TECHNISCHE UNIVERSITÄT BERLIN
FACHBEREICH INFORMATIK (20)
Institut für Software und Theoretische Informatik
Otto-Suhr-Allee 18/20
1000 Berlin 10

EINFÜHRUNG IN DIE SYNTAX UND SEMANTIK DER PRÄDIKATENLOGIK

(in Anlehnung an das Buch "Logikkalküle" von Michael Richter)

ausgearbeitet von Peter Padawitz

INHALT:

1. VORBEMERKUNGEN

Ein Kalkül ist eine Menge von Axiomen und Regeln, die man anwendet, um formale Beweise zu führen. Ein solcher Beweis besteht aus einer Folge von Formeln, von denen jede einzelne durch eine syntaktische Transformation aus einigen ihrer Vorgänger in der Folge hervorgeht. Der Kalkül legt fest, welche Transformationen erlaubt sind. Damit setzt er alle Formeln in Beziehung, die in der einen oder anderen Richtung ineinander transformierbar sind. Diese Relation soll nun eine vorgegebene semantische Beziehung zwischen den Formeln wiedergeben, bei Logikkalkülen die Folgerbarkeit: a soll genau dann in b transformierbar sein, wenn aus der Wahrheit von a die Wahrheit von b folgt. In diesem Sinne ist auch jeder Übersetzer, Optimierer oder Interpreter von Programmen oder Datenrepräsentationen ein Kalkül: a soll genau dann in b transformierbar sein, wenn a und b die gleiche Funktion bzw. das gleiche Datum repräsentieren. Die axiomatische Semantikdefinition einer Programmiersprache ist ein Kalkül, mit dem jedes Programm der Sprache zusammen mit seiner Ein/Ausgaberelation erzeugbar sein soll, d.h. ein Programm erfüllt eine vorgegebene Ein/Ausgaberelation, wenn sich Programm und Relation durch umgekehrte Anwendung der Regeln des Kalküls auf Axiome zurückführen lassen. (Mehr darüber in der Lehrveranstaltung "Semantik algorithmischer Sprachen".)

In all diesen Beispielen geht es letztlich darum, durch syntaktische (besser: "beweisartige", engl. proof-theoretical) Formelmanipulationen semantische Beziehungen sichtbar zu machen. Wir werden in diesem Skript Syntax, Semantik und Kalküle der klassischen Logik behandeln, wollen dabei aber soweit wie möglich Konstruktionen verwenden, die auch in den anderen oben genannten Beispielen eine wesentliche Rolle spielen. Von daher bieten sich algebraische Begriffsbildungen an, die Struktureigenschaften und -verträglichkeiten von syntaktischen und semantischen Objekten auf natürliche Weise präzisieren. Einen solchen Zugang hat Michael Richter in seinem Buch "Logikkalküle" gewählt, jedoch ist wegen der Behandlung weiterer Themen wie nichtklassischer Logiken und automatischer Beweisverfahren der Prädikatenlogik nur wenig Raum gewidmet.

In Kapitel 2 formulieren wir die Begriffe Syntax und Semantik in algebraischer Terminologie, um dann in Kapitel 3 und 4 Richters Kapitel über Aussagen- bzw. Prädikatenlogik in erweiterter und einheitlicherer Form darzustellen. Kapitel 5 beschäftigt sich mit dem Teil der Prädikatenlogik, der nur Gleichungen als

formalisierte Aussagen zuläßt und in der Theorie von Datenstrukturen eine ausgezeichnete Rolle spielt (vgl. die Lehrveranstaltung "Algebra für Informatiker").

2. TERME UND ALGEBREN

Gegeben seien eine abzählbare Menge Var von (Individuen-)variablen und eine abzählbare Familie $\Sigma=\{\Sigma_n\}_{n \in \mathbb{N}}$ von Symbolmengen, die Signatur genannt wird. Die Elemente von Σ_0 heißen Konstanten. Für alle $n \in \mathbb{N}$ heißt $\sigma \in \Sigma_n$ Operationssymbol mit der Stelligkeit n.

Variablen und Symbole werden zu Termen (symbolischen Ausdrücken) zusammengesetzt: Die Menge Term der Σ-Terme über Var ist induktiv wie folgt definiert:

2.1 Alle Variablen sind Terme;

2.2 ist σ ein Operationssymbol mit der Stelligkeit n und sind $t_1,\ldots,t_n$ Terme, dann ist auch $\sigma(t_1,\ldots,t_n)$ ein Term.

Aus 2.2 folgt insbesondere, daß alle Konstanten Terme sind.

Eine Menge A zusammen mit einer Funktion $\sigma_A:A^n \longrightarrow A$ für jedes Operationssymbol σ in Σ mit Stelligkeit n heißt Σ-Algebra. A heißt Trägermenge der Algebra, σ_A Interpretation von σ in A. Notationell unterscheiden wir nicht zwischen einer Algebra und ihrer Trägermenge. Abbildungen von Var nach A heißen Variablenbelegungen oder Zustände in A. Die Menge der Zustände in A bezeichnen wir mit A^{Var}.

Durch eine Σ-Algebra A wird die Bedeutung der Operationssymbole von Σ festgelegt. Damit ist auch die Semantik zusammengesetzter Operationen, die syntaktisch gerade durch Terme dargestellt sind, bestimmt. Der induktive Aufbau von Term erlaubt es uns nämlich, jede Variablenbelegung $u \in A^{Var}$ auf Terme fortzusetzen. Diese Fortsetzung ist also eine Funktion $u^*:Term \longrightarrow A$ mit folgender Definition:

2.3 Für alle Variablen x ist $u^*(x) = u(x)$, d.h. die Einschränkung von u^* auf Var, geschrieben $u^* \mid Var$, stimmt mit u überein;

2.4 jeder Term, der keine Variable ist, hat die Form $\sigma(t_1,\ldots,t_n)$, wobei σ ein Operationssymbol ist und $t_1,\ldots,t_n$ Terme sind, und wir definieren

$$u^*(\sigma(t_1,\ldots,t_n)) = \sigma_A(u^*t_1,\ldots,u^*t_n).$$

Aus 2.4 folgt insbesondere $u^*(\sigma) = \sigma_A$ für alle Konstanten σ.

Term läßt sich selbst zu einer Σ-Algebra erweitern: Für alle $\sigma \in \Sigma$ mit Stelligkeit n ist $\sigma_{Term}:Term^n \longrightarrow Term$ definiert durch

$$\sigma_{Term}(t_1,\ldots,t_n) = \sigma(t_1,\ldots,t_n).$$

Durch Induktion über den Aufbau von Term weist man nach, daß u^* die einzige Funktion f von Term nach A ist, für die

2.5 $$f|Var = u$$

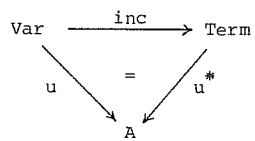
und

2.6 $f(\sigma_{Term}(t_1,\ldots,t_n)) = \sigma_A(ft_1,\ldots,ft_n)$
für alle $\sigma \in \Sigma$ mit Stelligkeit n
gilt.

2.6 besagt, daß die Anwendung der Abbildung f mit der Anwendung von σ vertauschbar ist: f ist ein Σ-Homomorphismus von Term nach A.

Zusammenfassend ist u^* also charakterisiert als der eindeutige Σ-Homomorphismus f von Term nach A, dessen Komposition mit inc, der Einbettung (inclusion) von Var in Term, mit u übereinstimmt:

2.7



Als Umkehrung der Konstruktion von u^* erhalten wir demnach für jeden Σ-Homomorphismus f:Term ⟶ A

2.8 $$(f|Var)^* = f.$$

Wir werden die im Funktionsdiagramm 2.7 dargestellte Beziehung zwischen Syntax (Σ) und Semantik (A) für logische Ausdrücke und ihre Bedeutung formulieren. Sind Σ und A aber z.B. Syntax und Semantik einer Programmiersprache oder eines Datentyps, dann ordnet u^* jedem Programm der Sprache bzw. jedem Objekt des Datentyps dessen Bedeutung zu. Mit diesen Interpretationen von 2.7 beschäftigen sich die Lehrveranstaltungen "Semantik algorithmischer Sprachen" bzw. "Algebra für Informatiker" und "Theorie von Datenstrukturen".

3. AUSSAGENLOGIK

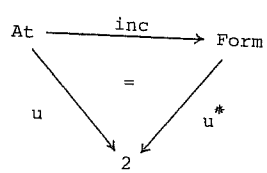
Syntax und Semantik aussagenlogischer Formeln werden in der Terminologie von Kapitel 2 folgendermaßen formuliert:

Die Menge Var der Variablen in aussagenlogischen Formeln bezeichnen wir als Menge At der atomaren Formeln. Die Menge aller aussagenlogischen Formeln heißt Form und ist definiert als die Menge der Σ-Terme über At, wobei die Signatur Σ durch $\Sigma_0 = \{0,1\}$, $\Sigma_1 = \{\neg\}$, $\Sigma_2 = \{\wedge, \vee, \longrightarrow, \longleftrightarrow\}$ und $\Sigma_n = \emptyset$ für alle $n > 2$ gegeben ist.

Aussagenlogische Formeln sind also aus atomaren Formeln, den Konstanten 0 ("falsch"), 1 ("wahr"), dem einstelligen Operationssymbol ¬ (Negation) und den zweistelligen Operationssymbolen ∧ (Konjunktion), ∨ (Disjunktion), ⟶ (Implikation) sowie ⟷ (logische Äquivalenz) aufgebaute symbolische Ausdrücke. Die Elemente von Σ werden auch als Junktoren bezeichnet.

Da wir uns hier mit zweiwertiger Logik befassen und eine aussagenlogische Formel a nur wahr oder falsch "sein" kann, wollen wir als Bedeutung von a nur die Symbole 0 oder 1 zulassen. Die Bedeutung von a hängt ab von der Bedeutung der atomaren Formeln, die in a vorkommen, und den "Wahrheitstafeln", durch die jeder einzelne Junktor als Operation auf der Menge $\{0,1\}$ interpretiert wird. Die Wahrheitstafeln machen also die Menge $\{0,1\}$ zu einer Σ-Algebra, die wir fortan mit $\underline{2}$ bezeichnen werden. Die Semantik aussagenlogischer Formeln ist demnach durch den folgenden Spezialfall von 2.7 definiert:
Bei Vorgabe einer Funktion u:At ⟶ $\underline{2}$ erhält die Formel a als Bedeutung das Bild von a unter der Σ-homomorphen Fortsetzung u^* von u:

3.1



3.2 Definition

Eine Funktion u:At ⟶ $\underline{2}$ heißt Modell von a ∈ Form, wenn $u^*(a) = 1$ gilt. Die Menge der Modelle von a bezeichnen wir mit Mod(a). a heißt erfüllbar, wenn $Mod(a) \neq \emptyset$.

Modell und Erfüllbarkeit sind für Formelmengen entsprechend definiert. Beachte, daß jede Funktion u:At $\longrightarrow$ 2 ein Modell der leeren Formelmenge ist.
Eine Formel a ist aus einer Formelmenge F folgerbar, geschrieben $F \models a$, wenn jedes Modell von F auch Modell von a ist.

Der Begriff der Folgerbarkeit faßt die Aufgabe mathematischer Beweise zusammen, die darin besteht, aus Voraussetzungen (Axiomen) Theoreme zu gewinnen, die in allen Umgebungen (Variablenbelegungen) gelten, in denen auch die Axiome erfüllt sind. Der Nachweis der Folgerbarkeit eines Theorems aus seinen Voraussetzungen spiegelt jedoch i.a. nicht den mathematischen Beweis dieses Theorems wider. Die Folgerung einer Aussage a aus einer Axiomenmenge F erfordert ja zunächst einmal das Auffinden sämtlicher Modelle von F. Wie man leicht sieht, gibt es 2^n mögliche Modelle von F, wobei n die Anzahl aller in Formeln von F vorkommenden Atome ist. Folgern hat demnach exponentiellen Aufwand. Bei $n = \infty$ läßt sich ein Beweis durch Folgern nicht mehr führen.

Demgegenüber erhält man mathematische Beweise i.a. ohne den semantischen Umweg über die Modelle der Axiome durch direktes Ableiten aus den Axiomen. Jeder Schritt einer Ableitung besteht in der Anwendung einer Regel R, die die Form

$$\frac{a_1,\ldots,a_n}{a},$$

hat, wobei $a, a_1, \ldots, a_n$ Formeln sind. $a_1,\ldots,a_n$ heißen Prämissen, a Konklusion von R. Zu einem (Ableitungs-)Kalkül gehören eine Menge von Formeln (Axiome) und eine endliche Menge von Regeln. Eine Ableitung einer Formel a aus einer Formelmenge F in einem Kalkül K ist eine Folge f von Formeln, in der jedes Folgenglied b entweder zu F gehört oder ein Axiom von K ist oder Konklusion einer Regel von K ist, deren Prämissen Vorgänger von b in f sind.

3.3 Definition

Der Hilbertkalkül für die Ableitung aussagenlogischer Formeln, kurz: Aussagenkalkül, besteht aus folgenden Axiomen und Regeln:

1) Boolesche Axiome:

$(a \wedge b) \leftrightarrow (b \wedge a)$ (Kommutativität von $\wedge$)

$(a \wedge (b \wedge c)) \leftrightarrow ((a \wedge b) \wedge c)$ (Assoziativität von $\wedge$)

$((a \wedge b) \vee b) \leftrightarrow b$ (Absorption)

$(a \wedge (b \vee c)) \leftrightarrow ((a \wedge b) \vee (a \wedge c))$ (Distributivität von $\wedge$)

$(a \wedge \neg a) \leftrightarrow 0$ ($\wedge$-Inversität von $\neg$)

$(a \vee 0) \leftrightarrow a$ ($\vee$-Neutralität der Null)

entsprechende Axiome für $\wedge$, $\vee$ bzw. 1 statt $\vee$, $\wedge$ bzw. 0

$(a \longrightarrow b) \leftrightarrow (\neg a \vee b)$ ($\longrightarrow$ - Eliminierung)

$(a \leftrightarrow b) \leftrightarrow ((a \longrightarrow b) \wedge (b \longrightarrow a))$ ($\leftrightarrow$ -Eliminierung)

2) Kongruenzaxiome:

$a \leftrightarrow a$ (Reflexivität von $\leftrightarrow$)

$(a \leftrightarrow b) \longrightarrow (b \leftrightarrow a)$ (Symmetrie von $\leftrightarrow$)

$((a \leftrightarrow b) \wedge (b \leftrightarrow c)) \longrightarrow (a \leftrightarrow c)$ (Transitivität von $\leftrightarrow$)

$(a \leftrightarrow b) \longrightarrow (\neg a \leftrightarrow \neg b)$ (Verträglichkeit von $\leftrightarrow$ mit $\neg$)

$((a \leftrightarrow b) \wedge (a' \leftrightarrow b')) \longrightarrow ((a \vee a') \leftrightarrow (b \vee b'))$

(Verträglichkeit von $\leftrightarrow$ mit $\vee$)

entsprechende Axiome für die Verträglichkeit von $\leftrightarrow$ mit $\wedge$, $\longrightarrow$ und $\leftrightarrow$

3) $a \longrightarrow (b \longrightarrow (a \wedge b))$ ($\wedge$-Einführung)

$(a \wedge b) \longrightarrow a$ ($\wedge$-Eliminierung)

4) $\dfrac{a,\ a \longrightarrow b}{b}$ (Modus ponens)

Charakteristisch für den Hilbertkalkül ist der Modus ponens als einzige Regel. Als Axiome haben wir gewählt: 1. logische Äquivalenzen, die den Gleichungen einer Booleschen Algebra entsprechen, 2. die Kongruenzeigenschaft der logischen Äquivalenz und 3. Axiome zur Einführung bzw. Eliminierung des Junktors $\wedge$. Dieses Ableitungssystem ist nicht minimal, es erlaubt uns aber die direkte Verwendung bekannter aussagenlogischer Gesetze beim Ableiten und erleichtert darüber hinaus das Verständnis des Zusammenhangs zwischen Hilbertkalkül und Folgerbarkeit, den wir im folgenden untersuchen wollen.

3.4 Definition

Seien $F \subseteq$ Form and $a \in$ Form. a heißt aus F ableitbar, geschrieben $F \vdash a$, wenn es eine Ableitung von a aus F im Aussagenkalkül von 3.3 gibt.

Als erstes stellt sich die Frage nach der Korrektheit des Aussagenkalküls: Sind alle aus einer Formelmenge F ableitbaren Formeln auch aus F folgerbar? Diese Frage beantwortet man durch Induktion über die (minimale) Länge einer Ableitung von F nach a. Das heißt, wir müssen zeigen:

1) Alle Formeln von F and alle Axiome aus 3.3.1 - 3.3.3 sind aus F folgerbar;

2) sind a and a $\longrightarrow$ b aus F folgerbar, dann ist auch b aus F folgerbar (Korrektheit des Modus ponens).

3.5 Satz

Der Aussagenkalkül ist korrekt.

Beweis:

1) $Mod(F) \subseteq Mod(F)$ impliziert $F \models a$ für alle $a \in F$.
Für alle Σ-Homomorphismen $h:Form \longrightarrow \underline{2}$ und alle a, b $\in$ Form gilt $h(a \longleftrightarrow b) = 1$ genau dann, wenn $h(a) = h(b)$. Da $\underline{2}$ eine Boolesche Algebra ist, gilt $h(a \wedge b) = h(a) \wedge_{\underline{2}} h(b) = h(b) \wedge_{\underline{2}} h(a) = h(b \wedge a)$ also $h((a \wedge b) \longleftrightarrow (b \wedge a)) = 1$. Insbesondere ist demnach für alle $u \in Mod(F)$ $u^*((a \wedge b) \longleftrightarrow (b \wedge a)) = 1$, d.h. das Kommutativitätsaxiom aus 3.3.1 ist aus F folgerbar. Entsprechend gilt die Folgerbarkeit der anderen Booleschen Axiome aus 3.3.1. Analog erhält man für jeden Σ-Homomorphismus $h:Form \longrightarrow \underline{2}$ und jedes Axiom a aus 3.3.2 und 3.3.3 $h(a) = 1$, was wie oben $F \models a$ impliziert.

2) Es gelte $F \models a$, $F \models (a \longrightarrow b)$ und sei $u \in Mod(F)$. Dann ist $u^*(a) = 1$ und $u^*(a) \longrightarrow_{\underline{2}} u^*(b) = u^*(a \longrightarrow b) = 1$, also $u^*(b) = 1$. Daraus folgt $F \models b$.

Ist der Aussagenkalkül mächtig genug, um in ihm alle folgerbaren Formeln ableiten zu können? Dies ist die Frage nach der Vollständigkeit eines Kalküls. Sie ist i.a. schwerer zu beantworten als die umgekehrte Frage nach der Korrektheit, weil die Menge der folgerbaren Formeln ja gerade nicht "konstruktiv" definiert ist, so daß sich Aussagen über alle folgerbaren Formeln nicht durch Induktion über ein Konstruktionsschema beweisen lassen.

Wir verwenden algebraische Methoden, um die Vollständigkeit des Aussagenkalküls zu zeigen: Zunächst wird für jede Formelmenge F eine dem Ableitungsoperator $\vdash$ entsprechende Kongruenzrelation $\sim_F$ auf Form gebildet und anschließend für jede nicht aus F ableitbare Formel a ein Σ-Homomorphismus h angegeben, der von Form über die Quotientenalgebra $Form/\sim_F$ nach $\underline{2}$ führt und alle Formeln von F auf die

Eins, aber a auf die Null abbildet. Daraus folgt, daß $h|At$ ein Modell von F, aber nicht von a ist, a also nicht aus F folgerbar ist.

3.6 Lemma

Sei $F \subseteq Form$ und $\sim_F$ die durch

$$a \sim_F b \text{ gdw } F \vdash (a \longleftrightarrow b)$$

definierte Relation auf Form: Zwei Formeln a und b heißen genau dann F-äquivalent, wenn die logische Äquivalenz von a und b aus F ableitbar ist.

1) $a \sim_F 1$ gilt genau dann, wenn a aus F ableitbar ist.

2) $\sim_F$ ist eine Kongruenzrelation.

3) $Form/\sim_F$ ist eine Boolesche Algebra.

Beweis:

1) Durch Anwendung von Axiomen und Regeln des Aussagenkalküls erhält man

$$F \vdash (a \longleftrightarrow 1)$$

gdw $F \vdash (a \longrightarrow 1)$ und $F \vdash (1 \longrightarrow a)$

gdw $F \vdash 1$ und $F \vdash a$

gdw $F \vdash a$.

2) erhält man mit dem Modus ponens aus den Kongruenzaxiomen des Aussagenkalküls und aus 3.3.3.

3) $Form/\sim_F$ ist eine Boolesche Algebra, wenn alle Gleichungen einer Booleschen Algebra in $Form/\sim_F$ gelten. So bedeutet z.B. die Kommutativität von $\wedge$ in $Form/\sim_F$, daß für alle a,b $\in$ Form $[a] \wedge_\sim [b] = [b] \wedge_\sim [a]$ ist, wobei $[a]$ die Äquivalenzklasse von a bzgl. $\sim_F$ und $\wedge_\sim$ die Interpretation von $\wedge$ in $Form/\sim_F$ ist. Wegen $[a] \wedge_\sim [b] = [a \wedge b]$ ist die Gleichung

$$[a] \wedge_\sim [b] = [b] \wedge_\sim [a]$$

gleichbedeutend mit

$$(a \wedge b) \sim_F (b \wedge a),$$

also auch mit

$$F \vdash ((a \wedge b) \longleftrightarrow (b \wedge a)).$$

Eine Ableitung dieser logischen Äquivalenz existiert trivialerweise, weil $(a \wedge b) \longleftrightarrow (b \wedge a)$ ein Axiom des Aussagenkalküls ist.
Da wir in dieser Weise alle Gleichungen einer Booleschen Algebra als Axiome des Aussagenkalküls formuliert haben, ist $Form/\sim_F$ eine Boolesche Algebra.

3.7 Satz

Der Aussagenkalkül ist vollständig.

Beweis:

Sei $F \subseteq Form$ und $a \in Form$ mit $F \nvdash a$.

Nach Lemma 3.6.1 ist $a \not\sim_F 1$ und $b \sim_F 1$ für alle $b \in F$. Die Funktion

$$g: Form/\sim_F \longrightarrow \underline{2}$$

definiert durch

$$g([c]) = \begin{cases} 0 & \text{falls } c \sim_F (c \wedge a) \\ 1 & \text{sonst} \end{cases}$$

ist ein Σ-Homomorphismus: Da sich alle Operationen einer Booleschen Algebra als Zusammensetzungen von $\neg$ und $\wedge$ darstellen lassen, genügt es, die Verträglichkeit von g mit $\neg$ und $\wedge$ zu zeigen.

Sei c, d $\in$ Form. $c \sim_F (c \wedge a)$ und $\neg c \sim_F (\neg c \wedge a)$ impliziert $(\neg c \vee \neg a) \sim_F (\neg c \wedge a)$, also $1 \sim_F ((\neg c \vee \neg a) \vee a) \sim_F ((\neg c \wedge a) \vee a) \sim_F a$ im Widerspruch zu $a \not\sim_F 1$. Im Fall $c \sim_F (c \wedge a)$ gilt also $\neg c \not\sim_F (\neg c \wedge a)$, so daß

$$g(\neg_\sim [c]) = g([\neg c]) = 1 = \neg_{\underline{2}}\, 0$$
$$= \neg_{\underline{2}}\, g([c]).$$

Wegen $\neg\neg c \sim_F c$ erhält man in analoger Weise die Verträglichkeit von g für den Fall, daß $c \not\sim_F (c \wedge a)$.

$c \sim_F (c \wedge a)$ oder $d \sim_F (d \wedge a)$ impliziert $(c \wedge d) \sim_F (c \wedge d \wedge a)$, also

$$g([c] \wedge_\sim [d]) = g([c \wedge d]) = 0$$
$$= g([c]) \wedge_{\underline{2}} g([d]).$$

$c \not\sim_F (c \wedge a)$ und $d \not\sim_F (d \wedge a)$ impliziert $\neg c \sim_F (\neg c \wedge a)$ und $\neg d \sim_F (\neg d \wedge a)$ (siehe oben).

Daraus folgt $\neg(c \wedge d) \sim_F (\neg c \vee \neg d) \sim_F$
$((\neg c \wedge a) \vee (\neg d \wedge a)) \sim_F (\neg(c \wedge d) \wedge (\neg c \vee a) \wedge (a \vee \neg d) \wedge a) \sim_F$
$(\neg(c \wedge d) \wedge a)$, also $(c \wedge d) \not\sim_F (c \wedge d \wedge a)$, so daß

$$g([c] \wedge_\sim [d]) = g([c \wedge d]) = 1$$
$$= 1 \wedge_{\underline{2}} 1 = g([c]) \wedge_{\underline{2}} g([d]).$$

Die Komposition von g mit der natürlichen Abbildung $nat: Form \longrightarrow Form/\sim_F$ liefert demnach einen Σ-Homomorphismus $h: Form \longrightarrow \underline{2}$, der a auf die Null und

alle Formeln von F auf die Eins abbildet. Damit ist $h|At \in Mod(F) - Mod(a)$, so daß $F \not\models a$.

Eine Formelmenge F heißt konsistent, wenn die Null nicht aus F ableitbar ist. Aus der Vollständigkeit und der Korrektheit des Aussagenkalküls folgt sofort die Äquivalenz von Konsistenz und Erfüllbarkeit (vgl. 3.2):

3.8 Satz

Sei $F \subseteq Form$. $F \vdash 0$ gilt genau dann, wenn $Mod(F) = \emptyset$.

Beweis: Aus $F \nvdash 0$ folgt $F \not\models 0$, d.h. es gibt ein Modell von F, das kein Modell von 0 ist. Also ist $Mod(F) \neq \emptyset$. $F \vdash 0$ impliziert $F \models 0$, d.h. $Mod(F) \subseteq Mod(0)$. Da es aber keine Modelle der Null gibt, ist auch Mod(F) leer.

4. Prädikatenlogik

Die Syntax der Prädikatenlogik erweitert in zweierlei Hinsicht die Syntax der Aussagenlogik. Erstens setzt sich die Menge der atomaren Formeln aus prädikativen Ausdrücken zusammen und nicht mehr aus unstrukturierten Booleschen Variablen. Zweitens werden Existenz- und Allquantoren als zusätzliche einstellige Operationssymbole eingeführt. Ihr Auftreten in einer Formel a bewirkt die Bindung von Individuenvariablen in atomaren Teilformeln von a, so daß die Semantik von a Existenz- bzw. Allaussagen über bestimmte Individuenbereiche enthält.

Zum Aufbau der Atomformeln gehen wir aus von einer abzählbaren Menge Var von Individuenvariablen und zwei Signaturen $\mathcal{O}$ und $\mathcal{P}$. Die Elemente von $\mathcal{O}$ nennen wir wie in Kapitel 2 Operationssymbole, die Elemente von $\mathcal{P}$ bezeichnen wir als Prädikatensymbole. Term sei die Menge der $\mathcal{O}$-Terme über Var. Die Menge At der prädikatenlogischen Atomformeln besteht dann aus allen Ausdrücken der Form $P(t_1,\ldots,t_n)$ mit $P \in \mathcal{P}_n$ und $t_1,\ldots,t_n \in$ Term.

Ein Paar $S = (A,\mathcal{R})$ heißt $(\mathcal{O},\mathcal{P})$-Struktur, wenn A eine $\mathcal{O}$-Algebra und $\mathcal{R}$ eine Menge von Relationen ist derart, daß für alle $n \in \mathbb{N}$ und alle $P \in \mathcal{P}_n$ eine n-stellige Relation P_S auf A zu $\mathcal{R}$ gehört. Die Trägermenge von A wird auch Individuenbereich von S genannt.

Im folgenden seien $\mathcal{O}$ und $\mathcal{P}$ fest gewählt. Wir schreiben deshalb für $(\mathcal{O},\mathcal{P})$-Struktur kurz Struktur.

Bei gegebener Belegung der Individuenvariablen bestimmt S die Semantik der Atomformeln, d.h. S induziert eine Abbildung

$$v_S : \mathrm{At} \longrightarrow [A^{\mathrm{Var}} \longrightarrow \underline{2}]$$

die durch

$$v_S(P(t_1,\ldots,t_n))(u) = P_S(u^* t_1,\ldots,u^* t_n)$$

definiert ist. Damit haben wir den Anschluß an die Aussagenlogik hergestellt, wo eine Bewertung der Atomformeln durch eine Funktion $u : \mathrm{At} \longrightarrow \underline{2}$ gegeben war. Dieser Funktion entspricht jetzt die Abbildung v_S.

Neben dem Aufbau der Atomformeln hatten wir als Charakteristikum der Prädikatenlogik den Gebrauch von Quantoren als logische Operationssymbole festgestellt. Dies bedeutet eine Erweiterung der Signatur Σ aus Kapitel 3 um je zwei einstellige Operationssymbole $\forall x$ und $\exists x$ für alle Individuenvariablen x. Die so erweiterte Signatur bezeichnen wir wieder mit Σ.

Die Menge Form der prädikatenlogischen Formeln ist definiert als die Menge der Σ-Terme über At. (Man beachte die Zweistufigkeit des Formelaufbaus: At besteht aus prädikativen Ausdrücken mit Argumenten aus Term, der Menge der $\mathcal{O}$-Terme über Var; Form besteht aus Σ-Termen über At.)

Analog zur Aussagenlogik (vgl. 3.1) soll die Semantik einer prädikatenlogischen Formel a als Bild von A unter der Fortsetzung von v_S auf Form definiert werden. Dazu müssen wir die Menge $\underline{A} := [A^{\mathrm{Var}} \longrightarrow \underline{2}]$ der Funktionen vonVariablenbelegungen in Wahrheitswerte zu einer Σ-Algebra machen. Die Junktoren haben wir in $\underline{2}$ durch Wahrheitstafeln interpretiert. Diese Interpretation setzt sich auf $\underline{A}$ folgendermaßen fort: Für alle n-stelligen Junktoren σ, alle $f_1,\ldots,f_n \in \underline{A}$ und alle $u \in A^{\mathrm{Var}}$ sei

$$\sigma_{\underline{A}}(f_1,\ldots,f_n)(u) = \sigma_{\underline{2}}(f_1 u,\ldots,f_n u).$$

Sei $u \in A^{\mathrm{Var}}$, $x \in$ Var und $a \in A$. Für die Variablenbelegung, die x den Wert a zuweist und sonst mit u übereinstimmt, schreiben wir u(a/x) („a für x in u").

$\forall x$ und $\exists x$ werden in $\underline{A}$ interpretiert durch

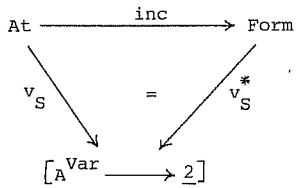
$$(\forall x)_{\underline{A}}(f)(u) = \begin{cases} 1 & \text{falls } fu' = 1 \text{ für alle } u' \in A^{\mathrm{Var}} \\ & \text{mit } u' = u(u'x/x) \\ 0 & \text{sonst} \end{cases}$$

bzw.

$$(\exists x)_{\underline{A}}(f)(u) = \begin{cases} 1 & \text{falls } u' \in A^{\mathrm{Var}} \text{ mit } fu' = 1 \\ & \text{und } u' = u(u'x/x) \text{ existiert} \\ 0 & \text{sonst} \end{cases}$$

Die Beziehung zwischen Syntax und Semantik prädikatenlogischer Formeln läßt sich demnach wie in der Aussagenlogik durch ein Funktionsdiagramm darstellen (vgl. 3.1) :

4.1



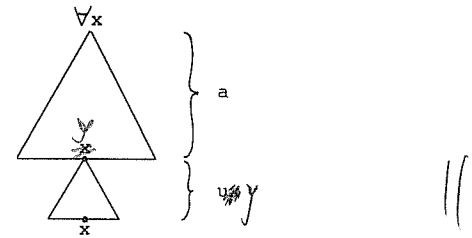
S ist ein Modell von a ∈ Form, wenn $v_S^*(a) = 1$ gilt. Den Begriff "folgerbar" übernehmen wir aus der Aussagenlogik (vgl. 3.2), jedoch ist eine prädikatenlogische Formel a bereits dann erfüllbar, wenn es eine Struktur S = (A, $\mathcal{R}$)und eine Variablenbelegung u in A mit $v_S^*(a)(u) = 1$ gibt.

Der Prädikatenkalkül

Der Aussagenkalkül (3.3) soll zu einem vollständigen Ableitungskalkül für prädikatenlogische Formeln erweitert werden. Dazu müssen wir ihn um Axiome und/oder Regeln zur Manipulation von quantifizierten Formeln ergänzen. Bei gegebener Struktur S = (A, $\mathcal{R}$) ist mit der oben definierten Semantik eine Formel $\forall x a$ genau dann wahr, wenn a für alle Belegungen von x in A wahr ist. Um diese Bedeutung von $\forall x a$ im Ableitungskalkül zu "simulieren", verwenden wir den Substitutionsoperator

$$sub : Form \longrightarrow [Term^{Var} \longrightarrow Form],$$

der, angewendet auf $\forall x a$, für jede Belegung $u \in Term^{Var}$ in allen atomaren Teilformeln von a die Variable x durch ux ersetzt. Vor der eigentlichen Substitution nimmt sub eine von u abhängige Umbenennung von x vor. Damit wird verhindert, daß ein Auftreten von x in ux durch $\forall x$ "gebunden" wird:



4.2 Definition

Sei a ∈ Form. Das Auftreten einer Variablen x in einer atomaren Teilformel b von a heißt frei in a, wenn es keine Teilformel $\forall xc$ oder $\exists xc$ von a gibt, die b enthält. fr(a) ist die Menge aller ausschließlich frei in a vorkommenden Variablen. Die Menge aller in a vorkommenden Variablen einschließlich der Variablenkomponente von Quantoren bezeichnen wir mit Var(a).

id sei die Identität auf Var.
Die Funktionen

$$rep, tut, sub : Form \longrightarrow [Term^{Var} \longrightarrow Form],$$

die Replacement-, Umbenennungs- bzw. Substitutionsoperator heißen, sind induktiv über dem Formelaufbau folgendermaßen definiert:

$$rep(P(t_1,\ldots,t_n))(u) = P(u^* t_1,\ldots,u^* t_n)$$

für alle $P(t_1,\ldots,t_n) \in At$ und $u \in Term^{Var}$,

$$rep(\sigma(a_1,\ldots,a_n))(u) = \sigma(rep(a_1)(u),\ldots,rep(a_n)(u))$$

für alle n-stelligen Junktoren σ; $a_1,\ldots,a_n \in$ Form und $u \in Term^{Var}$,

$$rep(Qxa)(u) = Qx(rep(a)(u(x/x)))$$

für $Q \in \{\forall, \exists\}$, alle a ∈ Form und $u \in Term^{Var}$;

$$tut(a) = a \text{ für alle } a \in At,$$

$$tut(\sigma(a_1,\ldots,a_n))(u) = \sigma(tut(a_1)(u),\ldots,tut(a_n)(u))$$

für alle n-stelligen Junktoren σ ; $a_1,\ldots,a_n \in$ Form und $u \in Term^{Var}$,

$$tut(Qxa)(u) = Qy(tut(rep(a)(id(y/x)))(u))$$

für $Q \in \{\forall, \exists\}$, alle a ∈ Form und u ∈ $Term^{Var}$, wobei y das erste Element in der Abzählung aller Individuenvariablen ist, das weder in a noch in u(fr(Qya)) vorkommt;

$$sub(a)(u) = rep(tut(a)(u))(u)$$

für alle a ∈ Form und u ∈ $Term^{Var}$.

a ∈ Form heißt mit u ∈ $Term^{Var}$ verträglich, wenn für jede Teilformel Qxb von a mit $Q \in \{\forall, \exists\}$ x nicht in u(fr(Qxb)) vorkommt.

Im folgenden Lemma wird gezeigt, daß der Umbenennungsoperator tut Formeln mit Variablenbelegungen verträglich macht.

4.3 Lemma

Für alle a ∈ Form und u ∈ $Term^{Var}$ gilt:

1) tut(a)(u) ist mit u verträglich,

2) fr(tut(a)(u)) = fr(a) .

Beweis:

Wir beschränken uns auf den Induktionsschritt von a nach Qxa, wobei $Q \in \{\forall, \exists\}$. Der Rest ist trivial.

1) O.B.d.A. sei $Q = \forall$. Alle Teilformeln $\exists$zb von tut(Qxa)(u) sind Teilformeln von tut(rep(a)(id(y/x)))(u), wobei y wie in 4.2 gewählt ist. Nach Induktionsvoraussetzung ist diese Formel mit u veträglich. Demzufolge ist auch tut(Qxa)(u) mit u verträglich.
Für alle Teilformeln $\forall$zb von tut(Qxa)(u) gilt entweder z=y und b=tut(rep(a)(id(y/x)))(u) oder $\forall$zb ist eine Teilformel von tut(rep(a)(id(y/x)))(u).
Im ersten Fall tritt z weder in a noch in u(fr($\forall$za)) auf. Nach 2) gilt dann fr($\forall$za)- {x} = fr($\forall$xa) = fr(tut($\forall$xa)(u)) = fr($\forall$zb).
Folglich kommt z in u(fr($\forall$zb)) nicht vor.

Im zweiten Fall kommt nach Induktionsvoraussetzung z in u(fr($\forall$zb)) nicht vor.

2) Nach Induktionsvoraussetzung gilt
fr(tut(Qxa)(u)) = fr(tut(rep(a)(id(y/x)))(u))- {y}
= fr (rep(a)(id(y/x)))- {y} = fr(a)- {x} = fr(Qxa).

4.4 Definition

Der Hilbertkalkül für die Ableitung prädikatenlogischer Formeln, kurz: Prädikatenkalkül, besteht aus dem Aussagenkalkül und folgenden Axiomen und Regeln (mit t ∈ Term und u ∈ $Term^{Var}$):

1) sub-Axiome:
sub(a)(id) ⟶ a
$\forall$xa ⟶ sub(a)(id(t/x))
sub(a)(id(t/x)) ⟶ $\exists$xa

2) Quantorregeln:

$$\frac{a}{sub(a)(u)}$$

$$\frac{a \longrightarrow sub(b)(id(y/x))}{a \longrightarrow \forall xb} \qquad y \notin fr(a) \cup fr(b)$$

$$\frac{sub(a)(id(y/x)) \longrightarrow b}{\exists xa \longrightarrow b} \qquad y \notin fr(a) \cup fr(b)$$

Zum Nachweis der Korrektheit des Prädikatenkalküls benötigen wir das folgende Lemma, in dem der Zusammenhang zwischen Termsubstitutionen und "semantischen" Variablenbelegungen hergestellt wird.

4.5 Lemma

Sei $S = (A, \mathcal{R})$ eine Struktur, $v = v_S^*$, $a \in \text{Form}$, $u \in \text{Term}^{Var}$ und $\bar{u} \in A^{Var}$.

1) $v(sub(a)(u))(\bar{u}) = v(a)(\bar{u}^* u)$.

2) $v(tut(a)(u)) = v(a)$.

3) Falls a mit u verträglich ist, gilt
$v(rep(a)(u))(\bar{u}) = v(a)(\bar{u}^* u)$.

Beweis:

1) erhält man aus 2), 4.3.1 und 3):

$v(sub(a)(u))(\bar{u}) = v(rep(tut(a)(u))(u))(\bar{u}) = v(tut(a)(u))(\bar{u}^* u) =$
$v(a)(\bar{u}^* u)$.

Bei 2) und 3) beschränken wir uns wie in Lemma 4.3 auf den Induktionsschritt von a nach Qxa. (Bei 3) lautet der Induktionsanfang:

$v(rep(P(t_1,\ldots,t_n))(u))(\bar{u}) = v(P(u^*t_1,\ldots,u^*t_n))(\bar{u})$

$= P_S(\bar{u}^*u^*t_1,\ldots,\bar{u}^*u^*t_n) = P_S((\bar{u}^*u)^*t_1,\ldots,(\bar{u}^*u)^*t_n)$

$= v(P(t_1,\ldots,t_n))(\bar{u}^*u)$.)

Sei y wie in Definition 4.2. Wir zeigen zunächst, daß a mit id(y/x) verträglich ist: Sei Qzb eine Teilformel von a. Dann gilt

$Var(id(y/x)(fr(Qzb))) = id(y/x)(fr(b) - \{z\})$

$\subseteq (fr(b) - \{z\}) \cup \{y\}$, d.h., wenn z in id(y/x)(fr(Qzb)) vorkommt, dann ist z = y. Da y in a nicht auftritt, kann Qzb keine Teilformel von a sein. Wegen 3) erhalten wir für alle $\bar{u} \in A^{Var}$

$v(rep(a)(id(y/x)))(\bar{u}) = v(a)(\bar{u}^* id(y/x))$. (*)

2) Nach Induktionsvoraussetzung und (*) gilt

$v(tut(Qxa)(u))(\bar{u})$

$= v(Qy(tut(rep(a)(id(y/x)))(u)))(\bar{u})$

$= \min\{ v(tut(rep(a)(id(y/x)))(u)))(w) \mid w = \bar{u}(wy/y) \}$

$= \min\{ v(rep(a)(id(y/x)))(w) \mid w = \bar{u}(wy/y) \}$

$= \min\{ v(a)(w^* id(y/x)) \mid w = \bar{u}(y/y) \}$

$= \min\{ v(a)(w) \mid w = \bar{u}(wx/x) \}$

$= v(Qxa)(\bar{u})$.

3) Sei Qxa mit u verträglich. Dann kommt x in u(fr(Qxa)) nicht vor. Folglich gibt es für alle $w \in A^{Var}$, die mit $\bar{u}$ nur in x nicht übereinstimmen, ein $\bar{w} \in A^{Var}$, das mit $\bar{u}^*u$ nur in x nicht übereinstimmt, derart, daß $\bar{w}$ auf allen freien Variablen von a mit $w^*u(x/x)$ zusammenfällt: Wähle $\bar{w}x = wx$ und $\bar{w}z = \bar{u}^*uz$ für alle $z \neq x$. Dann gilt $\bar{w}x = w^*u(x/x)x$ und $\bar{w}z = w^*uz = w^*u(x/x)z$ für alle $z \in fr(a) - \{x\}$, weil x in uz nicht vorkommt. Umgekehrt gibt es für alle $\bar{w} \in A^{Var}$ mit $\bar{w} = \bar{u}^*u(\bar{w}x/x)$ ein $w \in A^{Var}$ mit $w = \bar{u}(wx/x)$ derart, daß $w^*u(x/x)$ auf fr(a) mit $\bar{w}$ übereinstimmt: Wähle $wx = \bar{w}x$ und $wz = \bar{u}z$ für alle $z \neq x$. Dann gilt $w^*u(x/x)x = \bar{w}x$ und $w^*u(x/x)z = w^*uz = \bar{u}^*uz = \bar{w}z$ für alle $z \in fr(a) - \{x\}$, weil x in uz nicht vorkommt.

Damit erhält man nach Induktionsvoraussetzung: $v(rep(Qxa)(u))(\bar{u}) =$

$= v(Qx(rep(a)(u(x/x))))(\bar{u})$

$= \min\{ v(rep(a)(u(x/x)))(w) \mid w = \bar{u}(wx/x) \}$

$= \min\{ v(a)(w^*u(x/x)) \mid w = \bar{u}(wx/x) \}$

$= \min\{ v(a)(\bar{w}) \mid \bar{w} = \bar{u}^*u(\bar{w}x/x) \}$

$= v(a)(\bar{u}^*u)$.

4.6 Satz

Der Prädikatenkalkül ist korrekt.

Beweis:

Sei $S = (A, \mathcal{R})$ eine Struktur, $v = v_S^*$, $a,b \in \text{Form}$, $u \in A^{Var}$ und $t \in \text{Term}$.

Nach Lemma 4.5 gilt $v(sub(a)(id))(u) = v(a)(u^*id) = v(a)(u)$. Daraus folgt, daß S ein Modell des sub-Axioms

$$sub(a)(id) \longrightarrow a$$

ist.

Da $u^*id(t/x)$ mit u bis auf x übereinstimmt, gilt

$v(sub(a)(id(t/x)))(u) = v(a)(u^*id(t/x)) = 1$, wenn $v(\forall xa)(u) = 1$ ist.

Also ist S ein Modell von

$$\forall xa \longrightarrow sub(a)(id(t/x)).$$

Die Korrektheit des dritten sub-Axioms zeigt man analog.

Ist S ein Modell von a, dann gilt für alle $\bar{u} \in A^{Var}$ $v(a)(\bar{u}) = 1$.
Damit ist auch für alle $u \in Term^{Var}$ $v(sub(a)(u)(\bar{u}) = v(a)(\bar{u}^{*}u) = 1$.
Das beweist die Korrektheit der ersten Quantorregel.

Die Korrektheit der zweiten Quantorregel (und analog der dritten) erhält man wie folgt: Wir nehmen an, daß S kein Modell für die Konklusion $a \longrightarrow \forall xb$ ist. Dann gibt es $u \in A^{Var}$ mit $v(a)(u) = 1$ und $v(\forall xb)\, u) = 0$. Folglich existiert $\bar{u} \in A^{Var}$ mit $\bar{u} = u(\bar{u}x/x)$ und $v(b)(\bar{u}) = 0$. Sei $w \in A^{Var}$ definiert durch $wy = \bar{u}x$ und $wz = uz$ für $z \neq y$. w zeigt, daß S kein Modell der Prämisse

$$a \longrightarrow sub(b)(id(y/x))$$

ist: Wegen $y \notin fr(a) \cup fr(b)$ erhält man $v(a)(w) = v(a)(u) = 1$, aber $v(sub(b)(id(y/x)))(w) = v(b)(w^{*}id(y/x)) = v(b)(\bar{u}) = 0$.

Da auch der im Prädikatenkalkül enthaltene Aussagenkalkül korrekt ist (3.5), sind wir fertig.

Vollständigkeit des Prädikatenkalküls

Um zu zeigen, daß alle folgerbaren prädikatenlogischen Formeln im Prädikatenkalkül ableitbar sind, greifen wir zurück auf den Beweis der Vollständigkeit der Aussagenlogik (3.6., 3.7). Die im Lemma 3.6 formulierte Charakterisierung des Ableitungsoperators $\vdash$ als Kongruenzrelation $\sim_F$ auf Form gilt auch für den Prädikatenkalkül, weil dieser den Aussagenkalkül enthält.
Satz 3.7 stützte sich auf eine junktorenverträgliche Abbildung

$$g: Form/\sim_F \longrightarrow \underline{2}\ ,$$

die jetzt etwas anders definiert werden muß, damit sie später zu einer junktoren- und quantorenverträglichen Funktion

$$h : Form \longrightarrow [Term^{Var} \longrightarrow \underline{2}]$$

erweitert werden kann (vgl. die Definition von h im Beweis von 3.7).
Das folgende Lemma enthält die Schritte, über die wir zu h gelangen.

4.7 Lemma

Sei $F \subseteq Form$, $a \in Form$ und $[a]$ die Äquivalenzklasse von a bzgl. $\sim_F$.

1) $[\forall xa] = \inf \{ [sub(a)(id(t/x))] \mid t \in Term \}$,
$[\exists xa] = \sup \{ [sub(a)(id(t/x))] \mid t \in Term \}$.

2) Tarski's Lemma:
Sei B eine Boolesche Algebra und $\{X_n\}_{n \in \mathbb{N}}$ eine abzählbare Familie von Teilmengen von B derart, daß für alle $n \in \mathbb{N}$ $\inf X_n$ und $\sup X_n$ existiert. Dann gibt es für alle $x \in B$ mit $x \neq 1$ eine junktorenverträgliche Abbildung $g:B \longrightarrow \underline{2}$ mit $gx = 0$ und

$$g(\inf X_n) = \min \{ gy \mid y \in X_n \},$$
$$g(\sup X_n) = \max \{ gy \mid y \in X_n \} \qquad (*)$$

für alle $n \in \mathbb{N}$.

3) Für alle $n \in \mathbb{N}$ sei X_n von der Form $\{ [sub(a)(w)] \mid w = u(wx/x) \}$.
(Da Form, Var und $Term^{Var}$ abzählbar sind, gibt es nur abzählbar viele Mengen dieser Form.) Sei g eine junktorenverträgliche Funktion von $Form/\sim_F$ nach $\underline{2}$ mit $(*)$ und sei $K = (Term, \mathcal{R})$ die durch

$$P_K(t_1,\ldots,t_n) = g([P(t_1,\ldots,t_n)])$$

definierte kanonische Struktur bzgl. F und a.
Dann ist $v_K^{*}(a)(u) = g([sub(a)(u)])$.

Beweis:

1) Mit Hilfe des Aussagenkalküls zeigt man leicht, daß $[a] \leq [b]$ genau dann gilt, wenn $a \longrightarrow b$ aus F ableitbar ist. Aus dem zweiten sub-Axiom folgt, daß $[\forall xa]$ eine untere Schranke aller Äquivalenzklassen $[sub(a)(id(t/x))]$ mit $t \in Term$ ist. Andererseits liefert die zweite Quantorregel für alle solchen unteren Schranken $[b]$, daß $[b] \leq [\forall xa]$. Also ist $[\forall xa]$ die größte untere Schranke.

Die zweite Gleichung von 1) zeigt man analog.
Einen Beweis von Tarski's Lemma erhält man aus den Sätzen 1.5.7 und 1.5.11 bei Richter.

3) Wir zeigen die Behauptung durch Induktion über den Aufbau von a.

Für $a = P(t_1,\dots,t_n) \in At$ gilt $v_K^*(a)(u) = v_K(a)(u) =$
$P_K(u^*t_1,\dots,u^*t_n) = g([P(u^*t_1,\dots,u^*t_n)])$

Da $g([sub(-)(u)])$ eine junktorenverträgliche Funktion von Form nach $\underline{2}$ ist, können wir uns auf den Induktionsschritt von a nach Qxa beschränken. O.B.d.A. sei $Q = \forall$.

$g([sub(\forall xa)(u)])$

$= g([rep(tut(\forall xa)(u))(u)])$

$= g([rep(\forall yb)(u)])$ mit y wie in 4.2 und $b = tut(rep(a)(id(y/x)))(u)$

$= g([\forall y(rep(b)(u(y/y)))])$

$= g(inf\{[sub(rep(b)(u(y/y)))(id(t/x))] \mid t \in Term\}$ nach 1)

$= min\{g([sub(rep(b)(u(y/y)))(id(t/x))]) \mid t \in Term\}$ nach 2)

nach Voraussetzung

$= min\{v_K^*(rep(b)(u(y/y)))(id(t/x)) \mid t \in Term\}$

nach Induktionsvoraussetzung

$= min\{v_K^*(b)(id(t/x)^*u(y/y)) \mid t \in Term\}$

nach 4.5.1, weil b mit u(y/y) verträglich ist: Nach 4.3.1 ist b mit u verträglich.

Sei Qzc eine Teilformel von b. Falls z=y, gilt
u(fr(Qzc)) = u(y/y)(fr(Qzc)), so daß z in u(y/y)(fr(Qzc)) nicht vorkommt.

Falls $z \neq y$, gilt
$z \notin Var(u(fr(Qzc))) \cup \{y\} \supseteq Var(u(y/y)(fr(Qzc)))$.

$= min\{v_K^*(b)(w) \mid w = id^*u(wy/y)\}$

analog zum vorletzten Gleichungsschritt im Beweis von 4.5.3, denn aus $y \notin Var(u(fr(\forall ya)))$ und $fr(b) - \{y\} =$
$= fr(rep(a)(id(y/x))) - \{y\} \subseteq fr(a) - \{y\}$ folgt
$y \notin Var(u(fr(\forall yb)))$

$= min\{v_K^*(rep(a)(id(y/x)))(w) \mid w = u(wy/y)\}$

nach 4.5.2

$= min\{v_K^*(a)(w^*id(y/x)) \mid w = u(wy/y)\}$

nach 4.5.3, weil a mit id(y/x) verträglich ist (vgl.Beweis von 4.5.)

$= min\{v_K^*(a)(w) \mid w = u(wx/x)\}$

$= v_K^*(\forall xa)(u)$.

4.8 Satz

Der Prädikatenkalkül ist vollständig.

Beweis:

Sei $F \subseteq Form$ und $a \in Form$ mit $F \nvdash a$. Das erste sub-Axiom des Prädikatenkalküls und der Modus ponens implizieren $F \nvdash sub(a)(id)$. Die erste Quantorregel liefert $F \vdash sub(b)(u)$ für alle $b \in F$ und $u \in Term^{Var}$. Da der Prädikatenkalkül den Aussagenkalkül enthält, folgt $sub(a)(id) \not\sim_F 1$ und $sub(b)(u) \sim_F 1$ für alle $b \in F$ und $u \in Term^{Var}$. nach 3.6.4

Nach 4.7.2 gibt es eine junktorenverträgliche Abbildung
$g : Form/\sim_F \longrightarrow \underline{2}$ mit $g([sub(a)(id)]) = 0$ und

$g(inf\{[sub(c)(w)] \mid w = u(wx/x)\}) = min\{g([sub(c)(w)]) \mid w = u(wx/x)\}$

$g(sup\{[sub(c)(w)] \mid w = u(wx/x)\}) = max\{g([sub(c)(w)]) \mid w = u(wx/x)\}$

für alle $c \in Form$, $x \in Var$ und $u \in Term^{Var}$.

Nach 4.7.3 gilt für die kanonische Struktur K bzgl. F und a

$$v_K^*(c)(u) = g([sub(c)(u)])$$

für alle $c \in Form$ und $u \in Term^{Var}$. Daraus folgt
$v_K^*(a)(id) = 0$ und $v_K^*(b)(u) = 1$ für alle $b \in F$ und $u \in Term^{Var}$.
Demnach ist K ein Modell von F, aber nicht von a. Also gilt $F \not\models a$.

5. GLEICHUNGSLOGIK

In diesem Kapitel betrachten wir Strukturen $(A,\mathcal{R})$, in denen ein zweistelliges Prädikatensymbol von $\mathcal{P}$, das EQ heißen möge, als die Gleichheit auf A interpretiert wird. Solche Strukturen nennen wir Gleichheitsstrukturen. Dementsprechend heißen Gleichheitsstrukturen, die eine prädikatenlogische Formel a erfüllen, Gleichheitsmodelle von a, und $F \models_{EQ} a$ bedeutet, daß alle Gleichheitsmodelle von F auch a erfüllen.

5.1 Definition

Der Gleichheitskalkül besteht aus dem Prädikatenkalkül und folgenden Axiomen ($x,x_1,\ldots,x_n,y_1,\ldots,y_n$ sind Individuenvariablen):

$EQ(x,x)$ (Reflexivität von EQ)

$(EQ(x_1,y_1) \wedge \ldots \wedge EQ(x_n,y_n)) \longrightarrow EQ(f(x_1,\ldots,x_n),f(y_1,\ldots,y_n))$
für alle $f \in \mathcal{O}$ (Verträglichkeit von EQ mit f)

$(EQ(x_1,y_1) \wedge \ldots \wedge EQ(x_n,y_n)) \longrightarrow EQ(P(x_1,\ldots,x_n),P(y_1,\ldots,y_n))$
für alle $P \in \mathcal{P}$ (Verträglichkeit von EQ mit P)

Den Ableitungsoperator des Gleichheitskalküls bezeichnen wir mit $\vdash_{EQ}$.

Die Korrektheit des Gleichheitskalküls, d.h. "$F \vdash_{EQ} a$ impliziert $F \models_{EQ} a$" folgt sofort aus der Tatsache, daß die Gleichheit eine reflexive, $\mathcal{O}$- und $\mathcal{P}$-verträgliche Relation ist. Die Adäquatheit des Gleichheitskalküls für Folgerungen in Gleichheitsstrukturen zeigt der folgende Satz:

5.2 Satz

Der Gleichheitskalkül ist vollständig.

Beweis: Sei $F \not\vdash_{EQ} a$. Dann gilt: F ∪ EQ-Axiome $\not\vdash$ a. Da der Prädikatenkalkül vollständig ist (4.8), gibt es eine Struktur $S=(A,\mathcal{R})$, die F und die EQ-Axiome, aber nicht a erfüllt. Demnach existiert $u \in A^{Var}$ mit $v_S^*(a)(u) = 0$ und $v_S^*(b)(w) = 1$ für alle $b \in F$ und $w \in A^{Var}$. Da S die EQ-Axiome erfüllt, ist EQ_S eine Kongruenzrelation auf A (Symmetrie und Transitivität erhält man mit P = EQ im dritten EQ-Axiom). Folglich ist $S' = (A/EQ_S, \mathcal{R}')$ mit

$$P_{S'}([a_1],\ldots,[a_n]) = P_S(a_1,\ldots,a_n)$$

eine Gleichheitsstruktur: Die Wohldefiniertheit von $P_{S'}$ ergibt sich wie folgt:

Falls $[a_1] = [b_1],\ldots,[a_n] = [b_n]$, dann gibt es $u \in A^{Var}$ so, daß für alle $1 \leq i \leq n$ $v_S(EQ(x_i,y_i))(u) = EQ_S(ux_i,uy_i) = EQ_S(a_i,b_i) = 1$ gilt. Da S die EQ-Axiome erfüllt, folgt $P_S(a_1,\ldots,a_n) = P_S(ux_1,\ldots,ux_n) = v_S(P(x_1,\ldots,x_n))(u) = v_S(P(y_1,\ldots,y_n))(u) = P_S(uy_1,\ldots,uy_n) = P_S(b_1,\ldots,b_n)$.
Sei nat die natürliche Abbildung von A nach A/EQ_S. Für alle $u \in A^{Var}$ und $P(t_1,\ldots,t_n) \in At$ gilt also $v_S(P(t_1,\ldots,t_n))(u) = P_S(u^*t_1,\ldots,u^*t_n) = P_{S'}([u^*t_1],\ldots,[u^*t_n]) = P_{S'}((nat \circ u)^*t_1,\ldots,(nat \circ u)^*t_n) = v_{S'}(P(t_1,\ldots,t_n))(nat \circ u)$. Da v_S und $v_{S'}$ Σ-Homomorphismen sind, folgt $v_S^*(c)(u) = v_{S'}^*(c)(nat \circ u)$ für alle $c \in Form$, so daß man $v_{S'}^*(a)(nat \circ u) = 0$ und $v_{S'}^*(b)(nat \circ u) = 1$ für alle $b \in F$ erhält. Demnach ist S' ein Gleichheitsmodell von F, aber nicht von a, d.h. $F \not\models_{EQ} a$.

Eine atomare Formel der Form EQ(t,t') heißt Gleichung. Sei E eine Menge von Gleichungen. Die $\mathcal{O}$-Algebra A eines Gleichheitsmodells $S = (A,\mathcal{R})$ von E nennen wir ($\mathcal{O}$,E)-Algebra. Demnach ist eine $\mathcal{O}$-Algebra A genau dann eine ($\mathcal{O}$,E)-Algebra, wenn für alle $EQ(t,t') \in E$ und alle $u \in A^{Var}$ $u^*t = u^*t'$ gilt:
Für jedes Gleichheitsmodell $S = (A,\mathcal{R})$ von E erhält man nämlich

$$v_S^*(EQ(t,t'))(u) = EQ_S(u^*t,u^*t) = 1.$$

In der Lehrveranstaltung "Algebra für Informatiker" wird die Methode des Spezifizierens abstrakter Datentypen mit Hilfe von Gleichungen behandelt. Aus einer solchen Spezifikation lassen sich dann in dem folgenden Kalkül weitere als Gleichungen formulierte Eigenschaften des spezifizierten Datentyps ableiten.

5.3 Definition

Sei E eine Menge von Gleichungen. Der Gleichungskalkül für E besteht aus folgenden Axiomen und Regeln ($t,t',t_1,\ldots,t_n,t_1',\ldots,t_n'$ sind $\mathcal{O}$-Terme über Var; u ist aus $Term^{Var}$):

1) Die Axiome sind alle Gleichungen von E und die Formel EQ(t,t) (Reflexivität von EQ).

2) Substitutivitätsregel:

$$\frac{EQ(t,t')}{EQ(u^*t,u^*t')}$$

3) Kongruenzregeln:

$$\frac{EQ(t,t')}{EQ(t',t)}$$ (Symmetrie von EQ)

$$\frac{EQ(t_1,t_2),EQ(t_2,t_3)}{EQ(t_1,t_3)}$$ (Transitivität von EQ)

$$\frac{EQ(t_1,t_1'),\ldots,EQ(t_n,t_n')}{EQ(f(t_1,\ldots,t_n),f(t_1',\ldots,t_n'))}$$

für alle $f \in \mathcal{O}$ (Verträglichkeit von EQ mit f)

Für jede im Gleichungskalkül für E ableitbare Formel EQ(t,t') schreiben wir $t \equiv_E t'$.

5.4 Satz

Sei E eine Menge von Gleichungen. Der Gleichungskalkül für E ist korrekt, d.h. für alle $(\mathcal{O},E)$-Algebren A und alle $u \in A^{Var}$ gilt:

$t \equiv_E t'$ impliziert $u^*t = u^*t'$.

Beweis: Sei $E^* = \{(u^*t,u^*t') \mid EQ(t,t') \in E, u \in Term^{Var}\}$.
$\equiv_E$ ist die kleinste Kongruenzrelation auf Term, die E^* enthält. Andererseits ist auch die Menge aller $(t,t') \in Term^2$, für die für alle $w \in A^{Var}$ and alle $(\mathcal{O},E)$-Algebren A $w^*t = w^*t'$ gilt, eine Kongruenzrelation auf Term, die E^* enthält: Für $(u^*t,u^*t') \in E^*$ und $w \in A^{Var}$ erhält man nämlich
$w^*u^*t = (w^*u)^*t = (w^*u)^*t' = w^*u^*t'$, weil A die Gleichung EQ(t,t') erfüllt.

Für Kenner der "Algebra für Informatiker" sei bemerkt, daß Satz 5.4 für mehrsortige Signaturen $\mathcal{O}$ nicht gilt. Die Korrektheit bzgl. mehrsortiger Algebren erfordert einen modifizierten Gleichungskalkül. Dieser ist in der Arbeit "Completeness of many-sorted equational logic" von J.A. Goguen und J. Meseguer angegeben (SRI Report, 1981).
Die Vollständigkeit des Gleichungskalküls 5.3 beweisen wir im folgenden Satz:

5.5 Satz

Sei E eine Menge von Gleichungen. Der Gleichungskalkül für E ist vollständig: Wenn für alle $(\mathcal{O},E)$-Algebren A und alle $u \in A^{Var}$ $u^*t = u^*t'$ gilt, dann ist $t \equiv_E t'$.

Beweis: Da nat:Term $\longrightarrow$ Term/$\equiv_E$ surjektiv ist, gibt es für jede Variablenbelegung u in Term/$\equiv_E$ ein $w \in Term^{Var}$ mit $nat \circ w = u$. Durch Anwendung der

Substitutivitätsregel 5.3.2 erhält man für alle $EQ(t,t') \in E$
$u^*t = (nat \circ w)^*t = [w^*t] = [w^*t'] = (nat \circ w)^*t' = u^*t'$.
Demnach ist Term/$\equiv_E$ eine $(\mathcal{O},E)$-Algebra. Für alle $(\mathcal{O},E)$-Algebren A und alle $u \in A^{Var}$ gelte $u^*t = u^*t'$. Sei inc die Einbettung von Var in Term und [t] die Äquivalenzklasse von t bzgl. $\equiv_E$. Dann folgt
$[t] = nat(t) = (nat \circ inc)^*t = (nat \circ inc)^*t' = nat(t') = [t']$.

Der Beweis von Satz 5.5 zeigt, daß Term/$\equiv_E$ genau die Gleichungen erfüllt, die im Gleichungskalkül für E ableitbar sind. Kenner der "Algebra für Informatiker" weisen wir darauf hin, daß die initiale Semantik $T_{(\mathcal{O},E)}$ der Spezifikation $(\mathcal{O},E)$, nicht Term/$\equiv_E$, sondern deren Einschränkung Term'/$\equiv_E$ auf die Menge Term' aller variablenfreien $\mathcal{O}$-Terme ist. Demzufolge gibt es Gleichungen, die von $T_{(\mathcal{O},E)}$ erfüllt werden, die aber im Gleichungskalkül 5.3 nicht ableitbar sind. Was man zusätzlich braucht, um (Gleichungs-)Theoreme über abstrakte Datentypen zu beweisen, ist ein Schema zur Induktion über variablenfreie Terme. In der Literatur ist die Frage nach einem geeigneten Kalkül zum Ableiten aller in $T_{(\mathcal{O},E)}$ gültigen Gleichungen noch nicht befriedigend gelöst (vgl. hierzu Arbeiten von J.A. Goguen und F. Nourani).

Korrektheit und Vollständigkeit des Gleichheitskalküls 5.1 und des Gleichungskalküls 5.3 liefern uns die Äquivalenz beider Kalküle, was das Ableiten von Gleichungen betrifft: Für alle Mengen E von Gleichungen und alle $\mathcal{O}$-Terme t,t' gilt $E \vdash_{EQ} EQ(t,t')$ genau dann, wenn $t \equiv_E t'$.

In diesem Kapitel haben wir uns mit der Einschränkung prädikatenlogischer Formeln auf Gleichungen beschäftigt. In einem abschließenden Satz kehren wir von der Gleichungslogik zur Aussagenlogik zurück, indem wir als Menge $\mathcal{O}$ der Operationssymbole die Junktoren und als Menge E der Gleichungen die Gleichungen einer Booleschen Algebra wählen.

5.6 Satz

Sei $\mathcal{O}$ die Menge der aussagenlogischen Operationssymbole, E die Menge der Axiome
Gleichungen einer Booleschen Algebra und $F = \emptyset$. Zwei aussagenlogische Formeln ($\mathcal{O}$-Terme) t,t' sind genau dann F-äquivalent (vgl. 3.6), wenn $t \equiv_E t'$.

Beweis:

Eine logische Äquivalenz $t \leftrightarrow t'$ ist genau dann ein Boolesches Axiom aus 3.3.1, wenn $(t,t') \in E^*$ (vgl. Beweis von Satz 5.4). Demnach ist E^* in $\sim_F$ enthalten. Da $\sim_F$ eine Kongruenzrelation auf Term ist, folgt $\equiv_E \subseteq \sim_F$. Umgekehrt bedeutet $t \sim_F t'$, daß $t \leftrightarrow t'$ aus F ableitbar ist. Satz 3.5 (Korrektheit des Aussagenkalküls) liefert $(u^*t \leftrightarrow_{\underline{2}} u^*t') = u^*(t \leftrightarrow t') = 1$ für alle $u \in \underline{2}^{Var}$, weil F leer ist. Nach Definition von $\leftrightarrow_{\underline{2}}$ folgt $u^*t = u^*t'$ für alle $u \in \underline{2}^{Var}$.

Angenommen, es gibt eine Boolesche Algebra A und $w \in A^{Var}$ mit $w^*t \neq w^*t'$. Durch Anwendung Boolescher Gleichungen folgt daraus $(w^*t \leftrightarrow_A w^*t') \neq 1$. Im Beweis von Satz 3.7 (Vollständigkeit der Aussagenlogik) haben wir für ein $[a] \in Form/\sim_F$ mit $[a] \neq 1$ einen Σ-Homomorphismus g von $Form/\sim_F$ nach $\underline{2}$ konstruiert, der $[a]$ auf die Null abbildet. Ganz entsprechend erhält man einen Σ-Homomorphismus $g:A \longrightarrow \underline{2}$ mit $g(w^*t \leftrightarrow_A w^*t') = 0$. Daraus folgt $(gw)^*t = gw^*t \neq gw^*t' = (gw)^*t'$ im Widerspruch dazu, daß für alle $u \in \underline{2}^{Var}$ $u^*t = u^*t'$ gilt.

Demnach ist für alle $(\mathfrak{S},E)$-Algebren A und alle $w \in A^{Var}$ $w^*t = w^*t'$, so daß Satz 5.5 (Vollständigkeit des Gleichungskalküls für E) $t \equiv_E t'$ impliziert.

6. Theorembeweiser (Automatische Beweisverfahren)

In diesem Kapitel betrachten wir einen Kalkül, der zum automatischen (d.h. programmierbaren) Finden von Beweisen von Theoremen benutzt wird.

6.1. Beweisen und Testen

Definition 1:

Eine (mathematische, logische) Theorie T besteht aus einer Menge A von Formeln ohne freie Variable, genannt Axiome von T, in Signaturen σ und π von Operations- und Prädikatensymbolen, genannt die Sprache von T. Ein Satz von T ist eineFormel ohne freie Variable in der Sprache von T, die aus A folgt. (Ein Mathematiker nennt einen Satz, der ihm wichtig, neu oder schwierig erscheint, ein Theorem.) Eine Struktur, in der alle Axiome (und damit alle Sätze) von T wahr sind, heißt Modell von T.

Satz 1 (Kompaktheitssatz):

a ist ein Satz von T genau dann, wenn es endlich viele Axiome von T gibt, aus denen a folgt.

Beweis:

Die Aussage gilt für Ableitbarkeit, also wegen Korrektheit und Vollständigkeit auch für Folgerbarkeit.

Bemerkung:

1) Für Formeln ohne freie Variable gilt:
   $\{a_1,\dots a_n\} \models a \Leftrightarrow a_1 \wedge \dots \wedge a_n \longrightarrow a$ ist allgemeingültig.
2) Es ist also möglich und für das Automatisieren von Verfahren üblich, sich beim Beweisen von Sätzen auf die Untersuchung von Allgemeingültigkeit zu beschränken.

Satz 2 (Aufzählbarkeit)

Die allgemeingültigen Formeln der Prädikatenlogik in irgendeiner Sprache sind effektiv aufzählbar, d.h. es gibt einen Algorithmus, der zu einer Sprache (σ,π) als Eingabe die allgemeingültigen Formeln dieser Sprache in irgendeiner Reihenfolge ausgibt. (Das Ausgeben ist ein unendlicher Prozeß!)

Beweis:

Der Algorithmus erzeugt die Zeichenreihen, über $\sigma \cup \pi \cup \Sigma \cup Var \cup \{$Klammern, Komma$\}$, entscheidet, welche davon Ableitungen sind, und gibt von jeder Ableitung die letzte Formel aus.

Folgerung 1:

Es gibt Testverfahren auf Allgemeingültigkeit, d.h. einen Algorithmus, der zu einer Formel a als Eingabe mit der Anwort "ja" terminiert, falls a allgemeingültig ist, und sonst mit "nein" oder gar nicht terminiert.

Beweis:

Alle allgemeingültigen Formeln aufzählen und mit der Eingabe vergleichen.

Theorem 3 (Unentscheidbarkeit von Theorien):

Die "meisten" Theorien sind unentscheidbar, d.h. es gibt keinen Algorithmus, der für jede Formel der Signatur entscheidet, ob sie ein Satz der Theorie ist (der also zur Eingabe a mit "ja" terminiert, wenn a ein Satz der Theorie ist, und sonst mit "nein"; also immer terminiert).

Beweisidee:

a) Das Halteproblem ist für Programmiersprachen, die nicht zu ausdrucksschwach sind, unentscheidbar. Z.B. ist es nicht entscheidbar, ob ein ALGOL-Programm zu einer Eingabe terminiert. Schade! Oder nicht?
b) In einer Theorie T, die nicht zu ausdrucksschwach ist, kann man Berechnungen beschreiben. D.h. es gibt eine Formel a mit den freien Variablen x und y und zu jedem Programm P und zu jeder Eingabe e Terme t_p und t_e, so daß $sub(a)(\alpha(t_p/x,t_e/y))$ ein Satz von T ist genau dann, wenn P zu e terminiert.
c) Wäre T entscheidbar, wäre das Halteproblem entscheidbar.

Beispiele:

1) Die Peano-Arithmetik ist die Theorie mit der Sprache $\sigma_0=\{0\}$, $\sigma_1=\{s\}$, $\sigma_2=\{+,\cdot\}$, $\pi_2=\{=\}$, $\sigma_i=\emptyset$ für $i>2$, $\pi_i=\emptyset$ für $i\neq 2$ und den Axiomen
   $\forall x\forall y\,[sx=sy \longrightarrow x=y]$, $\forall x\neg[sx=0]$ (Nachfolger),
   $\forall x\,[x+0=x]$, $\forall x\forall y[x+sy=s(x+y)]$ (Addition),
   $\forall x\,[x\cdot 0=0]$, $\forall x\forall y[x\cdot sy=x\cdot y+x]$ (Multiplikation),
   $sub(a)(id(0/x)) \wedge \forall x[a \longrightarrow sub(a)(id(sx/x))] \longrightarrow \forall x a$ für jede Formel a mit x als einziger freier Variable (Induktion).
   Die Peano-Arithmetik ist unentscheidbar.

2) Die Theorie der Worte (Zeichenreihen über einem endlichen Alphabet) mit geeigneter Sprache (Verkettung von Worten) und Axiomen (welche?) ist unentscheidbar.

Folgerung 2 (Unentscheidbarkeit der Prädikatenlogik):

Für die "meisten" Signaturen ist die Allgemeingültigkeit von Formeln unentscheidbar, z.B. für ein zweistelliges Prädikatensymbol oder zwei einstellige Funktionssymbole.

Beweisidee:

Man suche eine unentscheidbare Theorie in der Sprache mit endlich vielen Axiomen.

Folgerung 3:

Das Testverfahren aus Folgerung 1 kann nicht zu einem Entscheidungsverfahren ausgebaut werden.

Hilfssatz:

Sei a eine Formel ohne freie Variable. Dann sind äquivalent:

(1) a ist unerfüllbar (widersprüchlich, widerspruchsvoll);

(2) $\neg a$ ist allgemeingültig; $\Leftrightarrow$ $a \neq 0 \Leftrightarrow a \vdash 0$

(3) $\neg a$ ist ableitbar: $\vdash \neg a$;

(4) a ist widerlegbar, d.h. $a \vdash 0$.

Folgerung 4:

Es gibt Testverfahren (aber keine Entscheidungsverfahren) auf Widersprüchlichkeit. Diese Verfahren versuchen, aus der eingegebenen Formel a einen Widerspruch (0) abzuleiten. Ist a widersprüchlich, so terminieren sie mit "ja", sonst mit "nein" oder gar nicht.

Aufgabe:

Finden Sie eine geeignete Sprache und Axiome für eine Theorie der Worte, so daß Sie alles, was Ihnen wichtig zu sein scheint, darin ausdrücken können. Sind Ihnen Berechnungen wichtig? Warnung: Dies ist eine Hartgummi-Aufgabe.

6.2. Grundresolution

Die Resolutionsmethode ist ein Testverfahren auf Widersprüchlichkeit, das einen Ableitungskalkül wie in Folgerung 4 oben benutzt.

Um einfache Ableitungen zu erhalten, werden Formeln zunächst in eine Normalform gebracht. Wir beginnen in diesem Abschnitt mit quantorenfreien Formeln.

Definition 1:

Ein Literal ist eine atomare Formel mit oder ohne Negationszeichen davor. Eine quantorenfreie Formel ist in konjunktiver Normalform, wenn sie eine Konjunktion von Disjunktionen von Literalen ist:

$$(a_{1,1} \vee \ldots \vee a_{1,n1}) \wedge \ldots \wedge (a_{m,1} \vee \ldots \vee a_{m,nm}),\ a_{i,j} \text{ Literale.}$$

Hilfssatz:

Seien a,b,c Formeln. Es sind allgemeingültig:

$\neg\neg a \longleftrightarrow a$ (doppelte Negation), $\neg(a \vee b) \longleftrightarrow (\neg a \wedge \neg b)$ (de Morgansche Gesetze),

$a \vee a \longleftrightarrow a$ (Idempotenz)

und die Booleschen Axiome aus 3.3(1) (dabei wird die $\longleftrightarrow$-Eliminierung wie ursprünglich im Skript als eine Formel benutzt).

Definition 2:

Zwei Formeln a,b heißen äquivalent, wenn $a \longleftrightarrow b$ allgemeingültig ist; erfüllbarkeitsgleich, wenn a erfüllbar ist genau dann, wenn b erfüllbar ist.

Satz 1:

Jede quantorenfreie Formel kann man effektiv in eine äquivalente Formel in konjunktiver Normalform überführen.

Beweis:

Mit den Äquivalenzen vom Hilfssatz 1 $\longrightarrow$ und $\longleftrightarrow$ eliminieren, Negationen nach innen treiben, doppelte Negationen eliminieren, Konjunktionen nach außen treiben (falls gewünscht, doppelte Disjunktionsglieder und Konjunktionsglieder eliminieren).

Definition 3:

Eine Klausel ist eine endliche Menge von Literalen; die Literale heißen ihre Summanden. Die leere Klausel wird mit $\square$ bezeichnet. Eine Klauselmenge ist eine endliche Menge von Klauseln; die Klauseln heißen ihre Faktoren. Eine Grundklausel(menge) enthält keine freien Variablen. Das Klauselbild einer Formel a

$$(a_{1,1} \vee \ldots \vee a_{1,n1}) \wedge \ldots \wedge (a_{m,1} \vee \ldots \vee a_{m,nm})$$

in konjunktiver Normalform ist die Klauselmenge K(a)

$$\{\{a_{1,1},\ldots,a_{1,n1}\},\ldots,\{a_{m,i},\ldots,a_{m,nm}\}\}.$$

Umgekehrt heißt die Formel ein Formelbild F(M) der Klauselmenge M (eindeutig nur bis auf die Reihenfolge der Konjunktions- und Disjunktionsglieder). Eine Grundklauselmenge heißt erfüllbar, wenn ihre Formelbilder erfüllbar sind; die Grundklauselmenge, die nur die leere Klausel enthält, ist unerfüllbar.

Bemerkung:

Klauselmengen sind also nur abgekürzte Darstellungen für Formeln in disjuktiver Normalform.

Folgerung 1:

Jede variablenfreie Formel kann man effektiv in eine erfüllbarkeitsgleiche Grundklauselmenge überführen.

Definition 4:

a) Der Grundresolutionskalkül ist folgendermaßen definiert:

Axiome: keine

Regel: Für Grundklauseln K_1, K_2 und ein Grundliteral L mit $L \in K_1$, $\neg L \in K_2$:

$$\frac{K_1,\ K_2}{(K_1 - \{L\}) \cup (K_2 - \{\neg L\})} \quad \text{(Grundresolventenregel)}$$

Die gewonnene Klausel heißt Resolvente von K_1 und K_2.

b) Eine Ableitung im Grundresolutionskalkül besteht aus einer Folge $M_1, \ldots, M_n$ von Grundklauselmengen, wobei für $1 < j \leq n$ die Menge M_j aus einer Menge M_i mit $i<j$ entsteht durch:
entweder Anwendung der Grundresolventenregel auf zwei Klauseln in M_i
oder Hinzufügen oder Weglassen von Klauseln aus M_i.

c) Eine Grundklausel K heißt ableitbar aus einer Grundklauselmenge M, wenn es eine Ableitung gibt, die mit M beginnt und mit $\{K\}$ endet. In Symbolen: $M \vdash K$.

Satz 2 (Korrektheit):

Der Grundresolutionskalkül ist korrekt bezüglich Erfüllbarkeit, d.h. ist M erfüllbar und $M \vdash K$, so ist $\{K\}$ erfüllbar.

Beweis:

$$\mathfrak{v}_S^*(F(K_1) \wedge F(K_1))(u) = 1 \Rightarrow$$
$$\mathfrak{v}_S^*(F((K_1 - \{L\}) \cup (K_2 - \{\neg L\})))(u) = 1$$

Folgerung 2:

$M \vdash \square \Rightarrow$ M unerfüllbar.

Satz 2 (Vollständigkeit):

Der Grundresolutionskalkül ist vollständig bezüglich Widerlegbarkeit, d.h.

M unerfüllbar $\Rightarrow M \vdash \square$.

Beweis:

Induktion über $n := (\sum_{K \in M} |K|) - |M|$.

Bemerkung:

Der Grundresolutionskalkül ist korrekt, aber nicht vollständig bezüglich Ableitbarkeit, d.h.

$F(M) \models F(K) \not\Rightarrow M \vdash K$.

Beweis:

$\{\{a\}\} \not\vdash \{a \wedge a\}$ für eine atomare Formel a.

Folgerung 3:

Für jede variablenfreie Formel a gilt:

a widersprüchlich $\Leftrightarrow K(a) \vdash \square$.

Der Grundresolutionskalkül liefert also für variablenfreie Formeln ein Testverfahren auf Widersprüchlichkeit. Das Testverfahren kann sogar zu einem Entscheidungsverfahren ausgebaut werden, da es zu jeder Grundklauselmenge M nur endlich viele Ableitungen gibt, die mit M beginnen und keine Wiederholungen enthalten.

Aufgaben:

1) Beweisen Sie die Äquivalenzen aus Hilfssatz 1: Doppelte Negation, de Morgansche Gesetze, Idempotenz für beliebige prädikatenlogische Formeln.

2) Es seien a,b,c atomare Formeln. Überführen Sie die folgenden Formeln in konjunktive Normalform und bestimmen Sie die Klauselbilder dazu:
$\neg\neg a \longleftrightarrow a$, $\neg(a \vee b) \longleftrightarrow (\neg a \vee \neg b)$, $a \vee b \vee c \longrightarrow a \wedge b \wedge c$.

3) Es sei a := Sch(m(p(e)),p(m(e))), b := T(f). Testen Sie $(a \wedge \neg a) \vee (b \wedge \neg b)$ mit dem Grundresolutionskalkül auf Widersprüchlichkeit.

Zum Abschluß dieses Abschnitts untersuchen wir ganz kurz, ob das Grundresolutionsverfahren praktisch anwendbar ist. Genaueres dazu in den Lehrveranstaltungen "Effiziente Algorithmen" und "Algorithmentheorie".

Definition 5:

Die Länge einer quantorenfreien Formel ist die Anzahl ihrer Junktoren:

$lg(a)=0$ für $a\in At$, $lg(\neg a)=lg(a)+1$,

$lg(a*b)=lg(a)+lg(b)+1$ für $\wedge,\vee,\longrightarrow$ oder $\longleftrightarrow$ als $*$.

Die Länge einer Klausel ist ihre Mächtigkeit, die Länge einer Klauselmenge die Summe der Mächtigkeiten ihrer Faktoren:

$$lg(\{a_1,\ldots,a_n\})=n,\quad lg(\{K_1,\ldots,K_m\})=\sum_{i=1}^{m} lg(K_i).$$

Bemerkung:

Sei a eine quantorenfreie Formel in konjunktiver Normalform, in der kein Konjunktionsglied ein Literal mehrfach enthält (siehe Ende des Beweises von Satz 1). Dann gilt

$lg(a)=lg(\mathfrak{K}(a))-1$.

Dagegen gibt es quantorenfreie Formeln a, deren konkunktive Normalform b gemäß Satz 1 exponentiell länger ist, d.h.

$lg(b)\geq 2^{lg(a)}$.

Beispiel:

Sei

$$a:=\bigvee_{i=1}^{m}(a_i\wedge\neg a_i),\ \text{wobei } m\geq 2.$$

Dann ist

$$b=\bigwedge_{f\in M}\ \bigvee_{i=1}^{m} a_{i,f(i)},$$

wobei

$M:=\{f:\{1,\ldots,m\}\longrightarrow\{0,1\}\}$ und $a_{i,1}$ bzw. $a_{i,0}$ für a_i bzw. $\neg a_i$ steht.

Definition 6:

Der Aufwand des Grundresolutionsverfahrens aus Folgerung 3 sei die Anzahl der "Schritte" für eine eingegebene Formel bis zur Termination, also

Aufwand: {variablenfreie Formeln} $\longrightarrow \mathbb{N}$.

Folgerung 4:

Für jedes $n\geq 2$ gibt es Formeln a mit $lg(a)=n$ und Aufwand$(a)\geq 2^n$.

Beweis:

Das Verfahren braucht für die Formeln aus der obigen Bemerkung schon so lange, um das Klauselbild herzustellen.

Bemerkung:

Das Grundresolutionsverfahren hat also exponentiellen Aufwand. Solche Verfahren sind nicht praktisch durchführbar: Die Verlängerung der Eingabe um 1 kann den Aufwand verdoppeln. Tatsächlich ist die Situation noch viel schlimmer: Für jedes genügend große n gibt es Klauselmengen M der Länge n, so daß es exponentiell viele Ableitungen aus M gibt, die nur die Grundresolventenregel benutzen. Beispiel:

$M=\{\{a_1,\ldots,a_m,b\},\{\neg a_1,\ldots,\neg a_m\}\}$ für $m\geq 9$; denn $lg(M)=2m+1$, Anzahl der Ableitunge aus M ist m!, also ist Aufwand $(F(M))>m!$ Um das Verfahren praktisch verwendbar zu machen, benutzt man daher Strategien, die mehr oder weniger geschickt aus den möglichen Ableitungen auswählen. Vergleiche dazu die Bücher

E. Bergmann, H. Noll: Mathematische Logik mit Informatik-Anwendungen, Springer Verlag 1977,

C.L.Chang, R.C.T. Lee: Symbolic Logic and Mechanical Theorem Proving. Academic Press 1973,

D.W. Loveland: Automated Theorem Proving: A Logical Basis. North-Holland Publ.Comp. 1978.

Aufgaben:

4) Beweisen Sie, daß der Grundresolutionskalkül ein Entscheidungsverfahren liefert (siehe Folgerung 3).

5) Beweisen Sie die Behauptungen in den beiden obigen Bemerkungen.

6.3. Das Resolutionsverfahren

In diesem Abschnitt dehnen wir das Grundresolutionsverfahren auf Formeln mit Variablen und Quantoren aus.

Definition 1:

Eine prädikatenlogische Formel a ist in pränexer Normalform, wenn alle Quantoren allen aussagenlogischen Junktoren vorangehen:

$a = Q_1 x_1 \ldots Q_n x_n b$, b quantorenfrei, $Q_i = \forall$ oder $Q_i = \exists$.

Die Folge $Q_1 x_1 \ldots Q_n x_n$ heißt Präfix von a, die Formel b Kern von a.

Hilfssatz 1:

Seien a,b Formeln. Es sind allgemeingültig:

$\neg\exists x a \leftrightarrow \forall x \neg a$, $\neg\forall x a \leftrightarrow \exists x \neg a$,

$(\exists x a \vee \exists x b) \leftrightarrow \exists x (a \vee b)$, $(\forall x a \wedge \forall x b) \leftrightarrow \forall x (a \wedge b)$,

$(\exists x a \overset{\vee}{\wedge} b) \leftrightarrow \exists x (a \overset{\vee}{\wedge} b)$, $(\forall x a \overset{\vee}{\wedge} b) \leftrightarrow \forall x (a \overset{\vee}{\wedge} b)$, wenn $x \notin fr(b)$

$\exists x a \leftrightarrow \exists y\ sub(a)(id(y/x))$, $\forall x a \leftrightarrow \forall y\ sub(a)(id(y/x))$, wenn $y \notin fr(a)$.

Satz 1:

Man kann jede Formel effektiv in eine äquivalente Formel in pränexer Normalform überführen.

Beweis:

Mit Hilfssatz 1 Quantoren nach außen ziehen.

Definition 2:

Eine Formel ist in Skolem(scher Normal)-Form, wenn sie in pränexer Normalform ist, wobei das Präfix nur Allquantoren enthält.

Hilfssatz 2:

Für jede prädikatenlogische Formel a gilt:

$\forall y_1 \ldots \forall y_m \exists x a$ erfüllbar $\Leftrightarrow \forall y_1 \ldots \forall y_m\ sub(a)(id(f(y_1,\ldots,y_m)/x))$ erfüllbar

wobei das Funktionssymbol f in a nicht vorkommt.

Beweis: $\Rightarrow$: Die Belegung u:Var $\longrightarrow$ A erfüllt die Formel

$\forall y_1 \ldots \forall y_m \exists x a$ in der Struktur S $\Leftarrow$

für alle $e_1,\ldots,e_m \in A$ gibt es $e \in A$, so daß die Belegung

$u' := u(e_1/y_1,\ldots,e_m/y_m,e/x)$ die Formel a in S erfüllt $\Rightarrow$

für alle $e_1,\ldots,e_m \in A$ erfüllt $u'' := u(e_1/y_1,\ldots,e_m/y_m,f_s(e_1,\ldots,e_m)/x)$ die Formel a in S, wobei die Funktion $f_s: A^m \longrightarrow A$ für alle $e_1,\ldots,e_m \in A$ ein $e \in A$ wie oben auswählt $\Leftarrow$

für alle $e_1,\ldots,e_m \in A$ erfüllt $u'' = u(e_1/y_1,\ldots,e_m/y_m) \overset{*}{\circ} id(f(y_1,\ldots,y_m)/x)$ die Formel a in S $\Leftarrow$ (Lemma 4.5.1.)

für alle $e_1,\ldots,e_m \in A$ erfüllt $u(e_1/y_1,\ldots,e_m/y_m)$ die Formel $sub(a)(id(f(y_1,\ldots,y_m)/x))$ in S $\Leftarrow$

u erfüllt $\forall y_1 \ldots \forall y_m$ $sub(a)(id(f(y_1,\ldots,y_m)/x))$ in S. $\Leftarrow$: Entsprechend umgekehrt.

Satz 2:

Man kann jede Formel effektiv in eine erfüllbarkeitsgleiche Formel (Def. 6.2.2) in Skolemform überführen.

Beweis:

Die Formel mit Satz 1 pränex machen, mit Hilfssatz 2 von außen nach innen die Existenzquantoren eliminieren.

Definition 3:

Das Klauselbild K(a) einer prädikatenlogischen Formel a ohne freie Variable erhält man, in dem man a mit Satz 2 in Skolemform überführt, das Präfix wegläßt, den Kern in konjunktiver Normalform bringt und davon das Klauselbild M herstellt. Umgekehrt heißt dann a oder die konjunktive Normalform des Kerns von a ein Formelbild F(M) der Klauselmenge M. Das Formelbild von $\{\square\}$ sei die Formel O. Eine Klauselmenge heißt erfüllbar, wenn ihre Formelbilder erfüllbar sind.

Bemerkung:

Für quantorenfreie Formeln und Grundklauselmengen stimmen diese Begriffe mit denen aus Def. 6.2.3 überein.

Folgerung 1:

Man kann jede Formel ohne freie Variablen effektiv in eine erfüllbarkeitsgleiche Klauselmenge überführen.

Bemerkung:

Die Variablen in einer Formel in Skolemform sind durch Allquantoren gebunden. im Klauselbild sind sie jedoch frei. Man findet daher in einer widersprüchlichen Klauselmenge Literale L und $\neg L$, die man durch die Resolventenregel eliminieren könnte, im Allgemeinen nicht direkt, sondern in-dem man geeignete Terme durch Substitutionen gleich macht. Beispiel: Die Formel $\forall x P(x) \wedge \forall y \neg P(y)$ ist widersprüchlich. Sie hat die Skolemform $\forall x \forall y [P(x) \wedge \neg P(y)]$ und das Klauselbild $\{\{P(x)\},\{\neg P(y)\}\}$. Durch die Substitution id(t/x,t/y) für irgendeinen Term t wird daraus $\{\{P(t)\},\{\neg P(t)\}\}$, woraus man durch Resolution $\{\square\}$ erhält. Ähnlich hat $\forall x P(x) \wedge \exists y \neg P(y)$ die Skolemform $\forall x [P(x) \wedge \neg P(f(x))]$ und das Klauselbild $\{\{P(x)\},\{\neg P(f(x))\}\}$, aus dem man erst durch Umbenennen von x in y in der ersten Klausel und durch die Substitution id(t/x,f(t)/y) die resolvierbare Klauselmenge $\{\{P(f(t))\},\{\neg P(f(t))\}\}$ erhält.

Definition 4:

Seien L und N zwei atomare Formeln, sei u:Var $\longrightarrow$ Term eine Substitution: u heißt Vereinheitlicher von L und N, wenn rep(L)(u)=rep(N)(u). Ist u ein Vereinheitlicher von L und N und gibt es zu jedem anderen Vereinheitlicher v von L und N eine Substitution w, so daß $v = w \overset{*}{\circ} u$, so heißt u ein allgemeinster Vereinheitlicher von L und N.

Satz 3:

Der folgende Algorithmus entscheidet, ob zwei atomare Formeln L und N vereinheitlicht werden können; falls ja, liefert er einen allgemeinsten Vereinheitlicher:
Falls L und N mit verschieden(stellig)en Prädikatensymbolen beginnen, sei die Ausgabe "nein". Anderenfalls sind L und N von der Form Pp bzw. Pq, wobei p und q Folgen von Termen sind. Setze u:=id.(+) Von links her streiche gleiche Anfangszeichen von p und q. Falls dabei p und q gelöscht werden, war p=q; gib u aus. Anderenfalls erhält man p' und q', die mit verschiedenen Zeichen beginnen. Es seien s und t die beiden eindeutig bestimmten Terme, mit denen p' und q' beginnen.

Falls der eine dieser Terme eine Variable x ist, die in dem anderen, bezeichnen wir ihn mit r, nicht vorkommt, setze u:=id(r/x)∘u, p:=rep(p')(id(r/x)), q:=rep(q')(id(r/x)) und beginne wieder bei (+). Anderenfalls gib "nein" aus.

In Funktionsprozeduren geschrieben:

Uni(P(s_1,...,s_m),Q(t_1,...,t_n)):=

if P≠Q or m≠n then "nein" else uni(s_1,...,s_m;t_1,...,t_n;id)

uni(p;q;u):= if p=q then u else

if p=wp', q=wq', p'≠q', interm(p',q')={x,r}, x∈Var-Var(r)

then uni(rep(p')(id(r/x)); rep(q')(id(r/x)); id(r/x)∘u) else "nein"

Beweis:

a) Man zeigt (erster Teil durch Induktion nach der Anzahl der Zeichen von p): Wenn uni(p;q;u)=v∘u, dann rep(p)(v)=rep(q)(v); wenn uni(p;q;u)="nein", dann rep(p)(v)≠ rep(q)(v) für alle v.

b) Man zeigt: Wenn Uni(L,N)=u, so ist u ein allgemeinster Vereinheitlicher von L und N; wenn Uni(L,N)="nein", so gibt es keinen Vereinheitlicher von L und N.

Beispiel:

Sei L:=P(fx,gx), N:=P(z,ghy). Dann gilt Uni(L,N)=uni(fx,gx;z,ghy;id). Dafür liefert der Algorithmus:

p	q	u	p'	q'	x	r
fx,gx	z,ghy	id	fx,gx	z,ghy	z	fx
fx,gx	fx,ghy	id(fx/z)	x	hy	x	hy
hy	hy	id(hy/z)∘id(fx/z)				

Also gilt Uni(L,N)=id(hy/x)∘id(fx/z) [≠id(hy/x,fx/z)!]

Definition 5:

Der Resolutionskalkül ist folgendermaßen definiert:

Axiome: keine

Regeln: (1) Für Klausel K und Permutation u:Var ⟶ Var

$$\frac{K}{rep(K)(u)}$$ (Umbenennung)

(2) Für Klausel K, Literale L, N∈K, allgemeinsten Vereinheitlicher u von L und N:

$$\frac{K}{rep(K)(u)}$$ (Vereinheitlichung)

(3) Für Klauseln K_1 und K_2 mit $Var(K_1)\cap Var(K_2)=\emptyset$, Literale $L_1\in K_1$, $\neg L_2\in K_2$, allgemeinsten Vereinheitlicher u von L_1 und L_2:

$$\frac{K_1,\ K_2}{(rep(K_1)(u)-rep(L_1)(u))\cup(rep(K_2)(u)-rep(\neg L_2)(u))}$$ (Resolventenregel)

Die so gewonnene Klausel heißt Resolvente von K_1 und K_2.

Ableitung und ableitbar sind definiert wie im Grundresolutionskalkül mit den obigen drei Regeln. Wir schreiben wieder M⊢K.

Satz 4 (Korrektheit):

Der Resolutionskalkül ist korrekt bezüglich Erfüllbarkeit, d.h. ist M erfüllbar und M⊢K, so ist K erfüllbar.

Beweis:

Korrektheit der Regeln nachprüfen.

Bemerkungen:

a) Für Grundklauseln stimmt der Resolutionskalkül mit dem Grundresolutionskalkül überein.

b) Der Kalkül ist sinnvoll definiert, da mit Satz 3 entscheidbar ist, ob eine Folge $M_1,...,M_n$ eine Ableitung ist.

Um die Widerlegungsvollständigkeit des Resolutionskalküls zu beweisen, brauchen wir die folgenden Begriffsbildungen. Sie sind von dem polnischen Logiker Jacques Herbrand parallel zu dem Vollständigkeitsbeweis von Gödel (1930) entwickelt werden.

Definition 6:

Sei M eine Klauselmenge:

a) Das Herbrand-Universum H(M) ist die Menge der variablenfreien Terme, die man aus den Operationssymbolen in M bilden kann. Enthält M keine Operationssymbole, sei H(M):={d} für ein beliebiges nullstelliges Operationssymbol d (Konstante).

b) Die Herbrand-Sättigung $S(M)$ ist die Menge aller "Beispielformeln" $F(K)$ für $K \in M$, in denen die freien Variablen durch "Beispiele" aus $H(M)$ ersetzt sind:

Beispiel: $S(M) := \{rep(F(K))(u);\ K \in M,\ u: Var \longrightarrow H(M)\}$.

$M = \{\{P(c),\ Q(x)\}, \{\neg P(f(d))\}\}$. Dann gilt:

$H(M) = \{c, d, f(c), f(d), f(f(c)), \ldots\}$;

$S(M) = \{P(c) \vee Q(c), P(c) \vee Q(d), P(c) \vee Q(f(c)),$

$P(c) \vee Q(f(d)), P(c) \vee Q(f(f(c))), \ldots, P(f(d))\}$

$S(M)$ ist also eine (i.A. unendliche) Menge von variablenfreien Disjunktionen.

Satz 5 (Herbrand-Theorem):

Sei M eine Klauselmenge:
M ist erfüllbar genau dann, wenn $S(M)$ erfüllbar ist.

Vereinbarung:

Zur Vereinfachung schreiben wir im Folgenden $u(N)$ oder uN statt $rep(N)(u)$ für Literale, Klauseln oder Formeln N.

Hilfssatz (Lifting Lemma):

Seien K_1, K_2 Klauseln. Es gebe Literale $L_1 \in K_1, \neg L_2 \in K_2$ und Substitutionen u_1, u_2, so daß $K'_1 := u_1 K_1$ und $K'_2 := u_2 K_2$ keine Variablen gemeinsam haben und $L'_1 := u_1 L_1$ und $L'_2 := u_2 L_2$ vereinheitlicht werden können, also auf K'_1 und K'_2 die Resolventenregel bezüglich der Literale L'_1 und L'_2 angewendet werden kann. Ist K' die Resolvente von K'_1 und K'_2 bezüglich L'_1 und L'_2, so gibt es Faktoren $v_i K_i$ von K_i und eine Resolvente K von $v_1 K_1$ und $v_2 K_2$ bezüglich $v_1 L_1$ und $v_2(\neg L_2)$ und eine Substitution u mit $uK = K'$. (D.h. die Resolvente K' von K'_1 und K'_2 wird zu einer Resolvente K von K_1 und K_2 "hochgehoben".)

Satz 6 (Widerlegungsvollständigkeit):

Sei S eine unerfüllbare Klauselmenge. Dann gilt $S \vdash \square$ im Resolutionskalkül.

Folgerung 2:

Das folgende Verfahren liefert also ein Testverfahren auf Widerlegbarkeit: Die Eingabeformel a mit Folgerung 1 in eine Klauselmenge M überführen und aus M mit dem Resolutionskalkül die leere Klausel $\square$ abzuleiten suchen (durch Aufzählung aller Ableitungen wie in Folgerung 6.1.1).

Denn nach Folgerung 1 und den Sätzen 4 und 6 ist a widerlegbar genau dann, wenn $M \vdash \square$. Dieses Verfahren für beliebige Formeln ohne freie Variable ist aber sicher noch weniger praktisch durchführbar als das Grundresolutionsverfahren für Formeln ohne Variable (siehe Ende von 6.2).

AUFGABEN: Untersuchen Sie eine der folgenden Formeln

a: $:= \forall x \forall y (\forall x \forall y\ P(x,y) \longrightarrow \neg(\forall x\ P(x,y) \longrightarrow \exists y\ P(x,y)))$

b: $:= \forall x\ (P(x) \wedge \forall y (P(y) \longrightarrow P(g(x,y))) \wedge \neg \forall y (Q(x,y) \longrightarrow P(y)))$

c: $:= \forall x\ P(x) \wedge \forall x \forall y (P(y) \longrightarrow Q(x,g(y))) \wedge \neg \forall x \forall y\ Q(x,g(y))$

d: $:= \forall x (P(x) \longrightarrow Q(x)) \wedge \forall y (Q(y) \longrightarrow R(y)) \wedge \neg \forall y [P(y) \longrightarrow R(y)]$

mit Hilfe des Resolutionskalküls. D.h. überführen Sie die Formel

(1) in pränexe Normalform,

(2) in Skolem-Form,

(3) in Klauselform.

(4) Testen Sie die Formel mit Hilfe des Resolutionskalküls auf Widerlegbarkeit.

Beweis von Satz 5: $\Rightarrow$:

Sei M erfüllbar. D.h. die Formel F(M) hat ein Modell S. Also ist $v_S^*(F(K))(\bar{u})=1$ für alle $\bar{u}:Var \longrightarrow S$ und alle $K \in M$. Daher ist nach Lemma 4.5.1 $v_S^*(rep(F(K))(u))(w)=v_S^*(F(K))(w^* \circ u)=1$ für alle $u:Var \longrightarrow H(M)$, $w:Var \longrightarrow S$, $K \in M$, da $w^* \circ u:Var \longrightarrow S$. Also ist S ein Modell von S(M).

$\Leftarrow$: Sei S(M) erfüllbar. D.h. die Menge S(M) hat ein Modell S. Also ist $v_S^*(a)(w)=1$ für alle $w:Var \longrightarrow S$ und alle $a \in S(M)$. D.h. $v_S^*(rep(F(K))(u))(w)=1$ für alle $w:Var \longrightarrow S$ und $K \in M$ und $u:Var \longrightarrow H(M)$. Sei $w:Var \longrightarrow S$ eine beliebige Belegung. Wir konstruieren eine Struktur H: Der Individuenbereich sei die Termmenge H(M) (Herbrand-Universum von M); die Operationen seien die Termoperationen,

d.h.

$$f_H(t_1,\ldots,t_n):=f(t_1,\ldots,t_n) \text{ für } t_1,\ldots,t_n \in H(M),\ f \in \Omega;$$

die Prädikate seien definiert durch

$$P_H(t_1,\ldots t_n):=v_S^*(P(t_1,\ldots,t_n))(w) \text{ für } t_1,\ldots t_n \in H(M),\ P \in ?\ .$$

Sei $u:Var \longrightarrow H(M)$ eine Belegung in H. Dann gilt für atomare Formeln $P(t_1,\ldots,t_n)$ (mit Variablen)

$$v_H^*(P(t_1,\ldots,t_n))(u)=P_H(u^*(t_1),\ldots,u^*(t_n))=$$

$$=v_S^*(P(u^*(t_1),\ldots,u^*(t_n))=(\text{Lemma } 4.5.1)\, v_S^*(rep(P(t_1,\ldots,t_n))(u))(w)\ .$$

Also gilt $v_H^*(a)(u)=v_S^*(rep(a)(u))(w)$ für alle atomaren Formeln a, also auch für alle Literale a, also auch für alle Konjunktionen a von Literalen.

Speziell gilt

$$v_H^*(F(K))(u)=v_S^*(rep(F(K))(u))(w)=1$$

für alle $K \in M$. Also ist H ein Modell von M.

Schluß für dieses Semester !

Beweise für Lifting-Lemma und Satz 6 z.B. in Bergmann-Noll.

SCHÖNE FERIEN