$, da $\natural$ auf den Wahrheitswerten die Identität ist. ■

Die vorgestellte Übersetzung respektiert die schwache Semantik, will man jedoch eine starke Semantik zugrunde legen, so muß man die a priori unendlich vielen Komprehensionsaxiome mitübersetzen.

4.9 Beispiel: Als Beispiel betrachten wir das Theorem, daß die Komposition von binären Relationen assoziativ ist. Formal betrachten wir eine Formelmenge Γ, die die Definition der Komposition und das negierte Theorem enthält, also (mit der Abkürzung $\tau = (\iota \times \iota \to o)$)

$$\begin{aligned}\Gamma = \{ & \forall \rho^\tau.\, \forall \sigma^\tau.\, \forall x^\iota.\, \forall y^\iota.\, (\rho \circ \sigma)(x,y) \Leftrightarrow (\exists z^\iota.\, \rho(x,z) \wedge \sigma(z,y)) \\ & \neg \forall \rho^\tau.\, \forall \sigma^\tau.\, \forall \mu^\tau.\, (\rho \circ \sigma) \circ \mu = \rho \circ (\sigma \circ \mu)\}\end{aligned}$$

Die Übersetzung lautet:

$$\begin{aligned}\Theta(\Gamma) = \{ & \forall \rho^{\bar{\tau}}.\, \forall \sigma^{\bar{\tau}}.\, \forall x^{\bar{\iota}}.\, \forall y^{\bar{\iota}}.\, \alpha^\tau(\alpha^{(\tau \times \tau \to \tau)}(\circ, \rho, \sigma), x, y) \Leftrightarrow (\exists z^{\bar{\iota}}.\, \alpha^\tau(\rho, x, z) \wedge \alpha^\tau(\sigma, z, y)) \\ & \neg \forall \rho^{\bar{\tau}}.\, \forall \sigma^{\bar{\tau}}.\, \forall \mu^{\bar{\tau}}.\, \alpha^{(\tau \times \tau \to \tau)}(\circ, \alpha^{(\tau \times \tau \to \tau)}(\circ, \rho, \sigma), \mu) = \\ & \qquad \alpha^{(\tau \times \tau \to \tau)}(\circ, \rho, \alpha^{(\tau \times \tau \to \tau)}(\circ, \sigma, \mu))\} \cup \Xi\end{aligned}$$

Dabei wird für den Beweis des Theorems aus Ξ das folgende Extensionalitätsaxiom benötigt: $\forall \rho^{\bar{\tau}}.\, \forall \sigma^{\bar{\tau}}.\, (\forall x^{\bar{\iota}}.\, \forall y^{\bar{\iota}}.\, \alpha^\tau(\rho, x, y) \Leftrightarrow \alpha^\tau(\sigma, x, y)) \Rightarrow \rho \doteq \sigma$.
Diese Formelmenge kann dann mit der Termrelativierung in unsortierte Logik erster Stufe übersetzt werden, so daß mit einem Resolutionsbeweis ein Widerspruch abgeleitet werden kann. ■

Kapitel 5

Sortierte Unifikation

Wesentlicher Bestandteil fortgeschrittener automatischer Beweiser ist die Unifikation, so spielt sie auch bei der Operationalisierung von Sortenlogiken eine herausragende Rolle. Wir werden daher in diesem Kapitel einen Unifikationsalgorithmus angeben, der wie die grundlegenden Begriffe der Sortenlogiken von Manfred Schmidt-Schauß [SS89] übernommen ist.
Das Ziel der Unifikation ist es, *Unifikationsprobleme* Γ zu lösen, das heißt zu Mengen $\{s_1 = t_1, \ldots, s_m = t_m\}$ eine Substitution σ (vergleiche Definition 2.6) zu finden, so daß $\sigma(s_1) = \sigma(t_1), \ldots, \sigma(s_m) = \sigma(t_m)$ gilt. Eine solche Substitution heißt *Unifikator*. Ist man gelegentlich daran interessiert nur festzustellen, *ob* zwei Terme unifizierbar sind, so will man meistens alle Unifikatoren repräsentieren. Dies funktioniert im unsortierten Fall (ebenso wie in eingeschränkten Fällen sortierter Logik) durch die Angabe eines einzigen *allgemeinsten Unifikators*, das heißt eines Unifikators, aus dem sich jeder andere durch Spezialisierung ergibt. Formal: Eine Substitution σ heißt allgemeinster Unifikator von s und t, wenn σ ein Unifikator ist und jeder andere Unifikator τ sich darstellen läßt als $\tau = \tau' \circ \sigma$, wobei τ' eine Substitution ist.
Eine *vollständige Menge* $\mathbf{U}_\Sigma(\Gamma)$ *von Unifikatoren* ist eine Teilmenge aller Substitutionen bezüglich Σ, so daß gilt:

Korrektheit: jedes Element auf $\mathbf{U}_\Sigma(\Gamma)$ ist ein Unifikator von Γ,

Vollständigkeit: zu jedem Unifikator τ von Γ gibt es ein Element $\sigma \in \mathbf{U}_\Sigma(\Gamma)$ und eine Substitution τ', so daß $\tau = \tau' \circ \sigma$.

Eine vollständige Menge $\mathbf{U}_\Sigma(\Gamma)$ von Unifikatoren heißt *minimal*, wenn gilt:

Minimalität: Für je zwei Unifikatoren σ und τ aus $\mathbf{U}_\Sigma(\Gamma)$ gilt, gibt es eine Unifikation τ' mit $\tau = \tau' \circ \sigma$, so gilt $\sigma = \tau$.

Die Menge aller minimalen Unifikatoren bezeichnen wir mit $\mathbf{MU}(\Gamma)$.
Man sagt dann, die Unifikation ist *unitär*, wenn für beliebige Unifikationsprobleme Γ die Menge $\mathbf{MU}(\Gamma)$ existiert und ihre Kardinalität kleiner gleich 1 ist (Unifikationstyp 1), *finitär*, wenn für beliebige Unifikationsprobleme Γ die Menge $\mathbf{MU}(\Gamma)$

existiert und ihre Kardinalität endlich ist (Unifikationstyp ω), *infinitär*, wenn für beliebige Unifikationsprobleme Γ die Menge $\mathbf{MU}(\Gamma)$ existiert und ihre Kardinalität für mindestens ein Unifikationsproblem unendlich ist (Unifikationstyp ∞), *nullär*, wenn es ein Unifikationsproblem Γ gibt, so daß $\mathbf{MU}(\Gamma)$ nicht existiert (Unifikationstyp 0).

5.1 DEFINITION: Wir sagen, ein Unifikationsproblem Γ ist *gelöst*, wenn

- es die Gestalt $\{x^1_{S_1} = t^1, \ldots, x^n_{S_n} = s^n\}$ hat,
- die x^i sind verschieden und kommen auf der rechten Seite der Gleichungen nicht vor.
- $\mathfrak{s}(x^i) \in \mathbf{SORT}_\Sigma(t^i)$, das heißt die minimale Sorte von x^i ist eine Sorte von t^i.

In diesem Fall erhält man aus Γ eine Substitution $\sigma_\Gamma = [x_1 \leftarrow t_1, \ldots, x_n \leftarrow t_n]$. Mit $\perp$ bezeichnen wir das unlösbare Unifikationsproblem. ■

Wir wollen uns nun einen nichtdeterministischen Unifikationsalgorithmus für die allgemeine sortierte Logik aus Kapitel 2 ansehen. Solche Algorithmen lassen sich besonders übersichtlich durch Regeln darstellen, die Unifikationsprobleme in Unifikationsprobleme überführen. Dabei steht Γ für ein Unifikationsproblem und $x = y, \Gamma$ steht für das Unifikationsproblem, das aus Γ und dem Problem $x = y$ besteht, also $\Gamma \cup \{x = y\}$. Für die Transformationsregeln definiert man:

5.2 DEFINITION (KORREKTHEIT UND VOLLSTÄNDIGKEIT VON TRANSFORMATIONEN): Wir nennen eine Transformation $\frac{\Gamma_1}{\Gamma_2}$ *korrekt*, wenn gilt $\mathbf{U}_\Sigma(\Gamma_1) \supseteq \mathbf{U}_\Sigma(\Gamma_2)$. Wir nennen eine Transformation $\frac{\Gamma_1}{\Gamma_2}$ *vollständig*, wenn gilt $\mathbf{U}_\Sigma(\Gamma_1) \subseteq \mathbf{U}_\Sigma(\Gamma_2)$.[1] ■

5.3 DEFINITION (NICHTDETERMINISTISCHER UNIFIKATIONSALGORITHMUS FÜR SORTIERTE LOGIK):

VV1 $\dfrac{x = x, \Gamma}{\Gamma}$

VV2 $\dfrac{x = y, \Gamma}{y = x, \Gamma}$ wenn $\mathfrak{s}(x) \sqsubseteq \mathfrak{s}(y)$

VV3 $\dfrac{x = y, \Gamma}{x = z, y = z, \Gamma}$

wenn $\mathfrak{s}(x) \not\sqsubseteq \mathfrak{s}(y)$ und $\mathfrak{s}(y) \not\sqsubseteq \mathfrak{s}(x)$ und S ist maximale Untersorte von $\mathfrak{s}(x)$ und $\mathfrak{s}(y)$ und z ist neue Variable mit $\mathfrak{s}(z) = S$.

VV4 $\dfrac{x = y, \Gamma}{x = f(s_1, \ldots, s_n), y = f(t_1, \ldots, t_n), s_1 = t_1, \ldots, s_n = t_n, \Gamma}$

wenn $\mathfrak{s}(x) \not\sqsubseteq \mathfrak{s}(y)$ und $\mathfrak{s}(y) \not\sqsubseteq \mathfrak{s}(x)$ und $f(s_1, \ldots, s_n) \lessdot S$ ist eine Termdeklaration mit $S \sqsubseteq \mathfrak{s}(x)$ und $f(t_1, \ldots, t_n) \lessdot R$ ist eine Termdeklaration mit $R \sqsubseteq \mathfrak{s}(y)$.

[1] Für die letzte Beziehung ist an sich nötig, über die verwendeten Variablen genau Buch zu führen. Einzelheiten findet man in [SS89, S.46].

VV5 $\dfrac{x = y, \Gamma}{\perp}$

wenn $\mathfrak{s}(x) \not\sqsubseteq \mathfrak{s}(y)$ und $\mathfrak{s}(y) \not\sqsubseteq \mathfrak{s}(x)$ und es gibt keine gemeinsame Untersorte von $\mathfrak{s}(x)$ und $\mathfrak{s}(y)$ und es gibt keine Termdeklarationen $f(s_1, \ldots, s_n) \lessdot S$ mit $S \sqsubseteq \mathfrak{s}(x)$ und $f(t_1, \ldots, t_n) \lessdot R$ mit $R \sqsubseteq \mathfrak{s}(x)$.

VT1 $\dfrac{x = t, \Gamma}{x = t, \Gamma[x \leftarrow t]}$

wenn x nicht in t, aber in Γ vorkommt und wenn $\mathfrak{s}(x) \in \mathbf{SORT}_\Sigma(t)$

VT2 $\dfrac{t = x, \Gamma}{x = t, \Gamma}$

wenn t keine Variable ist.

VT3 $\dfrac{x = f(t_1, \ldots, t_n), \Gamma}{x = f(s_1, \ldots, s_n), s_1 = t_1, \ldots, s_n = t_n, \Gamma}$

wenn x keine Variable in $f(t_1, \ldots, t_n)$ ist, $\mathfrak{s}(x)$ keine Sorte von $f(t_1, \ldots, t_n)$ ist und $f(s_1, \ldots, s_n) \lessdot S$ eine Termdeklaration ist mit $S \sqsubseteq \mathfrak{s}(x)$.

VT4 $\dfrac{x = f(t_1, \ldots, t_n), \Gamma}{\perp}$

wenn x keine Variable in $f(t_1, \ldots, t_n)$ ist, $\mathfrak{s}(x)$ keine Sorte von $f(t_1, \ldots, t_n)$ ist und es keine Termdeklaration $f(s_1, \ldots, s_n) \lessdot S$ mit $S \sqsubseteq \mathfrak{s}(x)$ gibt.

VT5 $\dfrac{x = t, \Gamma}{\perp}$

wenn x in t vorkommt.

TT1 $\dfrac{f(s_1, \ldots, s_n) = f(t_1, \ldots, t_n)}{s_1 = t_1, \ldots, s_n = t_n}$

TT2 $\dfrac{s = t}{\perp}$

wenn die Kopfsymbole von s und t nicht übereinstimmen.

Dabei muß jede Termdeklaration $t \lessdot S$ aus der Signatur Σ vollständig mit neuen Variablen umbenannt werden, bevor sie in einem Unifikationsschritt benutzt werden darf. ■

5.4 Bemerkung: Mit der Regel **VV1** werden triviale Beziehungen aus dem Unifikationsproblem gelöscht. Mit **VV2** richtet man Variablen bezüglich der Untersortenbeziehung. Ist dies nicht möglich, kann man mit **VV3** versuchen eine gemeinsame Untersorte für beide Variablen zu finden, ist auch dies nicht möglich, helfen vielleicht

Termdeklarationen weiter, die beiden Variablen gleichzumachen. Dazu dient die Regel **VV4**. Ist alles dies nicht möglich, ist ein Ausdruck $x = y$ nicht unifizierbar, was gerade **VV5** besagt. Damit sind alle Fälle behandelt, mit denen versucht werden kann, zwei Variablen zu unifizieren.
Die Regeln **VT1** bis **VT5** behandeln die Unifikation von Variablen mit nicht-variablen Termen, dabei wird unter den Voraussetungen der nächsten Definition davon ausgegangen, daß ein Ausdruck $x = t$ schon gelöst ist. Mit **VT1** kann eine solche Lösung auf den Rest des Unifikationsproblems angewendet werden. **VT2** dient dem Richten. **VT3** dient dem Abschwächen eines Problems mit einer Termdeklaration. **VT5** ist die übliche „occurs-check"-Regel und **VT4** deckt die restlichen Fälle ab, bei denen keine Unifikation möglich ist, also wenn der „occurs-check" nicht zuschlägt, aber es auch keine sortengerechte Substitution gibt, weder über Untersortenbeziehungen noch über Termdeklarationen.
Die Regeln **TT1** und **TT2** behandeln die Unifikation zweier nicht-variabler Terme. **TT1** ist die übliche Dekomposition, sie besagt, daß die Unifikation zweier nicht-variabler Terme auf die Unifikation aller Argumente des Kopfsymbols reduziert werden kann, wenn die Terme im Kopfsymbol übereinstimmen. Stimmen die beiden Terme nicht im führenden Funktionssymbol überein, sind die Terme nicht unifizierbar, was gerade **TT2** ausdrückt. ■

5.5 Beispiel: Sei eine Signatur gegeben mit
SORT $= \{\mathsf{gerade}, \mathbb{N}\}$, **SD** $= \{\mathsf{gerade} \sqsubseteq \mathbb{N}\}$
TD $= \{0 {<} \mathsf{gerade}, s {<} \mathbb{N} \to \mathbb{N}, s(s(x_{\mathsf{gerade}})) {<} \mathsf{gerade},$
$+ {<} \mathbb{N} \times \mathbb{N} \to \mathbb{N}, + {<} \mathsf{gerade} \times \mathsf{gerade} \to \mathsf{gerade}, y_{\mathbb{N}} + y_{\mathbb{N}} {<} \mathsf{gerade}\}$
Wir betrachten zu dieser Signatur verschiedene Unifikationsproblem:

- $\Gamma = \{s(s(x_{\mathbb{N}})) = +(y_{\mathbb{N}}, s(0))\} \overset{\mathbf{TT2}}{\leadsto}$ zu $\perp$.
- $\Gamma = \{+(x_{\mathsf{gerade}}, 0) = y_{\mathsf{gerade}}\} \overset{\mathbf{VT2}}{\leadsto} \{y_{\mathsf{gerade}} = +(x_{\mathsf{gerade}}, 0)\}$, was in gelöster Form ist.
- $\Gamma = \{y_{\mathsf{gerade}} = +(x_{\mathsf{gerade}}, 0), s(y_{\mathsf{gerade}}) = s(z_{\mathbb{N}})\} \overset{\mathbf{VT1}}{\leadsto}$
  $\{y_{\mathsf{gerade}} = +(x_{\mathsf{gerade}}, 0), s(+(x_{\mathsf{gerade}}, 0)) = s(z_{\mathbb{N}})\} \overset{\mathbf{TT1}}{\leadsto}$
  $\{y_{\mathsf{gerade}} = +(x_{\mathsf{gerade}}, 0), +(x_{\mathsf{gerade}}, 0) = z_{\mathbb{N}}\} \overset{\mathbf{VT2}}{\leadsto}$
  $\{y_{\mathsf{gerade}} = +(x_{\mathsf{gerade}}, 0), z_{\mathbb{N}} = +(x_{\mathsf{gerade}}, 0)\}$, was in gelöster Form ist.
- $\Gamma = \{x_{\mathsf{gerade}} = s(y_{\mathbb{N}})\} \overset{\mathbf{VT3}}{\leadsto}$ (da $s(s(x_{\mathsf{gerade}})) {<} \mathsf{gerade}$)
  $\{x_{\mathsf{gerade}} = s(s(z_{\mathsf{gerade}})), s(z_{\mathsf{gerade}}) = y_{\mathbb{N}}\} \overset{\mathbf{VT2}}{\leadsto}$
  $\{x_{\mathsf{gerade}} = s(s(z_{\mathsf{gerade}})), y_{\mathbb{N}} = s(z_{\mathsf{gerade}})\}$, was in gelöster Form ist. ■

5.6 BEMERKUNG: Um wenigstens einige der neu eingeführten Hilfvariablen (das heißt Variablen, die im ursprünglichen System nicht vorkommen) wieder loszuwerden kann man eine zusätzliche Regel einführen, nämlich

EH $\dfrac{x = t, \Gamma}{\Gamma}$

wenn x eine Hilfvariable ist, die nicht in Γ und nicht in t vorkommt und die Substitution $[x \leftarrow t]$ wohlsortiert ist. ■

Ohne Beweis übernehmen wir das folgende Lemma (vergleiche [SS89, Proposition I.4.9, p.19]), das man mit struktureller Induktion zeigen kann.

5.7 LEMMA: *Für jede Sorte $S \in$ **SORT**$_\Sigma$ und jeden wohlsortierten, nicht-variablen Term s der Sorte S gibt es eine Termdeklaration $(t{<}R) \in$ **TD**$_\Sigma$ mit $R \sqsubseteq S$ und eine wohlsortierte Substitution σ mit $\sigma(t) = s$.* ■

5.8 LEMMA: *Der angegebene Unifikationsalgorithmus ist korrekt.*
Beweis: Durch Anwendung der Regeln **VV1**, **VV2**, **VV5**, **VT2**, **VT4**, **VT5** und **TT2** vergrößert man die Menge der Unifikatoren sicher nicht. Hat man einen Unifikator für $x = z, y = z, \Gamma$, so ist das natürlich (mit Symmetrie und Transitivität der Gleichheit auch ein Unifikator von $x = y, \Gamma$, also ist **VV3** korrekt. Entsprechend kann man die Korrektheit von **VV4**, **VT3** und **TT1** zeigen, wobei man zusätzlich die Substitutionseigenschaft der Gleichheit braucht. Die Korrektheit der Regel **VT1** ergibt sich wiederum aus der Substitutionseigenschaft der Gleichheit. Hat man eine Lösung σ von $x = t, \Gamma[x \leftarrow t]$, so gilt insbesondere $\sigma(x) = \sigma(t)$, also gilt $\sigma = sigma \circ [x \leftarrow t]$ und man erhält in $\sigma(\Gamma) = \sigma \circ [x \leftarrow t](\Gamma) = \sigma(\Gamma[x \leftarrow t])$. ■

5.9 THEOREM: *Alle Regeln außer* **VV3**, **VV4**, **VT3** *sind vollständig.*
Beweis: Für die Regeln **VV1**, **VV2** und **VT2** ist Behauptung trivial.
VV5 ist vollständig, denn sei σ eine Substitution mit $\sigma(x) = \sigma(y)$. Da σ wohlsortiert ist, gilt $\mathfrak{s}(x) \in$ **SORT**$_\Sigma(\sigma(x))$ und $\mathfrak{s}(y) \in$ **SORT**$_\Sigma(\sigma(x))$. Deshalb gibt es eine Termdeklaration $f(s_1, \ldots, s_n){<}S$ mit $S \sqsubseteq \mathfrak{s}(x)$ und $f(r_1, \ldots, r_n){<}R$ mit $R \sqsubseteq \mathfrak{s}(y)$ (vergleiche Lemma 5.7).
Analog ergibt sich die Vollständigkeit von **VT4**. Die Vollständigkeit von **VT5** und **TT2** ergibt sich daraus, daß Terme der Form $x = t$, wo x in t vorkommt, oder die nicht mit dem gleichen Kopfsymbol anfangen, wie im unsortierten Fall nicht unifiziert werden können. ■

5.10 THEOREM: *Der angegebene Unifikationsalgorithmus ist vollständig.*
Beweis: Sei σ ein Unifikator (σ ist dann idempotent, das heißt es gilt $\sigma \circ \sigma = \sigma$) in $\mathbf{U}_\Sigma(\Gamma)$, dessen Träger (das ist die Menge $\{x \mid \sigma(x) \neq x\}$) gerade der Menge der Variablen aus Γ entspricht. Wir teilen Γ in zwei Teilmengen auf, Gleichungen die schon gelöst sind (bezeichnet mit Γ_L) und solche die noch unbehandelt sind (Γ_U genannt).

In der ersten Menge sind die gelösten Gleichungen in Γ und die Abkömmlinge die aus ihnen entstehen, wenn man die Regel **VT1** auf Γ anwendet.
Als fundiertes Komplexitätsmaß $\mu(\sigma, \Gamma)$ nehmen wir die Multimenge aller Termtiefen in $\sigma(\Gamma_U)$. Der Beweis läuft darüber, daß es zu (σ, Γ) ein Paar (σ', Γ') gibt, so daß Γ' aus Γ in einem Schritt mit einer Unifikationsregel abgeleitet werden kann, wobei σ' ein Unifikator von Γ' ist, so daß σ' auf alten Variablen mit σ übereinstimmt und daß gilt $\mu(\sigma', \Gamma') < \mu(\sigma, \Gamma)$.
Wenn $\mu(\sigma', \Gamma')$ minimal ist, ist die Multimenge leer, also sind alle Gleichungen gelöst und wir sind fertig. Nun zeigen wir, daß man im anderen Fall immer eine Regel anwenden kann, die die Komplexität verringert. Zuerst stellen wir fest, daß durch **VT1** die Komplexität nicht zunimmt. Diese Regel erhöht die Komplexität der Terme in $\sigma(\Gamma)$ nicht, da man aus $\sigma(x) = \sigma(t)$ erhält $\sigma \circ [x \leftarrow t] = \sigma$ wegen der Idempotenz von σ. Nach der Definition von Γ_L bleiben gelöste Gleichungen nach der Regelanwendung in dieser Menge.
Sei nun $s = t$ in Γ_U:
<u>1.Fall:</u> Weder s noch t sind Variablen. Ist Γ lösbar, dann ist **TT1** anwendbar und damit läßt sich ohne Veränderung der Lösungsmenge $\mu(\sigma, \Gamma)$ reduzieren.
<u>2.Fall:</u> Die Gleichung hat die Gestalt $x = f(t_1, \ldots, t_n)$. Dann kommt x nicht in $f(t_1, \ldots, t_n)$ vor und $\mathfrak{s}(x) \notin \mathbf{SORT}_\Sigma(f(t_1, \ldots, t_n))$ (sonst ist die Gleichung schon gelöst und **VT1** anwendbar. Den Fall $f(t_1, \ldots, t_n) = x$ brauchen wir nicht zu betrachten, da wir dann **VT2** anwenden, was ja eine vollständige Regel ist, die offensichtlich die Komplexität nicht vergrößert.) Dann existiert wieder eine Termdeklaration (vergleiche Lemma 5.7) $f(s_1, \ldots, s_n) \lessdot S$ mit $S \sqsubseteq \mathfrak{s}(x)$ und eine Substitution τ mit $\tau(f(s_1, \ldots, s_n)) = \sigma(f(t_1, \ldots, t_n))$. Mit **VT3** und **VT1** erhält man ein neues System Γ'. Da die Gleichung $x = f(s_1, \ldots, s_n)$ gelöst ist, haben wir $\mu(\sigma \cup \tau, \Gamma') < \mu(\sigma, \Gamma)$, da die Tiefe von $\sigma(f(t_1, \ldots, t_n))$ größer ist als die Termtiefen von $\sigma(t_i)$ und $\sigma(s_i)$. Weiterhin gilt $\sigma \cup \tau$ ist Lösung von Γ' mit $\sigma \cup \tau$ stimmt auf den Variablen von Γ mit σ überein.
<u>3.Fall:</u> Die Gleichung hat die Gestalt $x = y$. Ist **VV2** anwendbar, so führt die Anwendung dieser vollständigen Regel zu einer Verringerung der Komplexität, da ja eine Gleichung nun gelöst ist. Nehmen wir also an, daß **VV2** nicht anwendbar ist. Da $\sigma(x) = \sigma(y)$ gilt, haben wir $\mathfrak{s}(x) \in \mathbf{SORT}_\Sigma(\sigma(x))$ und $\mathfrak{s}(y) \in \mathbf{SORT}_\Sigma(\sigma(x))$. Wir betrachten nun zwei Teilfälle, nämlich $\sigma(x)$ ist Variable oder aber nicht.
Sei $\sigma(x)$ Variable, dann gilt $\mathfrak{s}(\sigma(x)) \sqsubseteq \mathfrak{s}(x)$ und $\mathfrak{s}(\sigma(x)) \sqsubseteq \mathfrak{s}(y)$, also gibt es eine maximale Sorte S mit $\mathfrak{s}(\sigma(x)) \sqsubseteq S \sqsubseteq \mathfrak{s}(x)$ und $S \sqsubseteq \mathfrak{s}(y)$. Dann ist aber Regel **VV3** anwendbar und wir erhalten ein neues System Γ'. Mit $\tau = [z \leftarrow \sigma(x)]$ haben wir $\mu(\sigma \cup \tau, \Gamma') < \mu(\sigma, \Gamma)$. Weiterhin gilt $\sigma \cup \tau$ ist Lösung von Γ' mit $\sigma \cup \tau$ stimmt auf den Variablen von Γ mit σ überein.
Sei $\sigma(x)$ keine Variable, dann gibt es Termdeklarationen $f(s_1, \ldots, s_n) \lessdot S$ mit $S \sqsubseteq \mathfrak{s}(x)$ und $f(t_1, \ldots, t_n) \lessdot R$ mit $R \sqsubseteq \mathfrak{s}(x)$ und eine Substitution τ, so daß gilt $\sigma(x) = \tau(f(s_1, \ldots, s_n))$ und $\sigma(y) = \tau(f(t_1, \ldots, t_n))$. Mit den Regeln **VV4** und **VT1** erhalten wir dann Γ'. Die Substitution $\sigma \cup \tau$ stimmt auf den Variablen von Γ wieder mit σ

überein. Weiterhin gilt $\mu(\sigma \cup \tau, \Gamma') < \mu(\sigma, \Gamma)$. ■

Wir stellen nun noch die wichtigsten Eigenschaften der allgemeinen sortierten Unifikation vor, ohne auf die Beweise einzugehen. Man findet diese in [SS89].

5.11 BEMERKUNG: Es gibt endliche Signaturen, in denen die Unifikation infinitär ist (vergleiche Übungsblatt 8, Aufgabe 1). ■

5.12 THEOREM: *Das allgemeine sortierte Unifikationsproblem ist unentscheidbar.* ■

Im Unifikationsalgorithmus sind es die Regeln **VV4** und **VT3**, die für eventuelles Nichtterminieren verantwortlich sind.

5.13 DEFINITION: Eine Signatur heißt *regulär*, wenn die Untersortenrelation $\sqsubseteq$ eine partielle Ordnung ist und für jeden Term t hat die Menge $\mathbf{SORT}_\Sigma(t)$ der Sorten von t eine eindeutige kleinste Sorte.[2]
Eine Signatur heißt *fast elementar*, wenn sie linear ist und in jeder Termdeklaration $t{<}S$ ist jede Variable ein direkter Unterterm von t, das heißt in Termdeklarationen $f(t_1, \ldots, t_m){<}S$ sind die Terme t_i entweder variablenfrei oder aber selbst schon Variablen, weiterhin darf keine Variable zweimal auftauchen. ■

5.14 THEOREM: *Sei Σ eine endliche Signatur.*

1. *Ist Σ elementar, dann ist die Unifikation entscheidbar.*
2. *Ist Σ linear, dann ist lineare Unifikation (das heißt Unifikation, auf Unifikationsproblemen Γ, in denen jede Variable höchstens einmal vorkommt) entscheidbar.*
3. *Ist Σ regulär und fast elementar, dann ist die Unifikation finitär und entscheidbar.*
4. *Sind alle Funktionssymbole entweder Konstanten oder einstellige Funktionen, dann ist die Unifikation entscheidbar.* ■

[2]Zum Beispiel die einfachen Signaturen sind stets regulär.

Kapitel 6

Sortierte Resolution

In diesem Kapitel wollen wir uns ansehen, wie Sorten in einem Resolutionskalkül verwendet werden können. Die Hauptvoraussetzung, die sortierte Unifikation, haben wir im vorigen Abschnitt kennengelernt. Um einen Resolutionskalkül anwenden zu können, muß zuerst aus Formeln die Klauselnormalform erzeugt werden. Wie dies geht, haben wir in Kapitel 1 gesehen. Zuerst werden die auf Seite 6 angegebenen Transformationsregeln angewendet, der einzige Unterschied besteht darin, daß nun die Variablen die Sorteninformation tragen, das heißt daß die Sorte als Index mit angegeben wird (zum Beispiel x_S). Durch Anwendung der Regeln erhält man eine Pränexform, bei der alle Quantoren vorne stehen. Dann werden Existenzquantoren durch Skolemisierung eliminiert. Dies geschieht wie folgt:
Eine Pränexformel φ die eine Unterformel der Form $\exists x_S.\ \psi$ enthält wird skolemisiert, in dem man die folgenden Schritte für jeden Existenzquantor von links nach rechts durchgehend durchführt. Zuerst wird die Menge der Variablen bestimmt, die in einem Allquantor[1] vor dem betreffenden Existenzquantor auftauchen, wir bezeichnen sie mit $\{x_1, \ldots, x_n\}$. Dann wird eine neue n-stellige Skolemfunktion f eingeführt, die die Sorte $f \lessdot \mathfrak{s}(x_1) \times \cdots \times \mathfrak{s}(x_n) \to S$ hat. Schließlich wird in ψ jedes Vorkommen von x_S durch $f(x_1, \ldots, x_n)$ ersetzt.
Es gilt das folgende Lemma.

6.1 LEMMA: *Sei φ eine Pränexformel bezüglich einer Signatur Σ und sei φ' die skolemisierte Form von φ bezüglich der Signatur Σ', die aus Σ durch Hinzunahme der Skolemfunkionen entsteht, dann hat φ ein Σ-Modell genau dann, wenn φ' ein Σ'-Modell hat.*

Beweis: Für den Beweis nehmen wir vereinfachend an, daß nur ein einziger Skolemisierungsschritt durchgeführt wird. Hat man ein Modell $\mathcal{M}$ von φ mit einer Belegung $[x_i \leftarrow a_i]$ und x_S wird durch $a \in \mathcal{U}_S$ in $\exists x_S.\ \psi$ interpretiert. Dann kann man aus dieser Σ-Interpretation eine Σ'-Interpretation für f konstruieren, die gerade f so interpretiert, daß $\mathcal{I}(f)(a_1, \ldots, a_n) = a$, ansonsten wird f beliebig interpretiert. Beachte, daß die Menge $\mathcal{U}_S$ als nicht-leer angenommen wird. Die umgekehrte Richtung zeigt man

[1] Alternativ dazu betrachtet man die in ψ frei vorkommenden Variablen.

entsprechend. ■

Auf der Klauselnormalform kann man nun den Resolutionskalkül anwenden. Mit den beiden Kalkülregeln **Res** und **Fak** werden ausgehend von einer Klauselmenge $\mathcal{C}$ neue Klauseln erzeugt und zu der aktuellen Klauselmenge hinzugefügt bis das einzige Axiom (für Falschheit), die leere Klausel $\Box$, abgeleitet ist. Hat man dies abgeleitet, so hat man gezeigt, daß die Ausgangsklauselmenge unerfüllbar ist. Zur Behandlung der Gleichheit hat man zusätzlich die Paramodulationsregel zur Verfügung.
Die allgemeine Resolutionsregel hat die Gestalt:

Res
$$\frac{\begin{array}{ll} L \vee K_1 \vee \ldots \vee K_n & \\ \neg L' \vee M_1 \vee \ldots \vee M_m & \sigma(L) = \sigma(L') \end{array}}{\sigma(K_1) \vee \ldots \vee \sigma(K_n) \vee \sigma(M_1) \vee \ldots \vee \sigma(M_m)}$$

Sei $L = P(s_1, \ldots, s_k)$ und $L' = P(t_1, \ldots, t_k)$, dann steht σ für ein beliebiges Element aus $\mathbf{MU}(\{s_1 = t_1, \ldots, s_k = t_k\})$.
Die Faktorisierungsregel ist:

Fak
$$\frac{L \vee L' \vee K_1 \vee \ldots \vee K_n \qquad \sigma(L) = \sigma(L')}{\sigma(L) \vee \sigma(K_1) \vee \ldots \vee \sigma(K_n)}$$

wobei σ wiederum ein allgemeinster Unifikator von L und L' ist.
Die Paramodulationsregel lautet:

Para
$$\frac{\begin{array}{ll} t \doteq s \vee K_1 \vee \ldots \vee K_n & \\ L[\ldots t' \ldots] \vee M_1 \vee \ldots \vee M_m & \sigma(t') = \sigma(t) \end{array}}{\sigma(L[\ldots s \ldots]) \vee \sigma(K_1) \vee \ldots \vee \sigma(K_n) \vee \sigma(M_1) \vee \ldots \vee \sigma(M_m)}$$

Bei Resolutionsregel und Paramodulationsregel nehmen wir an, daß die Klauseln variablendisjunkt sind. Im Falle der Gleichheitsbehandlung benötigt man die Reflexivität als zusätzliches Axiom, das heißt Axiome der Gestalt „$x_S = x_S$“. Weiterhin muß man vorsichtig sein, daß man durch die Paramodulation nicht zu Termen gelangt, die nicht wohlsortiert sind. Um dies zu vermeiden nehmen wir für das weitere an, daß wir eine Obersorte T zu allen Sorten haben und daß im Falle der Gleichheitsbehandlung die Klausel $x_T = x_T$ in jeder Klauselmenge ist.
Der Kalkül wird so verwendet, daß man mit der initialen Klauselmenge beginnend mit den Regeln **Res**, **Fak** und **Para** neue Klauseln herleitet und diese zur aktuellen Klauselmenge hinzufügt bis die leere Klausel abgeleitet ist. In diesem Fall war die Ausgangsformelmenge unerfüllbar. Für den seltenen Fall, daß keine Regel mehr anwendbar ist, die leere Klausel aber nicht abgeleitet werden konnte, war die Ausgangsklauselmenge erfüllbar. Beachte, daß durch den Kalkül a priori zur aktuellen Klauselmenge stets nur Klauseln hinzugefügt werden, es bleiben also zum Beispiel die Eltern eines Resolutionsschritts in der Klauselmenge enthalten. Allerdings hat man eine Reihe zusätzlicher Verfeinerungen entwickelt, die es erlauben, überflüssige Klauseln zu löschen.
Wie man sich leicht überlegen kann, gilt das folgende Lemma:

6.2 LEMMA: *Resolution, Faktorisierung und Paramodulation sind widerlegungskor-*

rekte Regeln. ■

Wie immer ist die Vollständigkeit schwerer zu zeigen als die Korrektheit. Wir werden dies im folgenden nur für Resolution und Faktorisierung tun. Für den allgemeinen Vollständigkeitsbeweis siehe [SS89, S.138 ff]. Es sei lediglich angemerkt, daß im Gegensatz zur unsortierten Paramodulation, wo man nicht in Variablen paramodulieren muß und auf die sogenannten funktional-reflexiven Axiome (das heißt auf Axiome der Gestalt „$f(x_1, \ldots, x_n) = f(x_1, \ldots, x_n)$") verzichten kann, dies im sortierten Fall nicht mehr der Fall ist:

6.3 BEISPIEL:

- Sei eine Signatur Σ mit **SORT** $= \{A, B, T\}$, **SD** $= \{A \sqsubseteq T, B \sqsubseteq T\}$ und **TD** $= \{a{<}A, B{<}B\}$ gegeben. Die Klauselmenge $\mathcal{C} = \{\{P(x_A)\}, \{\neg P(y_B)\}, \{a = b\}\}$ ist zwar unerfüllbar, aber dies kann nur gezeigt werden, in dem man in eine der Variablen x_A oder y_B paramoduliert, da x_A und y_B nicht unifizierbar sind.

- Sei die folgende nicht-einfache Signatur Σ mit

  SORT $= \{A, B, C, D, T\}$,
  SD $= \{A \sqsubseteq T, B \sqsubseteq T, C \sqsubseteq T, D \sqsubseteq T\}$ und
  TD $= \{a{<}A, b{<}B, c{<}C, d{<}D, g{<}T \to T, g{<}A \to C, g{<}B \to D\}$

  gegeben. Dann ist die Klauselmenge $\mathcal{C} = \{\{z_T = z_T\}, \{x_C \neq y_D\}, \{a = b\}\}$ unerfüllbar, da $g(a) = g(b)$ in allen Modellen gilt, was $x_C \neq y_D$ widerspricht. Mit Resolution, Faktorisierung und Paramodulation läßt sich dieser Widerspruch aber nicht ableiten, da das Funktionssymbol g nicht eingeführt werden kann. ■

Der Vollständigkeitsbeweis gliedert sich im wesentlichen in drei Teile: erstens den Beweis der Grundvollständigkeit (das heißt der Vollständigkeit, wenn die Klauselmenge keine Variablen enthält), zweitens das Herbrand-Theorem, das garantiert, daß man zu einer widerlegbaren Klauselmenge eine entsprechende variablenfreie Klauselmenge findet, und drittens ein Lifting-Lemma, durch das gewährleistet wird, daß man entsprechende Grundresolutionsschritte auf den allgemeinen Fall liften kann.

6.4 THEOREM: *Sei $\mathcal{C}_{\mathrm{gr}}$ eine unerfüllbare Grundklauselmenge ohne Gleichheit, dann gibt es eine Ableitung der leeren Klausel mit der Resolutions- und Faktorisierungsregel. Darüberhinaus ist Unerfüllbarkeit entscheidbar.*

Beweis: Der Beweis dieses Satzes wird nach Anderson-Bledsoe [AB70] per „k-Parameter-Induktion" geführt. Sei $k(\mathcal{C}_{\mathrm{gr}})$ definiert als die Summe $\sum_{C \in \mathcal{C}_{\mathrm{gr}}} |C| - 1$, wobei $|C|$ die Anzahl der Literale in der Klausel C angibt. Sei $\mathcal{C}_{\mathrm{gr}}$ unerfüllbar. Wenn gilt $k(\mathcal{C}_{\mathrm{gr}}) = 0$, dann bestehen alle Klauseln aus genau einem Literal, da die Klauselmenge unerfüllbar ist, muß sie eine Klausel und ihr Negat enthalten, die in einem Schritt mit der Resolutionsregel zur leeren Klausel resolvieren. Gilt $k(\mathcal{C}_{\mathrm{gr}}) > 0$, dann gibt es eine nicht-Unit-Klausel C in $\mathcal{C}$. Wir teilen die Klausel C in zwei (nicht-leere) Klauseln C_1 und C_2 auf und bilden zwei Klauselmengen $\mathcal{C}_1$ und $\mathcal{C}_2$, die aus $\mathcal{C}$ dadurch entstehen,

daß wir in $\mathcal{C}$ die Klausel C durch C_1 beziehungsweise C_2 ersetzen. Da die Klauseln C_1 und C_2 Teilklauseln von C sind, sind die neuen Klauselmengen unerfüllbar. Weiterhin gilt auch $k(\mathcal{C}_1) < k(\mathcal{C})$ und $k(\mathcal{C}_2) < k(\mathcal{C})$, also läßt sich nach Induktionsvoraussetzung sowohl für $\mathcal{C}_1$ als auch für $\mathcal{C}_2$ eine Widerlegung finden. Da aber alle alle Klauseln variablenfrei sind, lassen sich diese Widerlegungen zu einer Gesamtwiderlegung für $\mathcal{C}$ kombinieren.
Die Entscheidbarkeit ergibt sich daraus, daß nur eine endliche Anzahl von Klauseln aus der Klauselmenge $\mathcal{C}$ herleitbar ist. ■

Herbrand-Theorem:

6.5 THEOREM: *Sei eine Signatur Σ gegeben. Eine Klauselmenge $\mathcal{C}$ ist Σ-unerfüllbar, dann und nur dann wenn es eine endliche Menge von Σ-Grundinstanzen aus $\mathcal{C}$ gibt, die unerfüllbar ist.*

Beweis: Der Beweis dieses Satzes wird aus dem unsortierten Fall übertragen. Wir gehen nur auf die nicht-triviale Richtung des Theorems ein. Sei $\mathcal{C}$ eine Σ-unerfüllbare Klauselmenge. Zu dieser Klauselmenge können wir durch Klauselnormalformbildung von $\Re(\mathcal{C}) = \hat{\Re}(\mathcal{C}) \cup \mathbf{AX}_\Sigma$ die zugehörige unsortierte Klauselmenge bilden. Nach dem Herbrand-Theorem für unsortierte Logik findet man zu dieser Klauselmenge eine $\Re(\Sigma)$-unerfüllbare Grundklauselmenge $\mathcal{C}_{\Re,\mathrm{gr}} \cup \mathbf{AX}_{\Sigma,\mathrm{gr}}$ aus Instanzen von $\hat{\Re}(\mathcal{C}) \cup \mathbf{AX}_\Sigma$. Insbesondere ist $\mathcal{C}_{\Re,\mathrm{gr}} \cup \mathbf{AX}_\Sigma$ widerlegbar. Es könnte nun sein, daß einige der Klauseln in $\mathcal{C}_{\Re,\mathrm{gr}}$ nicht Σ-Grundinstanzen von $\mathcal{C}$ entsprechen. Diese Klauseln können aufgrund der Struktur der Relativierung gelöscht werden, da sie nicht zur Widerlegung beitragen. Sie haben die Struktur $\neg P_S(t)$, wobei t ein Grundterm ist, der nicht die Sorte S hat. (Für Einzelheiten siehe [SS89, S.93].) Wir können schließen, daß die so entstandene Klauselmenge eine endliche Σ-unerfüllbare Menge von Σ-Grundinstanzen von Klauseln aus $\mathcal{C}$ ist. ■

6.6 LEMMA: (LIFTING-LEMMA) *Seien C_1' und C_2' zwei Instanzen von C_1 und C_2 und C' eine Resolvente von C_1' und C_2', dann gibt es eine Resolvente C von C_1 und C_2, so daß C' eine Instanz von C ist.*

Beweis: Wir nehmen an, daß die Variablen in C_1 und C_2 disjunkt sind (andernfalls wird umbenannt). Seien L_1' und L_2' die Literale auf denen resolviert wird und sei C' definiert als $(\gamma(C_1')\backslash\gamma(L_1')) \cup (\gamma(C_2')\backslash\gamma(L_2'))$, wobei γ ein allgemeinster Unifikator von L_1' und L_2' ist. Da C_1' und C_2' Instanzen von C_1 und C_2 sind, gibt es eine Substitution θ mit $C_1' = \theta(C_1)$ und $C_2' = \theta(C_2)$. Seien nun $L_i^1, \ldots, L_i^{r_i}$ die Literale in C_i, die gerade L_i' entsprechen (das heißt $\theta(L_i^1) = \ldots = \theta(L_i^{r_i}) = L_i'$) für $i = 1, 2$. Wenn $r_i > 1$, bestimme einen allgemeinsten Unifikator λ_i von $\{L_i^1, \ldots, L_i^{r_i}\}$ und definiere $L_i = \lambda_i(L_i^1)$ für $i = 1, 2$. (Beachte, daß λ_i allgemeinster Unifikator ist, also alle $\lambda_i(L_i^j)$ mit $j = 1, \ldots, r_i$ übereinstimmen.) Dann kommt L_i im Faktor $\lambda_i(C_i)$ von C_i vor. Wenn $r_i = 1$, sei λ_i die leere Substitution und wieder $L_i = \lambda(L_i^1)$. Wir definieren $\lambda = \lambda_1 \cup \lambda_2$ Dann ist L_i' eine Instanz von L_i. Da L_1' und L_2' unifizierbar sind, sind L_1

und L_2 unifizierbar. Sei σ der allgemeinste Unifikator von L_1 und L_2 und

$$\begin{aligned}
C = \ & (\sigma(\lambda(C_1))\backslash\sigma(L_1)) \cup (\sigma(\lambda(C_2))\backslash\sigma(L_2)) \\
= \ & (\sigma(\lambda(C_1))\backslash\sigma(\lambda(\{L_1^1,\ldots,L_1^{r_1}\}))) \cup (\sigma(\lambda(C_2))\backslash\sigma(\lambda(\{L_2^1,\ldots,L_2^{r_2}\}))) \\
= \ & (\sigma\circ\lambda(C_1)\backslash\sigma\circ\lambda(\{L_1^1,\ldots,L_1^{r_1}\})) \cup (\sigma\circ\lambda(C_2)\backslash\sigma\circ\lambda(\{L_2^1,\ldots,L_2^{r_2}\}))
\end{aligned}$$

Also ist C eine Resolvente von C_1 und C_2. Wegen

$$\begin{aligned}
C' = \ & (\gamma(C_1')\backslash\gamma(L_1')) \cup (\gamma(C_2')\backslash\gamma(L_2')) \\
= \ & (\gamma(\theta(C_1))\backslash\gamma(\theta(\{L_1^1,\ldots,L_1^{r_1}\}))) \cup (\gamma(\theta(C_2))\backslash\gamma(\theta(\{L_2^1,\ldots,L_2^{r_2}\}))) \\
= \ & (\gamma\circ\theta(C_1)\backslash\gamma\circ\theta(\{L_1^1,\ldots,L_1^{r_1}\})) \cup (\gamma\circ\theta(C_2)\backslash\gamma\circ\theta(\{L_2^1,\ldots,L_2^{r_2}\}))
\end{aligned}$$

ist C' eine Instanz von C. Da $\sigma\circ\lambda$ allgemeiner ist als $\gamma\circ\theta$ sind wir fertig. ■

6.7 THEOREM: *Sei $\mathcal{C}$ eine unerfüllbare Klauselmenge ohne Gleichheit, dann gibt es mit Resolutions und Faktorisierung eine Ableitung der leeren Klausel.*

Beweis: Nach dem Herbrand-Theorem gibt es zu $\mathcal{C}$ eine endliche unerfüllbare Grundklauselmenge $\mathcal{C}_{\text{gr}}$ aus Instanzen von $\mathcal{C}$. Diese Klauselmenge kann aufgrund der Grundvollständigkeit (vergleiche Satz 6.4) widerlegt werden. Mit dem Lifting-Lemma folgt, daß man einen solchen Beweis zu einem Beweis auf der ursprünglichen Klauselmenge liften kann. ■

Zum Abschluß dieses Kapitels wollen wir uns ein Beispiel ansehen. Ein Beispiel haben wir in Beispiel 3.6 kennengelernt, wo wir die Operationalisierung durch die Termrelativierung durchgeführt haben. Ein weiteres Beispiel war Schuberts „Steamroller" (vergleiche Übungsblatt 1, Aufgabe 2, Übungsblatt 2, Aufgabe 1, Übungsblatt 4, Aufgabe 2 und Übungsblatt 5, Aufgabe 1). In diesen Fällen lag eine einfache, baumartige Signatur vor. Im folgenden Beispiel ist die Sortenstruktur etwas komplizierter. Es ist nach [SS89] zitiert.

6.8 BEISPIEL: DER LÖWE UND DAS EINHORN: „Als Alice in den Wald des Vergessens kam, vergaß sie nicht alles, sondern nur gewisse Dinge. Sie vergaß oft ihren Namen und am wahrscheinlichsten vergaß sie den Wochentag. Nun waren der Löwe und das Einhorn häufige Besucher dieses Waldes, allerdings handelt es sich um seltsame Geschöpfe. Der Löwe lügt montags, dienstags und mittwochs und sagt an den anderen Wochentagen die Wahrheit, wohingegen das Einhorn donnerstags, freitags und samstags lügt und an den anderen Tagen die Wahrheit sagt. Eines Tages begegnete Alice dem Löwen und dem Einhorn, die sich unter einem Baum ausruhten. Sie sagten zu Alice:

Löwe: Gestern war einer meiner Lügentage.
Einhorn: Gestern war einer meiner Lügentage.

Alice, die ein aufgewecktes Mädchen war, konnte aus diesen Angaben den Wochentag herleiten. Welcher war es?"

Wie sieht eine Formalisierung in Logik aus? In unsortierter Logik kann man eine Formulierung wählen, die die einstelligen Prädikate Mo, Di, Mi, Do, Fr, Sa und So

für die Wochentage verwendet, eine einstellige Funktion „$\mathsf{gestern}$“ mit der üblichen Semantik und einstellige Prädikate „$\mathsf{lügt\_löwe}$“ und „$\mathsf{lügt\_einhorn}$“ die für die Lügentage des Löwen beziehungsweise des Einhorns stehen. Weiterhin verwenden wir ein dreistelliges Prädikat $\mathsf{sagt}(x, y, z)$, das genau dann wahr ist, wenn x am Tage z sagt, daß er am Tage y lügt. Man hat dann die folgende Problemstellung:
Axiomatisierung der Wochentage:

$\forall x.\, \mathsf{Mo}(x) \Leftrightarrow \neg(\mathsf{Di}(x) \vee \mathsf{Mi}(x) \vee \mathsf{Do}(x) \vee \mathsf{Fr}(x) \vee \mathsf{Sa}(x) \vee \mathsf{So}(x))$
$\forall x.\, \mathsf{Di}(x) \Leftrightarrow \neg(\mathsf{Mo}(x) \vee \mathsf{Mi}(x) \vee \mathsf{Do}(x) \vee \mathsf{Fr}(x) \vee \mathsf{Sa}(x) \vee \mathsf{So}(x))$
$\forall x.\, \mathsf{Mi}(x) \Leftrightarrow \neg(\mathsf{Mo}(x) \vee \mathsf{Di}(x) \vee \mathsf{Do}(x) \vee \mathsf{Fr}(x) \vee \mathsf{Sa}(x) \vee \mathsf{So}(x))$
$\forall x.\, \mathsf{Do}(x) \Leftrightarrow \neg(\mathsf{Mo}(x) \vee \mathsf{Di}(x) \vee \mathsf{Mi}(x) \vee \mathsf{Fr}(x) \vee \mathsf{Sa}(x) \vee \mathsf{So}(x))$
$\forall x.\, \mathsf{Fr}(x) \Leftrightarrow \neg(\mathsf{Mo}(x) \vee \mathsf{Di}(x) \vee \mathsf{Mi}(x) \vee \mathsf{Do}(x) \vee \mathsf{Sa}(x) \vee \mathsf{So}(x))$
$\forall x.\, \mathsf{Sa}(x) \Leftrightarrow \neg(\mathsf{Mo}(x) \vee \mathsf{Di}(x) \vee \mathsf{Mi}(x) \vee \mathsf{Do}(x) \vee \mathsf{Fr}(x) \vee \mathsf{So}(x))$
$\forall x.\, \mathsf{So}(x) \Leftrightarrow \neg(\mathsf{Mo}(x) \vee \mathsf{Di}(x) \vee \mathsf{Mi}(x) \vee \mathsf{Do}(x) \vee \mathsf{Fr}(x) \vee \mathsf{Sa}(x))$

Die Funktion „$\mathsf{gestern}$“ hat die Axiomatisierung:

$\forall x.\, \mathsf{Mo}(\mathsf{gestern}(x)) \Leftrightarrow \mathsf{Di}(x)$
$\forall x.\, \mathsf{Di}(\mathsf{gestern}(x)) \Leftrightarrow \mathsf{Mi}(x)$
$\forall x.\, \mathsf{Mi}(\mathsf{gestern}(x)) \Leftrightarrow \mathsf{Do}(x)$
$\forall x.\, \mathsf{Do}(\mathsf{gestern}(x)) \Leftrightarrow \mathsf{Fr}(x)$
$\forall x.\, \mathsf{Fr}(\mathsf{gestern}(x)) \Leftrightarrow \mathsf{Sa}(x)$
$\forall x.\, \mathsf{Sa}(\mathsf{gestern}(x)) \Leftrightarrow \mathsf{So}(x)$
$\forall x.\, \mathsf{So}(\mathsf{gestern}(x)) \Leftrightarrow \mathsf{Mo}(x)$

Die Prädikate $\mathsf{lügt\_löwe}$ und und $\mathsf{lügt\_einhorn}$ ergeben sich als:

$\forall x.\, \mathsf{lügt\_löwe}(x) \Leftrightarrow \mathsf{Mo}(x) \vee \mathsf{Di}(x) \vee \mathsf{Mi}(x)$
$\forall x.\, \mathsf{lügt\_einhorn}(x) \Leftrightarrow \mathsf{Do}(x) \vee \mathsf{Fr}(x) \vee \mathsf{Sa}(x)$

Die Axiomatisierung des Prädikates sagt ist:

$\forall y, z.\, \mathsf{lügt\_löwe}(y) \wedge \mathsf{lügt\_löwe}(z) \Rightarrow \neg\mathsf{sagt}(\mathsf{Löwe}, y, z)$
$\forall y, z.\, \mathsf{lügt\_löwe}(y) \wedge \neg\mathsf{lügt\_löwe}(z) \Rightarrow \mathsf{sagt}(\mathsf{Löwe}, y, z)$
$\forall y, z.\, \neg\mathsf{lügt\_löwe}(y) \wedge \mathsf{lügt\_löwe}(z) \Rightarrow \mathsf{sagt}(\mathsf{Löwe}, y, z)$
$\forall y, z.\, \neg\mathsf{lügt\_löwe}(y) \wedge \neg\mathsf{lügt\_löwe}(z) \Rightarrow \neg\mathsf{sagt}(\mathsf{Löwe}, y, z)$
$\forall y, z.\, \mathsf{lügt\_einhorn}(y) \wedge \mathsf{lügt\_einhorn}(z) \Rightarrow \neg\mathsf{sagt}(\mathsf{Einhorn}, y, z)$
$\forall y, z.\, \mathsf{lügt\_einhorn}(y) \wedge \neg\mathsf{lügt\_einhorn}(z) \Rightarrow \mathsf{sagt}(\mathsf{Einhorn}, y, z)$
$\forall y, z.\, \neg\mathsf{lügt\_einhorn}(y) \wedge \mathsf{lügt\_einhorn}(z) \Rightarrow \mathsf{sagt}(\mathsf{Einhorn}, y, z)$
$\forall y, z.\, \neg\mathsf{lügt\_einhorn}(y) \wedge \neg\mathsf{lügt\_einhorn}(z) \Rightarrow \neg\mathsf{sagt}(\mathsf{Einhorn}, y, z)$

Das Theorem lautet dann:

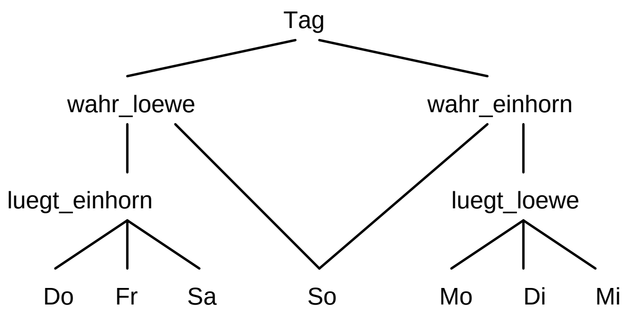
$\exists x.\, \mathsf{sagt}(\mathsf{Löwe}, \mathsf{gestern}(x), x) \wedge \mathsf{sagt}(\mathsf{Einhorn}, \mathsf{gestern}(x), x).$

Wobei eine konstruktive Antwort erwartet wird.
In dieser Formulierung ist das Problem natürlich ziemlich kompliziert, da die Darstellung der Wochentage umständlich ist. Von daher bietet es sich an eine sortierte Formulierung zu wählen, mit der man die Struktur der Tage statt mit vielen Formeln über die Sortenstruktur darstellt. Entscheidend für die Effizienz einer solchen Repräsentation ist dabei die geschickte Wahl der Sorten, der Untersortenbeziehung und der Termdeklarationen.

$$\mathbf{SORT} = \{\mathsf{tag}, \mathsf{wahr\_l\ddot{o}we}, \mathsf{wahr\_einhorn}, \mathsf{l\ddot{u}gt\_l\ddot{o}we}, \mathsf{l\ddot{u}gt\_einhorn}, \mathsf{Mo}, \mathsf{Di}, \mathsf{Mi}, \mathsf{Do}, \mathsf{Fr}, \mathsf{Sa}, \mathsf{So}\}$$

Mit der Sortenstruktur

$$\begin{aligned}\mathbf{SD} = \{&\mathsf{Do} \sqsubseteq \mathsf{l\ddot{u}gt\_einhorn}, \mathsf{Fr} \sqsubseteq \mathsf{l\ddot{u}gt\_einhorn}, \mathsf{Sa} \sqsubseteq \mathsf{l\ddot{u}gt\_einhorn},\\ &\mathsf{Mo} \sqsubseteq \mathsf{l\ddot{u}gt\_l\ddot{o}we}, \mathsf{Di} \sqsubseteq \mathsf{l\ddot{u}gt\_l\ddot{o}we}, \mathsf{Mi} \sqsubseteq \mathsf{l\ddot{u}gt\_l\ddot{o}we},\\ &\mathsf{So} \sqsubseteq \mathsf{wahr\_einhorn}, \mathsf{So} \sqsubseteq \mathsf{wahr\_l\ddot{o}we}, \mathsf{l\ddot{u}gt\_einhorn} \sqsubseteq \mathsf{wahr\_l\ddot{o}we},\\ &\mathsf{l\ddot{u}gt\_l\ddot{o}we} \sqsubseteq \mathsf{wahr\_einhorn}, \mathsf{wahr\_einhorn} \sqsubseteq \mathsf{tag}, \mathsf{wahr\_l\ddot{o}we} \sqsubseteq \mathsf{tag}\}\end{aligned}$$



Die Funktion **gestern** wird nun wie folgt spezifiziert:

$$\begin{aligned}\mathbf{TD} = \{&\mathsf{gestern} \lessdot \mathsf{Di} \to \mathsf{Mo},\\ &\mathsf{gestern} \lessdot \mathsf{Mi} \to \mathsf{Di},\\ &\mathsf{gestern} \lessdot \mathsf{Do} \to \mathsf{Mi},\\ &\mathsf{gestern} \lessdot \mathsf{Fr} \to \mathsf{Do},\\ &\mathsf{gestern} \lessdot \mathsf{Sa} \to \mathsf{Fr},\\ &\mathsf{gestern} \lessdot \mathsf{So} \to \mathsf{Sa},\\ &\mathsf{gestern} \lessdot \mathsf{Mo} \to \mathsf{So}\}\end{aligned}$$

Soweit steckt alles in der Signatur. Die restlichen Formeln vereinfachen sich allerdings auch.

$$\begin{array}{ll}
\forall y_{\mathsf{l\ddot{u}gt_l\ddot{o}we}}.\ \forall z_{\mathsf{l\ddot{u}gt_l\ddot{o}we}}. & \neg\mathsf{sagt}(\mathsf{L\ddot{o}we}, y, z)\\
\forall y_{\mathsf{l\ddot{u}gt_l\ddot{o}we}}.\ \forall z_{\mathsf{wahr_l\ddot{o}we}}. & \mathsf{sagt}(\mathsf{L\ddot{o}we}, y, z)\\
\forall y_{\mathsf{wahr_l\ddot{o}we}}.\ \forall z_{\mathsf{l\ddot{u}gt_l\ddot{o}we}}. & \mathsf{sagt}(\mathsf{L\ddot{o}we}, y, z)\\
\forall y_{\mathsf{wahr_l\ddot{o}we}}.\ \forall z_{\mathsf{wahr_l\ddot{o}we}}. & \neg\mathsf{sagt}(\mathsf{L\ddot{o}we}, y, z)\\
\forall y_{\mathsf{l\ddot{u}gt_einhorn}}.\ \forall z_{\mathsf{l\ddot{u}gt_einhorn}}. & \neg\mathsf{sagt}(\mathsf{Einhorn}, y, z)\\
\forall y_{\mathsf{l\ddot{u}gt_einhorn}}.\ \forall z_{\mathsf{wahr_einhorn}}. & \mathsf{sagt}(\mathsf{Einhorn}, y, z)\\
\forall y_{\mathsf{wahr_einhorn}}.\ \forall z_{\mathsf{l\ddot{u}gt_einhorn}}. & \mathsf{sagt}(\mathsf{Einhorn}, y, z)\\
\forall y_{\mathsf{wahr_einhorn}}.\ \forall z_{\mathsf{wahr_einhorn}}. & \neg\mathsf{sagt}(\mathsf{Einhorn}, y, z)
\end{array}$$

Das Theorem lautet dann:

$\exists x_{\mathsf{tag}}.\ \mathsf{sagt}(\mathsf{L\ddot{o}we}, \mathsf{gestern}(x), x) \land \mathsf{sagt}(\mathsf{Einhorn}, \mathsf{gestern}(x), x)$.

Wobei eine konstruktive Antwort erwartet wird.

Die Skolemnormalform lautet dann:

$$
\begin{array}{l}
\{\{\neg\mathsf{sagt}(\mathsf{L\ddot{o}we}, y_{\mathsf{l\ddot{u}gt\_l\ddot{o}we}}, z_{\mathsf{l\ddot{u}gt\_l\ddot{o}we}})\} \\
\{\mathsf{sagt}(\mathsf{L\ddot{o}we}, y_{\mathsf{l\ddot{u}gt\_l\ddot{o}we}}, z_{\mathsf{wahr\_l\ddot{o}we}})\} \\
\{\mathsf{sagt}(\mathsf{L\ddot{o}we}, y_{\mathsf{wahr\_l\ddot{o}we}}, z_{\mathsf{l\ddot{u}gt\_l\ddot{o}we}})\} \\
\{\neg\mathsf{sagt}(\mathsf{L\ddot{o}we}, y_{\mathsf{wahr\_l\ddot{o}we}}, z_{\mathsf{wahr\_l\ddot{o}we}})\} \\
\{\neg\mathsf{sagt}(\mathsf{Einhorn}, y_{\mathsf{l\ddot{u}gt\_einhorn}}, z_{\mathsf{l\ddot{u}gt\_einhorn}})\} \\
\{\mathsf{sagt}(\mathsf{Einhorn}, y_{\mathsf{l\ddot{u}gt\_einhorn}}, z_{\mathsf{wahr\_einhorn}})\} \\
\{\mathsf{sagt}(\mathsf{Einhorn}, y_{\mathsf{wahr\_einhorn}}, z_{\mathsf{l\ddot{u}gt\_einhorn}})\} \\
\{\neg\mathsf{sagt}(\mathsf{Einhorn}, y_{\mathsf{wahr\_einhorn}}, z_{\mathsf{wahr\_einhorn}})\} \\
\{\neg\mathsf{sagt}(\mathsf{L\ddot{o}we}, \mathsf{gestern}(x_{\mathsf{tag}}), x_{\mathsf{tag}}) \vee \neg\mathsf{sagt}(\mathsf{Einhorn}, \mathsf{gestern}(x_{\mathsf{tag}}), x_{\mathsf{tag}})\}\}
\end{array}
$$

Um einen Widerspruch herzuleiten gibt es nun genau zwei Möglichkeiten zur Resolution mit dem erste Literal der letzten Klausel und zwei zum zur Resolution mit der zweiten, also insgesamt vier Möglichkeiten. Versuchen wir erst das erste Literal wegzuresolvieren. Die beiden Möglichkeiten (mit der zweiten und der dritten Klausel) ergeben die Unifikationsprobleme:

$$
\begin{array}{c}
\{y_{\mathsf{l\ddot{u}gt\_l\ddot{o}we}} = \mathsf{gestern}(x_{\mathsf{tag}}), z_{\mathsf{wahr\_l\ddot{o}we}} = x_{\mathsf{tag}}\} \\
\{y_{\mathsf{wahr\_l\ddot{o}we}} = \mathsf{gestern}(x_{\mathsf{tag}}), z_{\mathsf{l\ddot{u}gt\_l\ddot{o}we}} = x_{\mathsf{tag}}\}
\end{array}
$$

Das erste Problem hat die eindeutige Lösung:

$$[x_{\mathsf{tag}} \leftarrow v_{\mathsf{Do}}, z_{\mathsf{wahr\_l\ddot{o}we}} \leftarrow v_{\mathsf{Do}}, y_{\mathsf{l\ddot{u}gt\_l\ddot{o}we}} \leftarrow \mathsf{gestern}(v_{\mathsf{Do}})].$$

Das zweite Problem hat eine Lösung (mit $x_{\mathsf{tag}} \leftarrow v_{\mathsf{Mo}}$), die in die Irre führt. Die Resolvente zur Lösung des ersten Problems lautet dann:

$$\neg\mathsf{sagt}(\mathsf{Einhorn}, \mathsf{gestern}(v_{\mathsf{Do}}), v_{\mathsf{Do}})$$

Dies ergibt nach Resolution mit der siebten Klausel die leere Klausel. Der Donnerstag ist also der Tag, an dem sowohl der Löwe als auch das Einhorn sagen, daß sie am Tag vorher gelogen haben.
Das Ergebnis ergibt sich im sortierten Fall aus zwei Resolventen und auch der Suchraum ist relativ klein (initial drei mögliche Resolventen, von denen eine direkt zur Lösung führt). ■

Kapitel 7

Ausblick

In der Vorlesung sind wir auf einige Bereiche nicht eingegangen. Zwei wichtige wollen wir im folgenden kurz vorstellen.

7.1 Sortengenerierung

In Kapitel 3 haben wir gesehen, wie man aus einer sortierten Formulierung eine widerlegungsäquivalente unsortierte Formulierung finden kann. In diesem Abschnitt wollen wir uns der umgekehrten Frage zuwenden und sehen, wie man in einer unsortierten beziehungsweise nur teilweise sortierten Formulierung aus einstelligen Prädikatssymbolen Sortensymbole machen kann und dadurch die Formulierung für Beweise zu vereinfachen. Dieser Teil ist direkt aus der Dissertation von Manfred Schmidt-Schauß übernommen [SS88] und nicht in [SS89] zu finden. Da die Generierung von Sorten aus der initialen Problemstellung insoweit eingeschränkt ist, als daß nur einstellige Prädikate in Sorten konvertiert werden können, die initial in einer bestimmten syntaktischen Gestalt vorkommen, stellen wir die Generierung auf der Klauselebene vor, so daß sie auch durchgeführt werden kann, nachdem eine Reihe von Ableitungsschritten durchgeführt wurde, die ihre Anwendung erst ermöglichen.
Die Sortengenerierung geschieht dadurch, daß man die folgenden Regeln anwendet, wobei SPE die Beziehung zwischen Sorten und Prädikaten beinhaltet, und letztendlich darüber Buch führt welche Sorten konvertiert wurden.
Die Regeln sind wie folgt gegeben (wobei der **if**-Teil der Regeln nur hinreichend getestet werden muß. Es darf also durchaus passieren, daß Fälle durchschlüpfen, da ja die Regeln **Res** und **Fak** immer noch zur Verfügung stehen.):

BT1	Einführung von Sorten
if	CS enthält ein einstelliges Prädikat P mit Definitionssorte S_{DP} und $\exists x_{S_{DP}}.\, P(x)$ ist aus CS ableitbar und P ist noch nicht in eine Sorte überführt worden.
then	füge neues Sortensymbol S_P zur Signatur füge neue Konstante c der Sorte S_P zur Signatur füge die Untersortenbeziehung $S_P \sqsubseteq S_{DP}$ hinzu füge $P \Leftrightarrow S_P$ zu SPE.

BT2	Hinzufügung neuer Termdeklarationen zu Σ
if	CS enthält die Klausel $C = \{P(t)\}$ wobei t keine Variable ist, mit $P \Leftrightarrow S_P$ in SPE.
then	füge Termdeklaration $t \lessdot S_P$ zu **TD**.

BT3	Hinzufügung neuer Untersortendeklarationen
if	CS enthält die Klausel $C = \{P(x)\}$ wobei x die Sorte $\mathfrak{s}(x)$ hat mit $P \Leftrightarrow S_P$ in SPE.
then	füge Untersortendeklaration $\mathfrak{s}(x) \sqsubseteq S_P$ zu **SD**.

BT4	Sortenveränderung von Variablen in Klauseln
if	CS enthält die Klausel $C = \{-P(x)\} \cup A$ wobei x die Sorte $\mathfrak{s}(x)$ hat mit $P \Leftrightarrow S_P$ in SPE und $S_P \cap \mathfrak{s}(x) = T$.
then	lösche das Literal $-P(x)$ aus C ersetze x in A durch eine neue Variable der Sorte T.

Neben weiteren Regeln zur Vereinfachung der Problemstellung, die zum einen Redundanzen löschen, zum anderen unter gewissen Bedingungen Schnittsorten einführen, werden diese Regeln angewendet bis eine der folgenden Terminierungsbedingungen erfüllt ist:

- Wenn $P \Leftrightarrow S_P$ und $-P \Leftrightarrow S_{-P}$ in SPE sind und S_P und S_{-P} eine gemeinsame Untersorte haben, dann ist die Spezifikation widersprüchlich.
- Wenn die Klauselmenge leer und SPE leer ist, dann war die ursprüngliche Klauselmenge erfüllbar. (Diese Bedingung schlägt natürlich nur zu, wenn Löschungen von Redundanzen vorgenommen werden können.)
- Wenn die Klauselmenge die leere Klausel enthält, ist eine Widerlegung gefunden.

7.2 Dynamische Sorten

Zu dem vorgestellten Sortenmechanismus gibt es eine Menge von Erweiterungen, die die Ausdrucksmächtigkeit der Sortensprache erhöhen. Diese betreffen das Offenlassen der Sorteninformation für Variable in der Quantifizierung und Festlegung durch das Vorkommen in Termen oder auch den Abschluß der Sortenstruktur unter Schnitten (vergleiche zum Beispiel [Coh87]). Diese Erweiterungen sind insoweit einander ähnlich als die Sortenstruktur a priori gegeben ist. Dies ist im Fall der *dynamischen Sorten* nicht mehr der Fall. Wir können im folgenden auf die dynamischen Sorten nicht vertieft eingehen, sondern werden nur die Idee kurz vorstellen. Ausführliche Informationen findet man in [Wei89, Wei91].
Im bisherigen Ansatz sind die Termdeklarationen und Untersortenbeziehungen fest in die Signatur eingebaut. Im Fall der dynamischen Logik werden Termdeklarationen der Form $t \lessdot S$ als atomare Formeln aufgefaßt, es sind also Formeln möglich wie $t \lessdot S \wedge P(a)$. Dies hat mehrere Konsequenzen:

- Die Ausdruckskraft der Logik erhöht sich durch entsprechende Ausdrücke stark. Während bislang nur gesagt werden konnte, daß ein Term eine bestimmte Sorte hat, ist es nun auch möglich zu sagen, daß ein Term eine bestimmte Sorte *nicht* hat, das heißt es ist möglich zu sagen $-1 \not\lessdot \mathbb{N}$. Dies erlaubt es Sorten in bestimmten Umfang zu definieren, wie $\mathbb{R}^*$ als die reellen Zahlen ohne Null durch die Formel: $\forall x.\, x \lessdot \mathbb{R}^* \Leftrightarrow x \lessdot \mathbb{R} \wedge x \neq 0$. Weiterhin ist in diesem Ansatz die Behandlung von leeren Sorten einfach zu handhaben, man kann zum Beispiel sagen $\forall x.\, x \not\lessdot S$.

- Der Unifikationsalgorithmus wird nun komplizierter, denn die Sorteninformationen für Terme sind nicht absolut gegeben, sondern im allgemeinen nur modulo einem Constraint. Dies hat mehrere Konsequenzen. Zum einen kann es sein, daß zu einem bestimmten Zeitpunkt in der Deduktion zwei Terme t_1 und t_2 nicht unifizierbar sind (weil eine bestimmte Sorteninformation wie $t_1 \lessdot S$ fehlt), es aber in einem späteren Zustand werden, da nun die Sorteninformation vorliegt. Weiterhin treten im allgemeinen Residuen bei der Resolution auf, das heißt, wenn man zwei komplementäre Literale wegschneidet, so geschieht dies modulo einer Bedingung an die Unifizierbarkeit. Um die leeren Sorten korrekt abhandeln zu können, reicht es nicht zwei komplementäre Klauseln der Art $\{P(x_S)\}$ und $\{\neg P(x_S)\}$ zu finden, sondern man muß auch noch zeigen, daß die Sorte S nichtleer ist.

Besondere Schwierigkeiten bereitet in diesem Fall die Frage der Vollständigkeit des Kalküls. Dafür ist es insbesondere wichtig, festzulegen, welche Sorteninformation man dem Unifikationsalgorithmus zu welcher Zeit zur Verfügung stellt.

Literaturverzeichnis

[AB70] R. Anderson und W.W. Bledsoe. A linear format for resolution with merging and a new technique for establishing completeness. *Journal of the ACM*, **17**:525–534, 1970.

[AINP90] Peter B. Andrews, Sunil Issar, Dan Nesmith und Frank Pfenning. The TPS theorem proving system. In Mark E. Stickel, Herausgeber, *Proceedings of the 10th CADE*, Seiten 641–642, Kaiserslautern, 1990. Springer Verlag, Berlin. LNAI 449.

[And71] Peter B. Andrews. Resolution in type theory. *Journal of Symbolic Logic*, **36**(3), 1971.

[Bar77] Jon Barwise (Herausgeber). *Handbook of Mathematical Logic*, Band 90 in *Studies in Logic and the Foundations of Mathematics*. North-Holland Publishing Company, Amsterdam, Niederlande, 1977.

[BB87] Karl Hans Bläsius und Hans-Jürgen Bürckert, Herausgeber. *Dedukionssysteme – Automatisierung des logischen Denkens*. Oldenbourg, München, 1987.

[Bib82] Wolfgang Bibel. *Automated Theorem Proving*. Vieweg, Wiesbaden, 1982.

[BM79] Robert S. Boyer und J Strother Moore. *A Computational Logic*. Academic Press, New York, USA, 1979.

[Chu36] Alonzo Church. An unsolvable problem of elementary number theory. *American Jounal of Mathematics*, **58**:345–363, 1936.

[Chu40] Alonzo Church. A formulation of the simple theory of types. *Journal of Symbolic Logic*, **5**:56–68, 1940.

[CL73] Chin-Liang Chang und Richard Char-Tung Lee. *Symbolic Logic and Mechanical Theorem Proving*. Academic Press, New York, USA, 1973.

[Coh87] Anthony G. Cohn. A more expressive formulation of many sorted logics. *Journal of Automated Reasoning*, **3**:113–200, 1987.

[Duf91] David Duffy. *Principles of Automated Theorem Proving.* John Wiley & Sons, Chichester, Großbritannien, 1991.

[EFT86] Heinz-Dieter Ebbinghaus, Jörg Flum und Wolfgang Thomas. *Einführung in die mathematische Logik.* Wissenschaftliche Buchgesellschaft, Darmstadt; zweite Auflage, 1986.

[EO87] Norbert Eisinger und Hans Jürgen Ohlbach. Deduktionssysteme – Grundlagen und Beispiele. In Karl Hans Bläsius und Hans-Jürgen Bürckert, Herausgeber, *Dedukionssysteme – Automatisierung des logischen Denkens,* Kapitel II. Oldenbourg, München, 1987.

[Fit90] Melvin Fitting. *First-Order Logic and Automated Theorem Proving.* Springer Verlag, New York, USA, 1990.

[GG89] Dov M. Gabbay und Franz Guenthner. *Handbook of Philosophical Logic,* Band 164–167 in *Studies in Epistemology, Logic, Methodology, and Philosophy of Science.* D. Reidel Publishing Company, Dordrecht, Niederlande, 1983-1989.

[GN87] Michael R. Genesereth und Nils J. Nilsson. *Logical Foundations of Artificial Intelligence.* Morgan Kaufmann, San Mateo, Kalifornien, USA, 1987.

[Göd30] Kurt Gödel. Die Vollständigkeit der Axiome des logischen Funktionenkalküls. *Monatshefte für Mathematik und Physik,* **37**:349–360, 1930.

[Göd31] Kurt Gödel. Über formal unentscheidbare Sätze der Principia Mathematica und verwandter Systeme I. *Monatshefte für Mathematik und Physik,* **38**:173–198, 1931.

[Hen50] Leon Henkin. Completeness in the theory of types. *Journal of Symbolic Logic,* **15**:81–91, 1950.

[Hue73] Gérard P. Huet. A mechanization of type theory. In *Proceedings of the Third International Joint Conference on Artificial Intelligence,* Seiten 139–146, 1973.

[Ker91] Manfred Kerber. How to prove higher order theorems in first order logic. In John Mylopoulos und Ray Reiter, Herausgeber, *Proceedings of the 12th IJCAI,* Seiten 137–142, Sydney, 1991. Morgan Kaufman, San Mateo, California, USA.

[Lov78] D.W. Loveland. *Automated Theorem Proving: A Logical Basis.* North-Holland, New York, USA, 1978.

[Men87] Elliott Mendelson. *Introduction to Mathematical Logic.* Wadsworth & Brooks/Cole, Monterey, Kalifornien, USA; dritte Auflage, 1987.

[Obe62] Arnold Oberschelp. Untersuchungen zur mehrsortigen Quantorenlogik. *Mathematische Annalen*, **145**:297–333, 1962.

[Ric78] Michael M. Richter. *Logikkalküle*. Leitfäden der angewandten Mathematik und Mechanik LAMM. Teubner, Stuttgart, 1978.

[Rob65] John Alan Robinson. A machine oriented logic based on the resolution principle. *Journal of the ACM*, **12**:23–41, 1965.

[Rus08] Bertrand Russell. Mathematical logic as based on the theory of types. *American Journal of Mathematics*, XXX:222–262, 1908.

[Sch38] A. Schmidt. Über deduktive Theorien mit mehreren Sorten von Grunddingen. *Mathematische Annalen*, **115**:485–506, 1938.

[Sko19] Albert Thoralf Skolem. Logisch-kombinatorische Untersuchungen über die Erfüllbarkeit oder Beweisbarkeit mathematischer Sätze. *Skrifter-Vidensk*, **I**:1–36, 1919.

[SS88] Manfred Schmidt-Schauß. *Computational Aspects of an Order-Sorted Logic with Term Declarations*. Dissertation, Fachbereich Informatik, Universität Kaiserslautern, Kaiserslautern, 1988.

[SS89] Manfred Schmidt-Schauß. *Computational Aspects of an Order-Sorted Logic with Term Declarations*. LNAI 395. Springer Verlag, Berlin, 1989.

[Tur37] Alan Turing. On computable numbers, with an application to the Entscheidungsproblem. *Proceedings of the London Mathematical Society, Second Series*, **42**:230–265; **43**:544–546, 1937.

[Wei89] Christoph Weidenbach. A resolution calculus with dynamic sort structures and partial functions. SEKI Report SR-89-23, Fachbereich Informatik, Universität Kaiserslautern, Kaiserslautern, 1989. Kurzversion in ECAI'90, S. 688–693.

[Wei91] Christoph Weidenbach. A sorted logic using dynamic sorts. Technischer Bericht MPI-I-91-218, Max-Planck-Institut für Informatik, Im Stadtwald, Saarbrücken, 1991. Kurzversion in IJCAI'93, S. 60–65.

Index

Sommersemester 1994 FB Informatik, Manfred Kerber

Ausgabe: 14.4.1994

Übungsblatt 1

Sortenlogiken und ihre Automatisierung

Aufgabe 1: Man bringe die Formel
$\forall\varepsilon(\varepsilon > 0 \Rightarrow \exists\delta(\delta > 0 \wedge \forall x\forall y(|x - y| < \delta \Rightarrow |f(x) - f(y)| < \varepsilon)))$ in Klauselnormalform.

Aufgabe 2:

a) Übersetze das folgende Problem in klassische Prädikatenlogik:
Wölfe, Füchse, Vögel, Raupen und Schnecken sind Tiere, und es gibt Individuen von jeder dieser Arten. Außerdem gibt es Körner, und Körner sind Pflanzen. Jedes Tier frißt alle Pflanzen oder alle wesentlich kleineren Tiere, die irgendwelche Pflanzen fressen. Raupen und Schnecken sind wesentlich kleiner als Vögel, diese sind wesentlich kleiner als Füchse, die ihrerseits wesentlich kleiner als Wölfe sind. Wölfe fressen weder Füchse noch Körner, während Vögel zwar Raupen fressen, aber keine Schnecken. Raupen und Schnecken fressen gewisse Pflanzen.

   Theorem: Es gibt ein Tier, das ein körnerfressendes Tier frißt.

b) Bilde die zugehörige Klauselmenge.

c) Beweise das Theorem mit Hilfe des Resolutionskalküls.

Sommersemester 1994 FB Informatik, Manfred Kerber
Ausgabe: 21.4.1994

Übungsblatt 2

Sortenlogiken und ihre Automatisierung

Aufgabe 1: Übersetze das folgende Problem in sortierte Prädikatenlogik:
Wölfe, Füchse, Vögel, Raupen und Schnecken sind Tiere, und es gibt Individuen von jeder dieser Arten. Außerdem gibt es Körner, und Körner sind Pflanzen. Jedes Tier frißt alle Pflanzen oder alle wesentlich kleineren Tiere, die irgendwelche Pflanzen fressen. Raupen und Schnecken sind wesentlich kleiner als Vögel, diese sind wesentlich kleiner als Füchse, die ihrerseits wesentlich kleiner als Wölfe sind. Wölfe fressen weder Füchse noch Körner, während Vögel zwar Raupen fressen, aber keine Schnecken. Raupen und Schnecken fressen gewisse Pflanzen.
Theorem: Es gibt ein Tier, das ein körnerfressendes Tier frißt.

Aufgabe 2: Um welche Art von Signatur handelt es sich bei der folgenden Spezifikation:

a) $\mathbf{SORT} = \{\mathsf{gerade}, \mathsf{ungerade}, \mathbb{N}\}$
$\mathbf{SD} = \{\mathsf{gerade} \sqsubseteq \mathbb{N}, \mathsf{ungerade} \sqsubseteq \mathbb{N}\}$
$\mathbf{TD} = \Big\{0 {<} \mathsf{gerade}, s {<} \mathbb{N} \to \mathbb{N}, s(s(x_{\mathsf{gerade}})) {<} \mathsf{gerade}\Big\}.$

b) $\mathbf{SORT} = \{\mathsf{gerade}, \mathsf{ungerade}, \mathbb{N}\}$
$\mathbf{SD} = \{\mathsf{gerade} \sqsubseteq \mathbb{N}, \mathsf{ungerade} \sqsubseteq \mathbb{N}\}$
$\mathbf{TD} = \{0 {<} \mathsf{gerade}, s {<} \mathbb{N} \to \mathbb{N}, s(s(x_{\mathsf{gerade}})) {<} \mathsf{gerade},$
$+ {<} \mathbb{N} \times \mathbb{N} \to \mathbb{N}, + {<} \mathsf{gerade} \times \mathsf{gerade} \to \mathsf{gerade}, y_{\mathbb{N}} + y_{\mathbb{N}} {<} \mathsf{gerade}\}$

Sommersemester 1994 FB Informatik, Manfred Kerber
Ausgabe: 28.4.1994

Übungsblatt 3

Sortenlogiken und ihre Automatisierung

Aufgabe 1: Wie in Bemerkung 2.9 angegeben, kann man Signaturen auch angeben, ohne eine Untersortendeklarationen zuzulassen. Wie muß man dann die Definition sortierter Terme 2.7 abändern? Suche eine möglichst einfache Lösung.

Aufgabe 2: Gebe ein Modell für die Formelmenge von Aufgabe 1 Übungsblatt 2 an.

Aufgabe 3: Man beweise Lemma 2.10

Sommersemester 1994 FB Informatik, Manfred Kerber
Ausgabe: 6.5.1994

Übungsblatt 4

Sortenlogiken und ihre Automatisierung

Aufgabe 1: Zeige, daß die Relativierungsregeln für den Allquantor und den Existenzquantor mit der Beziehung $\forall = \neg\exists\neg$ verträglich ist, das heißt zeige die Äquivalenz $\hat{\Re}(\forall x_S.\, \varphi) \Leftrightarrow \hat{\Re}(\neg\exists x_S.\, \neg\varphi)$.

Aufgabe 2: Relativiere die Lösung von Übungsblatt 2, Aufgabe 1 und vergleiche das Ergebnis mit der Lösung von Übungsblatt 1, Aufgabe 2.

Aufgabe 3: Sei eine Signatur Σ gegeben mit

$$\begin{aligned}
\mathbf{SORT} &= \{\mathsf{lebewesen}, \mathsf{mensch}, \mathsf{tier}, \mathsf{männlich}, \mathsf{mann}\},\\
\mathbf{SD} &= \{\mathsf{mensch} \sqsubseteq \mathsf{lebewesen}, \mathsf{tier} \sqsubseteq \mathsf{lebewesen}, \mathsf{männlich} \sqsubseteq \mathsf{lebewesen},\\
&\qquad \mathsf{mann} \sqsubseteq \mathsf{männlich}, \mathsf{mann} \sqsubseteq \mathsf{mensch}\},\\
\mathbf{TD} &= \{\mathsf{max} \lessdot \mathsf{mann}, \mathsf{vater} \lessdot \mathsf{lebewesen} \rightarrow \mathsf{männlich}\},\\
\mathbf{PD} &= \{\mathsf{sterblich} \lessdot \mathsf{lebewesen}\}
\end{aligned}$$

Weiterhin haben wir die Formelmenge

$$\begin{aligned}
\Gamma = \{&\forall m_{\mathsf{mensch}}.\, \mathsf{sterblich}(m),\\
&\neg\mathsf{sterblich}(\mathsf{vater}(\mathsf{max})),\\
&\exists m_{\mathsf{mann}}.\, \mathsf{sterblich}(m)\}
\end{aligned}$$

Relativiere die Formelmenge Γ.

Sommersemester 1994 FB Informatik, Manfred Kerber
Ausgabe: 19.5.1994

Übungsblatt 5

Sortenlogiken und ihre Automatisierung

Aufgabe 1: Relativiere die Lösung von Übungsblatt 2, Aufgabe 1 mithilfe der Termrelativierung und beweise das termrelativierte Theorem mithilfe des (unsortierten) Resolutionskalküls. Vergleiche das Ergebnis mit dem ursprünglichen Resolutionsbeweis.

Aufgabe 2: Sei eine Signatur Σ gegeben mit

$$\begin{aligned}
\mathbf{SORT} &= \{\mathsf{lebewesen}, \mathsf{mensch}, \mathsf{pflanze}, \mathsf{mann}\},\\
\mathbf{SD} &= \{\mathsf{mensch} \sqsubseteq \mathsf{lebewesen}, \mathsf{pflanze} \sqsubseteq \mathsf{lebewesen}, \mathsf{mann} \sqsubseteq \mathsf{mensch}\},\\
\mathbf{TD} &= \{\mathsf{max} \lessdot \mathsf{mann}, \mathsf{vater} \lessdot \mathsf{mensch} \to \mathsf{mann}\},\\
\mathbf{PD} &= \{\mathsf{sterblich} \lessdot \mathsf{lebewesen}\}
\end{aligned}$$

Weiterhin haben wir die Formelmenge

$$\begin{aligned}
\Gamma = \{&\forall m_{\mathsf{mensch}}.\, \mathsf{sterblich}(m),\\
&\neg\mathsf{sterblich}(\mathsf{vater}(\mathsf{max})),\\
&\exists m_{\mathsf{mann}}.\, \mathsf{sterblich}(m)\}
\end{aligned}$$

Bilde die Termrelativierung $\Re^{\mathrm{T}}$ der Formelmenge Γ. Ist $\Re^{\mathrm{T}}(\Gamma)$ erfüllbar?

Aufgabe 3: Man zeige im ersten Teil des Beweises des Sortentheorems für die Termrelativierung per vollständiger Induktion über den Termaufbau,

$$\tilde{\mathcal{I}}_{\tilde{\xi}} \circ \Re^{\mathrm{T}} = \mathcal{I}_{\xi}$$

Sommersemester 1994 FB Informatik, Manfred Kerber
Ausgabe: 26.5.1994

Übungsblatt 6

Sortenlogiken und ihre Automatisierung

Aufgabe 1: Gegeben sei eine Signatur mit **SORT** $= \{\mathbb{N}, \mathbb{Z}\}$, **SD** $= \{\mathbb{N} \sqsubseteq \mathbb{Z}\}$, **TD** $= \{0 {<} \mathbb{N}; s {<} \mathbb{N} \to \mathbb{N}; + {<} \mathbb{Z} \times \mathbb{Z} \to \mathbb{Z}\}$ und **PD** $= \{\doteq {<} \mathbb{Z} \times \mathbb{Z}\}$ sowie die folgende Formelmenge $\Gamma = \{\forall x_{\mathbb{Z}}.\, \forall y_{\mathbb{Z}}.\, x + y \doteq y + x; \forall x_{\mathbb{N}}.\, 0 + x \doteq x; \neg s(0) + 0 \doteq s(0)\}$. Bilde die Termrelativierung und zeige, daß die termrelativierte Formelmenge unerfüllbar ist.

Aufgabe 2: Bestimme die Ordnung der folgenden Terme der Logik höherer Stufe (beziehungsweise der Formeln der Logik höherer Stufe):

- x_ι
- $f_{(\iota\to\iota)\to\iota}(x_{(\iota\to\iota)})$
- $\forall P^{(\iota\to o)}.\, P(0) \land (\forall n^\iota.\, P(n) \Rightarrow P(n+1)) \Rightarrow \forall n^\iota.\, P(n)$

Aufgabe 3: Welche der folgenden Formeln sind Komprehensionsaxiome?

- $\exists P^o.\, P \Leftrightarrow \forall x^\iota.\, Q(x) \lor R(x)$
- $\forall x^\iota.\, \exists P^{(\iota\to o)}.\, P(x) \Leftrightarrow Q(x) \lor R(x)$
- $\exists x^\iota.\, \forall P^{(\iota\times\iota\to o)}.\, \exists y^\iota.\, P(x,y) \Leftrightarrow Q(x) \lor R(y)$
- $\forall x^\iota.\, \exists P^{(\iota\to o)}.\, P(x) \lor R(x) \Leftrightarrow Q(x)$

Sommersemester 1994 FB Informatik, Manfred Kerber
Ausgabe: 9.6.1994

Übungsblatt 7

Sortenlogiken und ihre Automatisierung

Aufgabe 1: Zeige die Beziehung $\hat{\mathcal{I}}_{\hat{\xi}} \circ \hat{\Theta} = \mathcal{I}_{\xi}$ in Theorem 4.7 durch vollständige Induktion über den Aufbau der Terme und Formeln.

Aufgabe 2: Betrachte die Formelmenge

$$\begin{aligned}\Gamma = \{&\forall P^{(\iota\to o)}\textbf{.}\, \forall x^{\iota}\textbf{.}\, \mathsf{mensch}(x) \wedge P(x) \Rightarrow\ P(\mathsf{vater}(x)) \vee P(\mathsf{mutter}(x)),\\ &\mathsf{mensch}(\mathsf{max}),\\ &\mathit{hat\_Schweißfüße}(\mathsf{max}),\\ &\neg \mathit{hat\_Schweißfüße}(\mathsf{vater}(\mathsf{max})),\\ &\neg \mathit{hat\_Schweißfüße}(\mathsf{mutter}(\mathsf{max}))\}\end{aligned}$$

Man übersetze Γ in sortierte Logik erster Stufe und zeige, daß $\Theta(\Gamma)$ unerfüllbar ist.

Aufgabe 3: Betrachte die Formelmenge

$$\begin{aligned}\Gamma = \{&\forall P^{(\iota\to o)}\textbf{.}\, \forall x^{\iota}\textbf{.}\, \mathsf{mensch}(x) \wedge P(x) \Rightarrow\ P(\mathsf{vater}(x)) \vee P(\mathsf{mutter}(x)),\\ &\mathsf{mensch}(\mathsf{max}),\\ &\mathit{trinkt\_kaffee}(\mathsf{max}) \Rightarrow \mathit{schläft\_schlecht}(\mathsf{max}),\\ &\neg(\mathit{trinkt\_kaffee}(\mathsf{vater}(\mathsf{max})) \Rightarrow \mathit{schläft\_schlecht}(\mathsf{vater}(\mathsf{max}))),\\ &\neg(\mathit{trinkt\_kaffee}(\mathsf{mutter}(\mathsf{max})) \Rightarrow \mathit{schläft\_schlecht}(\mathsf{mutter}(\mathsf{max})))\}\end{aligned}$$

Übersetze Γ mit Θ. Ist $\Theta(\Gamma)$ unerfüllbar? Man zeige die Unerfüllbarkeit von Γ bezüglich der Standardsemantik durch den Übersetzungsansatz. Welches Komprehensionsaxiom ist dazu erforderlich?

Sommersemester 1994 FB Informatik, Manfred Kerber
Ausgabe: 16.6.1994

Übungsblatt 8

Sortenlogiken und ihre Automatisierung

Aufgabe 1: Gegeben sei eine Signatur Σ mit $\bar{\Sigma} = \{a, f\}$, **SORT** $= \{S, T\}$ und **TD** $= \{a{<}S, a{<}T, f{<}S \to S, f{<}T \to T\}$.
Löse das Unifikationsproblem $\Gamma = \{x_S = y_T\}$.

Aufgabe 2: Gegeben sei eine Signatur Σ mit $\bar{\Sigma} = \{0, s, *\}$, **SORT** $= \{\mathbb{N}, \mathbb{Z}, \mathbb{R}\}$, **SD** $= \{\mathbb{N} \sqsubseteq \mathbb{Z}, \mathbb{Z} \sqsubseteq \mathbb{R}\}$ und
TD $= \{0{<}\mathbb{N}, s{<}\mathbb{N} \to \mathbb{N}, s{<}\mathbb{Z} \to \mathbb{Z}, s{<}\mathbb{R} \to \mathbb{R},$
$*{<}\mathbb{N} \times \mathbb{N} \to \mathbb{N}, *{<}\mathbb{Z} \times \mathbb{Z} \to \mathbb{Z}, *{<}\mathbb{R} \times \mathbb{R} \to \mathbb{R}, *(x_{\mathbb{Z}}, x_{\mathbb{Z}}){<}\mathbb{N}\}$.
Löse die folgenden Unifikationsprobleme:

- $\Gamma = \{z_{\mathbb{N}} = *(u_{\mathbb{Z}}, v_{\mathbb{Z}})\}$
- $\Gamma = \{z_{\mathbb{N}} = s(u_{\mathbb{R}})\}$
- $\Gamma = \{*(z_{\mathbb{N}}, w_{\mathbb{R}}) = *(s(1), s(u_{\mathbb{N}}))\}$
- $\Gamma = \{s(z_{\mathbb{N}}) = 0\}$
- $\Gamma = \{*(z_{\mathbb{N}}, w_{\mathbb{R}}) = s(0)\}$

Übungsblatt 9

Sortenlogiken und ihre Automatisierung

Aufgabe 1: Man axiomatisiere und löse das folgende Problem in sortierter Logik. „Als Alice in den Wald des Vergessens kam, vergaß sie nicht alles, sondern nur gewisse Dinge. Sie vergaß oft ihren Namen und am wahrscheinlichsten vergaß sie den Wochentag. Nun waren der Löwe und das Einhorn häufige Besucher dieses Waldes, allerdings handelt es sich um seltsame Geschöpfe. Der Löwe lügt montags, dienstags und mittwochs und sagt an den anderen Wochentagen die Wahrheit, wohingegen das Einhorn donnerstags, freitags und samstags lügt und an den anderen Tagen die Wahrheit sagt. Eines Tages begegnete Alice dem Löwen und dem Einhorn, die sich unter einem Baum ausruhten. Sie sagten zu Alice:

Löwe:	Gestern war einer meiner Lügentage.
Einhorn:	Gestern war einer meiner Lügentage.

Alice, die ein aufgewecktes Mädchen war, konnte aus diesen Angaben den Wochentag herleiten. Welcher war es?“