Logik für Informatiker[1] Sätze und Definitionen

Martin Ziegler

Freiburg 2006/2007

[1]Version 1.3, 2009-01-25

Inhaltsverzeichnis

Kapitel 1

Aussagenlogik

1.1 Grundbegriffe

Definition (Syntax…)**.** Aussagenlogische Formeln sind Zeichenreihen, die sich aus den *Variablen* $A_0, A_1, \ldots$, den *Junktoren* $\wedge$, $\vee$ und $\neg$ und den beiden Klammern (und) nach den folgenden Regeln aufbauen.

- Variable sind Formeln.
- Wenn F_1 und F_2 Formeln sind, dann ist auch $(F_1 \wedge F_2)$ eine Formel (*Konjunktion*).
- Wenn F_1 und F_2 Formeln sind, dann ist auch $(F_1 \vee F_2)$ eine Formel (*Disjunktion*).
- Wenn F eine Formel ist, dann ist auch $\neg F$ eine Formel (*Negation*).

Man liest $\wedge$ als „und", $\vee$ als „oder" und $\neg$ als „nicht". Wir verwenden die folgenden Abkürzungen:

$$
\begin{array}{rcl}
(F \to G) & = (\neg F \vee G) & (\textit{Implikation}) \\
(F \leftrightarrow G) & = ((F \to G) \wedge (G \to F)) & (\textit{Äquivalenz}) \\
\bigwedge_{i<n} F_i & = (F_0 \wedge \ldots \wedge (F_{n-2} \wedge F_{n-1}) \ldots) & \\
\bigvee_{i<n} F_i & = (F_0 \vee \ldots \vee (F_{n-2} \vee F_{n-1}) \ldots) & \\
\top & = (A_0 \vee \neg A_0) & (\textit{Wahr}) \\
\bot & = (A_0 \wedge \neg A_0) & (\textit{Falsch})
\end{array}
$$

Man liest $\to$ als „impliziert" und $\leftrightarrow$ als „äquivalent". Gelegentlich fassen wir $\top$ und $\bot$ als nicht–zusammengesetzte Formeln auf. Die *leere* Konjunktion $\bigwedge_{i<0} F_i$ wird mit $\top$, die leere Disjunktion mit $\bot$ identifiziert.

Wir könnten auf $\vee$ als Grundsymbol verzichten und $(F \vee G)$ als Abkürzung für $\neg(\neg F \wedge \neg G)$ auffassen. In der Prädikatenlogik (Kapitel 2) werden wir so verfahren.

Hilfssatz. *Eine Formel kann kein echtes Anfangsstück einer anderen Formel sein.*

Lemma 1.1.1 (Eindeutige Lesbarkeit von Formeln)**.** *Jede Formel ist* ***entweder*** *eine*

- *eine Variable,*
- *eine Konjunktion* $(F_1 \wedge F_2)$*,*
- *eine Disjunktion* $(F_1 \vee F_2)$
- *oder eine Negation* $\neg F$

Die Formeln F_1*,* F_2 *bzw.* F *sind jeweils eindeutig bestimmt.*

Wenn keine Mißverständnisse auftreten können, lassen wir gelegentlich Klammern weg.

Definition. Die Wahrheitswerte 0 und 1 stehen für „falsch“ und „wahr“. Sei $\mathcal{D}$ eine Menge von Variablen. Eine Belegung ist eine Funktion $\mathcal{A} : \mathcal{D} \to \{0, 1\}$.

Wir lassen die Junktoren auf den Wahrheitswerten wie folgt operieren:

$$w_1 \wedge w_2 = \begin{cases} 1 & \text{wenn } w_1 = 1 \text{ und } w_2 = 1 \\ 0 & \text{sonst} \end{cases}$$

$$w_1 \vee w_2 = \begin{cases} 1 & \text{wenn } w_1 = 1 \text{ oder } w_2 = 1 \\ 0 & \text{sonst} \end{cases}$$

$$\neg w = \begin{cases} 1 & \text{wenn } w = 0 \\ 0 & \text{wenn } w = 1 \end{cases}$$

Mit Wahrheitstafeln drückt sich das so aus:

$\wedge$	0	1
0	0	0
1	0	1

$\vee$	0	1
0	0	1
1	1	1

$\neg$	0	1
	1	0

Definition (... und Semantik)**.** Sei $\mathcal{A}$ eine Belegung der Variablen von F, dann ist der Wahrheitswert $\mathcal{A}(F)$ von F

- $\mathcal{A}(B)$, wenn $F = B$ eine Variable ist,
- $\mathcal{A}(F_1) \wedge \mathcal{A}(F_2)$, wenn $F = (F_1 \wedge F_2)$,
- $\mathcal{A}(F_1) \vee \mathcal{A}(F_2)$, wenn $F = (F_1 \vee F_2)$,

- $\neg\mathcal{A}(F')$, wenn $F = \neg F'$.

Wenn $\mathcal{A}(F) = 1$, sagen wir dafür auch „$\mathcal{A}$ erfüllt F“, „$\mathcal{A}$ ist Modell von F“ oder wir schreiben $\mathcal{A} \models F$.

Definition. Eine Formel F heißt allgemeingültig oder (aussagenlogische) Tautologie, wenn alle Belegungen F erfüllen. F heißt erfüllbar, wenn es eine Belegung gibt, die F erfüllt.

Lemma 1.1.2.

- *F genau dann eine Tautologie, wenn $\neg F$ nicht erfüllbar ist.*
- *F ist genau dann erfüllbar, wenn $\neg F$ keine Tautologie ist.*

1.2 Äquivalente Formeln und Normalformen

Definition. Zwei Formeln F und G heißen äquivalent, wenn sie dieselben Modelle haben. Wir schreiben dafür

$$F \equiv G$$

- $F \equiv G$ genau dann, wenn $F \leftrightarrow G$ allgemeingültig ist.
- F ist genau dann allgemeingültig, wenn $F \equiv \top$.
- F ist genau dann erfüllbar, wenn $F \not\equiv \bot$.
- $\equiv$ ist eine Äquivalenzrelation.

Definition. Sei F eine Formel, aufgebaut aus $\wedge$, $\vee$, $\neg$, $\top$ und $\bot$. Die duale Formel F^* entsteht aus F durch Vertauschen von $\wedge$ und $\vee$ und von $\top$ und $\bot$.

Lemma 1.2.1. *$F \equiv G$ genau dann, wenn $F^* \equiv G^*$.*

Satz 1.2.2. *Es gelten die folgenden grundlegenden Äquivalenzen* ***und ihre Duale****. (A, B, C sind Variable.)*

$$
\begin{aligned}
(A \wedge A) &\equiv A && \textit{(Idempotenz)}\\
(A \wedge B) &\equiv (B \wedge A) && \textit{(Kommutativität)}\\
((A \wedge B) \wedge C) &\equiv (A \wedge (B \wedge C)) && \textit{(Assoziativität)}\\
(A \wedge (A \vee B)) &\equiv A && \textit{(Absorption)}\\
(A \wedge (B \vee C)) &\equiv ((A \wedge B) \vee (A \wedge C)) && \textit{(Distributivität)}\\
(\bot \wedge A) &\equiv \bot\\
(A \wedge \neg A) &\equiv \bot
\end{aligned}
$$

$A \vee \neg A \equiv \top$, das duale der letzten Äquivalenz heißt *Tertium non datur*.

Definition. Seien F und H Formeln und A eine Variable. Dann bezeichnen wir mit $F(H/A)$ die Formel, die aus F dadurch entsteht, daß wir jedes Vorkommen von A in F durch H ersetzen.

Lemma 1.2.3 (Ersetzungslemma)**.** *$F \equiv G$ impliziert $F(H/A) \equiv G(H/A)$ und $H(F/A) \equiv H(G/A)$.*

Es gelten die *de Morganschen Regeln*

$$
\begin{aligned}
\neg(A \wedge B) &\equiv (\neg A \vee \neg B)\\
\neg(A \vee B) &\equiv (\neg A \wedge \neg B)\\
\neg\neg A &\equiv A
\end{aligned}
$$

Definition. Eine Literal ist ein Variable oder eine negierte Variable. Ein Formel der Form

$$\bigvee_{i<m} \bigwedge_{j<n_i} L_{ij}$$

für Literale L_{ij} hat disjunktive Normalform. Eine Formel der Form

$$\bigwedge_{i<m} \bigvee_{j<n_i} L_{ij}$$

hat konjunktive Normalform.

Satz 1.2.4. *Jede Formel ist zu einer Formel in disjunktiver Normalform und zu einer Formel in konjunktiver Normalform äquivalent.*

Folgerung 1.2.5. *Jede Äquivalenz $F \equiv G$ läßt sich mit Hilfe des Ersetzungslemmas 1.2.3 aus den Grundäquivalenzen von Satz 1.2.2 formal ableiten.*

1.3 Boolesche Algebren

Definition. Eine Boolesche Algebra $(B, 0, 1, \sqcap, \sqcup, {}^c)$ eine Menge B mit zwei ausgezeichneten Elementen 0 und 1 und Operationen $\sqcap, \sqcup : B^2 \to B$ und ${}^c : B \to B$, für die folgenden Gleichungen gelten:

$$
\begin{aligned}
a \sqcap a &= a & a \sqcup a &= a & (1.1)\\
a \sqcap b &= b \sqcap a & a \sqcup b &= b \sqcup a & (1.2)\\
(a \sqcap b) \sqcap c &= a \sqcap (b \sqcap c) & (a \sqcup b) \sqcup c &= a \sqcup (b \sqcup c) & (1.3)\\
a \sqcap (a \sqcup b) &= a & a \sqcup (a \sqcap b) &= a & (1.4)\\
a \sqcap (b \sqcup c) &= (a \sqcap b) \sqcup (a \sqcap c) & a \sqcup (b \sqcap c) &= (a \sqcup b) \sqcap (a \sqcup c) & (1.5)\\
0 \sqcap a &= 0 & 1 \sqcup a &= 1 & (1.6)\\
a \sqcap a^c &= 0 & a \sqcup a^c &= 1 & (1.7)
\end{aligned}
$$

Die Axiome entsprechen den grundlegenden Äquivalenzen von Satz 1.2.2. Das zweite Distributivitätsaxiom ist überflüssig. In Booleschen Algebren gelten die de Morganschen Regeln.

Eine Struktur $(V, \sqcap, \sqcup)$, in der (1.1), (1.2), (1.3) und (1.4) gelten, ist ein *Verband*. Wenn $(P, \leq)$ eine partielle Ordnung ist, in der je zwei Element ein Supremum und ein Infimum haben, ist $(P, \inf, \sup)$ ein Verband. Jeder Verband hat diese Gestalt, wenn man $a \leq b$ durch $a \sqcap b = a$ oder, äquivalent, durch $a \sqcup b = b$ definiert. Die Gleichungen (1.6) bedeuten, daß 0 und 1 kleinstes und größtes Element sind; (1.7) drückt aus, daß a^c ein *Komplement* von a ist. In einem *distributiven* Verband, d.h. in einem Verband, in dem (1.5) gilt, gibt es immer nur höchstens ein Komplement. Eine Boolesche Algebra ist also nichts anderes als ein „komplementärer distributiver Verband".

Beispiel 1: Sei X eine Menge. Dann ist die Potenzmenge $\mathfrak{P}(X)$ zusammen mit den ausgezeichneten Elementen $\emptyset$ und X und den Operationen Durchschnitt, Vereinigung und Komplement $A^c = X \setminus A$ eine Boolesche Algebra, die *Potenzmengenalgebra* von X.

Satz 1.3.1. *Endliche Boolesche Algebren sind isomorph zu Potenzmengenalgebren. Jede unendliche Boolesche Algebra ist isomorph zu einer Unteralgebra einer Potenzmengenalgebra.*

Beispiel 2: Die Elemente der *Lindenbaumalgebra* LA_n sind Äquivalenzklassen $F/\equiv$ von Formeln F, die höchstens die Variablen $A_0, \ldots, A_{n-1}$ enthalten. Wenn man definiert

$$
\begin{aligned}
0 &= \top/\equiv\\
1 &= \bot/\equiv\\
(F/\equiv) \sqcap (G/\equiv) &= (F \wedge G)/\equiv\\
(F/\equiv) \sqcup (G/\equiv) &= (F \vee G)/\equiv\\
(F/\equiv)^c &= \neg F/\equiv,
\end{aligned}
$$

wird LA_n zu einer Booleschen Algebra. LA_n ist isomorph zur Potenzmengenalgebra einer Menge mit 2^n Elementen. LA_n wir von den $A_i/\equiv$ „frei“ erzeugt.

1.4 Die Resolutionmethode

Definition. Eine *Klausel* C ist eine endliche Menge von Literalen.

Eine Belegung $\mathcal{A}$ der Variablen erfüllt die Klausel C, wenn $\mathcal{A}(L) = 1$ für ein $L \in C$. Anders ausgedrückt, wenn $\mathcal{A} \models \bigvee_{L \in C} L$. Sei $\mathcal{C}$ eine Menge von Klauseln. $\mathcal{A}$ erfüllt $\mathcal{C}$, wenn $\mathcal{A}$ alle $C \in \mathcal{C}$ erfüllt. Das kann man auch schreiben als

$$\mathcal{A} \models \bigwedge_{C \in \mathcal{C}} \bigvee_{L \in C} L.$$

Definition. Sei A eine Variable, $C = \{A\} \cup P$ und $D = \{\neg A\} \cup Q$ zwei Klauseln. Dann ist $P \cup Q$ eine Resultante von C und D.

Satz 1.4.1 (Die Resolutionsmethode)**.** *Eine endliche Menge $\mathcal{C}$ von Klauseln ist genau dann erfüllbar, wenn sich aus $\mathcal{C}$ durch sukzessives Bilden von Resultanten niemals die leere Klausel ergibt.*

Der Beweis beruht auf folgender Beobachtung: Sei mit $\mathcal{C}|A = 1$ die Menge der Klauseln bezeichnet, die man aus $\mathcal{C}$ erhält, wenn man $A = 1$ setzt. Das heißt, daß man alle Klauseln, in denen A vorkommt, wegläßt und in den verbleibenden Klauseln alle Vorkommen von $\neg A$ streicht. Entsprechend sei $\mathcal{C}|A = 0$ definiert. Dann gilt das folgende Lemma.

Lemma 1.4.2.

- *Sei $\mathcal{A}$ eine Belegung der Variablen von $\mathcal{C}$ und $\mathcal{A}(A) = w$. Dann ist $\mathcal{A}$ ein Modell von $\mathcal{C}$ von $\mathcal{A}$ genau dann, wenn $\mathcal{A}$ ein Modell von $\mathcal{C}|A = w$ ist.*

- *Aus $\mathcal{C}$ ergibt sich durch sukzessives Bilden von Resultanten genau dann die leere Klausel, wenn das gleiche für $\mathcal{C}|A = 1$ und für $\mathcal{C}|A = 0$ gilt.*

1.5 Der Satz von Cook

Definition. Sei $\mathbb{A}$ ein endliches Alphabet und $W \subset \mathbb{A}^*$ eine Menge von Wörtern. W heißt polynomial, wenn es eine Maschine M mit polynomialer Laufzeit[1] gibt, die für jedes w entscheidet, ob $w \in W$. Wir schreiben dann

$$W \in \mathsf{P}.$$

Definition. W heißt nicht–deterministisch polynomial , wenn es eine nicht–deterministische[2] Maschine mit polynomialer Laufzeit gibt, sodaß $w \in W$ genau dann, wenn M mit Input w einen Lauf hat, der w akzeptiert. Wir schreiben

$$W \in \mathsf{NP}$$

Es ist klar, daß P eine Teilmenge von NP ist.

Um aussagenlogische Formeln in einem endlichen Alphabet zu schreiben, verwenden wir statt der Variablen A_0, A_1, ... nicht–leere Wörter über einem vernünftigen Alphabet, zum Beispiel $\{\mathrm{A}, \mathrm{B}, \ldots, \mathrm{Z}\}$. Es ist klar, daß die Menge aller Formeln zu P gehört. Die Menge

$$\mathsf{SAT}$$

aller erfüllbaren Formeln gehört zu NP.

Satz 1.5.1 (Cook)**.** SAT *ist NP-vollständig. Das heißt: Sei* $W \in \mathsf{NP}$*. Dann läßt sich jedes Wort* w *in polynomialer Zeit in eine Formel* F *umwandeln, sodaß*

$$w \in W \;\Leftrightarrow\; F \in \mathsf{SAT}.$$

Folgerung 1.5.2. $\mathsf{NP} = \mathsf{P}$ *ist äquivalent zu* $\mathsf{SAT} \in \mathsf{P}$.

Das Erfüllbarkeitsproblem einer Formel in konjunktiver Normalform ist ebenfalls NP-vollständig:

Proposition 1.5.3. *Zu jeder Formel läßt sich in polynomialer Zeit eine Menge von Klauseln angeben, die genau dann erfüllbar ist, wenn* F *erfüllbar ist.*

[1] Es gibt ein Polynom $p(x)$, sodaß die Maschine bei Input w nach höchstens $p(|w|)$ vielen Schritten stoppt.

[2] Das Programm einer solchen Maschine enthält Befehle, die nicht zwingend ausgeführt werden müssen.

1.6 Der Kompaktheitsatz der Aussagenlogik

Satz 1.6.1 (Kompaktheitssatz)**.** *Eine (unendliche) Menge von Formeln ist genau dann erfüllbar, wenn jede endliche Teilmenge erfüllbar ist.*

Der Satz gilt auch, wenn man überabzählbar viele Variablen zuläßt. Man braucht dazu

Hilfssatz (Zorns Lemma)**.** *Sei* $(P, \leq)$ *eine partiell geordnete Menge. Wenn jede linear geordnete Teilmenge von* P *eine obere Schranke hat, hat* P *ein maximales Element.*

Definition. Sei T eine Menge von Formeln. Eine Formel F folgt aus T, wenn jedes Modell von T auch ein Modell von F ist.

Wir schreiben dafür

$$T \vdash F.$$

Folgerung 1.6.2. *Wenn* F *aus* T *folgt, dann folgt* F *schon aus einer endlichen Teilmenge von* T.

Die nächste Proposition ist ein Anwendungsbeispiel. Ein Graph ist f–färbbar, wenn man jeder seiner Ecken eine der Farben $0, 1, \ldots, f$ so zuordnen kann, daß verbundene Ecken verschiedene Farben haben.

Proposition 1.6.3. *Sei* f *eine natürliche Zahl. Ein Graph* G *ist genau dann* f*–färbbar, wenn jeder endliche Teilgraph von* G f*–färbbar ist.*

Kapitel 2

Prädikatenlogik

2.1 Strukturen und Formeln

Definition. Eine Sprache L ist eine Menge von Konstanten, Funktionszeichen und Relationszeichen. Funktionszeichen und Relationszeichen haben eine (positive) *Stelligkeit*.

Definition. Sei L eine Sprache. Eine L–Struktur ist ein Paar

$$\mathfrak{A} = (A, (Z^{\mathfrak{A}})_{Z \in L}),$$

wobei

A	eine nicht–leere Menge (die *Grundmenge von* $\mathfrak{A}$) ist,
$Z^{\mathfrak{A}} \in A,$	wenn Z eine Konstante ist,
$Z^{\mathfrak{A}} : A^n \longrightarrow A,$	wenn Z ein n–stelliges Funktionszeichen ist und
$Z^{\mathfrak{A}} \subset A^n,$	wenn Z ein n–stelliges Relationszeichen ist.

Definition. Zwei L–Strukturen $\mathfrak{A}$ und $\mathfrak{B}$ heißen isomorph, $\mathfrak{A} \cong \mathfrak{B}$, wenn es einen *Isomorphismus* $F : \mathfrak{A} \longrightarrow \mathfrak{B}$ gibt, eine Bijektion $F : A \longrightarrow B$, die mit den Interpretationen der Zeichen aus L kommutiert:

$F(Z^{\mathfrak{A}})$	$=$	$Z^{\mathfrak{B}}$	(Z eine Konstante aus L)
$F(Z^{\mathfrak{A}}(a_1, \ldots, a_n))$	$=$	$Z^{\mathfrak{B}}(F(a_1), \ldots, F(a_n))$	(Z ein n–stelliges Funktionszeichen aus L, $a_1, \ldots, a_n \in A$)
$R^{\mathfrak{A}}(a_1, \ldots, a_n)$	$\Leftrightarrow$	$R^{\mathfrak{B}}(F(a_1), \ldots, F(a_n))$	(Z ein n–stelliges Relationszeichen aus L, $a_1, \ldots, a_n \in A$)

Ein Isomorphismus von $\mathfrak{A}$ mit sich selbst heißt Automorphismus.

Das Inverse eines Isomorphismus und die Verknüpfung von Isomorphismen sind wieder Isomorphismen. Daraus folgt, daß

$$\mathrm{Aut}(\mathfrak{A}),$$

die Menge aller Automorphismen von $\mathfrak{A}$, eine Gruppe ist.

Wir fixieren ein Folge von Variablen $v_0, v_1, \ldots$.

Definition. Ein L–Term ist eine Zeichenfolge, die nach den folgenden Regeln gebildet ist:

T1 Jede Variable ist ein L–Term.

T2 Jede Konstante aus L ist ein L–Term.

T3 Wenn f ein n–stelliges Funktionszeichen aus L ist und wenn $t_1, \ldots, t_n$ L–Terme sind, dann ist auch $ft_1 \cdots t_n$ ein L–Term.

Um Terme besser lesbar zu machen, schreiben wir häufig $f(t_1, \ldots, t_n)$ statt $ft_1 \cdots t_n$. Wenn f einstellig ist, auch $t_1 f$, wenn f zweistellig ist, auch $t_1 f t_2$. Zum Beispiel steht $(x+y) \cdot (z+w)$ für $\cdot + xy + zw$ und $(x \circ y)^{-1}$ für ${}^{-1} \circ xy$.

Lemma 2.1.1 (Eindeutige Lesbarkeit von Termen)**.** *Für jeden L–Term t tritt genau einer der folgenden drei Fälle ein:*

1. *t ist eine Variable,*

2. *t ist eine Konstante,*

3. *$t = ft_1 \cdots t_n$, wobei f ein n–stelliges Funktionszeichen und $t_1, \ldots, t_n$ L–Terme sind.*

Im letzten Fall sind f und $t_1, \ldots, t_n$ eindeutig bestimmt.

Das Lemma folgt aus den nächsten Hilfsatz

Hilfssatz. *Kein L–Term ist echtes Anfangsstück eines anderen L–Terms.*

L–Formeln sind Zeichenreihen, die aus den Zeichen aus L, den Klammern (und) als Hilfszeichen und den folgenden *logischen* Zeichen gebildet ist:

Variable	$v_0, v_1, \ldots$
Gleichheitszeichen	$\doteq$
Junktoren	$\neg$ (Negation), $\wedge$ (Konjunktion)
Existenzquantor	$\exists$

Man liest $\doteq$ als „gleich“, $\neg$ als „nicht“, $\wedge$ als „und“ und $\exists$ als „es gibt ein“.

Definition. Eine L–Formel ist

F1 $t_1 \doteq t_2$, wenn t_1, t_2 L–Terme sind,

F2 $Rt_1 \ldots, t_n$, wenn R ein n–stelliges Relationszeichen aus L und $t_1, \ldots, t_n$ L–Terme sind,

F3 $\neg\psi$, wenn ψ eine L–Formel ist,

F4 $(\psi_1 \wedge \psi_2)$,wenn ψ_1 und ψ_2 L–Formeln sind,

F5 $\exists x\ \psi$,wenn ψ eine L–Formel und x eine Variable ist.

Formeln der Form **F1** und **F2** heißen *Primformeln*.

Wir verwenden folgende Abkürzungen:

$$\begin{aligned}
(\psi_1 \vee \psi_2) &= \neg(\neg\psi_1 \wedge \neg\psi_2) \\
(\psi_1 \rightarrow \psi_2) &= \neg(\psi_1 \wedge \neg\psi_2) \\
(\psi_1 \leftrightarrow \psi_2) &= ((\psi_1 \rightarrow \psi_2) \wedge (\psi_2 \rightarrow \psi_1)) \\
\forall x\ \psi &= \neg\exists x \neg\psi \\
(\psi_0 \wedge \cdots \wedge \psi_n) &= \underbrace{(\cdots(}_{n\text{ -mal}}\ \psi_0 \wedge \psi_1) \wedge \cdots \psi_n) \\
(\psi_0 \vee \cdots \vee \psi_n) &= \underbrace{(\cdots(}_{n\text{ -mal}}\ \psi_0 \vee \psi_1) \vee \cdots \psi_n)
\end{aligned}$$

Die Disjunktion $\vee$ liest man als „oder“, die Implikation $\rightarrow$ als „impliziert“, die Äquivalenz $\leftrightarrow$ als „genau dann, wenn“ und den Allquantor $\forall$ als „für alle“.

Statt Rt_1t_2 schreiben wir auch t_1Rt_2 und statt $\exists x_1 \cdots \exists x_n$ schreiben wir $\exists x_1 \ldots x_n$ (ebenso für $\forall$). Zur besseren Lesbarkeit der Formeln gebrauchen wir überflüssige Klammern. Wir lassen auch Klammern weg und lesen die Formeln gemäß der *Bindungsstärke* der logischen Zeichen:

Höchste Bindungsstärke: $\neg\ \exists\ \forall$

$\wedge$

$\vee$

Niedrigste Bindungsstärke: $\rightarrow\ \leftrightarrow$

Zum Beispiel steht $\neg\phi \wedge \psi \rightarrow \chi$ für

$$((\neg\phi \wedge \psi) \rightarrow \chi) = \neg((\neg\phi \wedge \psi) \wedge \neg\chi).$$

Lemma 2.1.2 (Eindeutige Lesbarkeit von Formeln)**.** *Für jede L–Formel ϕ tritt genau einer der folgenden Fälle ein.*

1. $\phi = t_1 \doteq t_2$ *für L–Terme t_1, t_2*

2. $\phi = Rt_1 \cdots t_n$ *für ein n–stelliges Relationszeichen R aus L und L–Terme $t_1, \ldots, t_n$*

3. $\phi = \neg\psi$ *für eine L–Formel ψ*

4. $\phi = (\psi_1 \wedge \psi_2)$ *für L–Formeln ψ_1 und ψ_2*

5. $\phi = \exists x\ \psi$ *für eine L–Formel ψ und eine Variable x*

In jedem der Fälle sind die Terme t_i, das Relationszeichen R, die Formeln ψ, ψ_1, ψ_2 und die Variable x jeweils eindeutig bestimmt.

Zum Beweis braucht man den folgenden Hilfssatz.

Hilfssatz. *Keine L–Formel ist echtes Anfangsstück einer anderen L–Formel.*

2.2 Semantik

Definition. Sei $\mathfrak{A}$ eine L–Struktur. Eine Belegung ist eine Funktion

$$\beta : \{v_0, v_1 \ldots\} \longrightarrow A$$

von der Menge der Variablen in die Grundmenge von $\mathfrak{A}$.

Definition. Für L–Terme t, L–Strukturen $\mathfrak{A}$ und Belegungen β definieren wir $t^{\mathfrak{A}}[\beta]$ durch

$$\begin{aligned} v_i^{\mathfrak{A}}[\beta] &= \beta(v_i) \\ c^{\mathfrak{A}}[\beta] &= c^{\mathfrak{A}} \\ ft_1 \cdots t_n^{\mathfrak{A}}[\beta] &= f^{\mathfrak{A}}(t_1^{\mathfrak{A}}[\beta], \ldots, t_n^{\mathfrak{A}}[\beta]). \end{aligned}$$

Lemma 2.2.1. *Wenn die Belegungen β und γ auf den Variablen, die in t vorkommen übereinstimmen, ist $t^{\mathfrak{A}}[\beta] = t^{\mathfrak{A}}[\gamma]$.* □

Wenn wir einen Term in der Form $t(x_1, \ldots, x_n)$ schreiben, meinen wir:

1. daß die x_i paarweise verschiedene Variable sind,
2. daß in t nur Variable aus $\{x_1, \ldots, x_n\}$ vorkommen.

Wenn dann $a_1, \ldots, a_n$ Elemente der Struktur $\mathfrak{A}$ sind, ist wegen 2.2.1

$$t^{\mathfrak{A}}[a_1, \ldots, a_n]$$

durch $t^{\mathfrak{A}}[\beta]$ für eine Belegung β mit $\beta(x_i) = a_i$ wohldefiniert.

Die folgende rekursive Definition der Semantik von Formeln ist sinnvoll, wegen 2.1.2.

Definition. Sei $\mathfrak{A}$ eine L–Struktur. Wir definieren für Belegungen β und L–Formeln ϕ die Relation

$$\mathfrak{A} \models \phi[\beta]$$

— *ϕ trifft in $\mathfrak{A}$ auf β zu*— durch Rekursion über den Aufbau von ϕ:

$$\begin{aligned} \mathfrak{A} \models t_1 \doteq t_2\,[\beta] &\Leftrightarrow t_1^{\mathfrak{A}}[\beta] = t_2^{\mathfrak{A}}[\beta] \\ \mathfrak{A} \models Rt_1 \cdots t_n\,[\beta] &\Leftrightarrow R^{\mathfrak{A}}(t_1^{\mathfrak{A}}[\beta], \ldots, t_n^{\mathfrak{A}}[\beta]) \\ \mathfrak{A} \models \neg\psi\,[\beta] &\Leftrightarrow \mathfrak{A} \not\models \psi\,[\beta] \\ \mathfrak{A} \models (\psi_1 \wedge \psi_2)\,[\beta] &\Leftrightarrow \mathfrak{A} \models \psi_1\,[\beta] \text{ und } \mathfrak{A} \models \psi_2\,[\beta] \\ \mathfrak{A} \models \exists x\psi\,[\beta] &\Leftrightarrow \text{ es gibt ein } a \in A \text{ mit } \mathfrak{A} \models \psi\,[\beta\frac{a}{x}]. \end{aligned}$$

Dabei ist $\beta\frac{a}{x}(y) = \begin{cases} \beta(y) & \text{, wenn } y \neq x \\ a & \text{, wenn } y = x. \end{cases}$

Definition. Die Variable x kommt *frei* in der Formel ϕ vor, wenn sie an einer Stelle vorkommt, die nicht im Wirkungsbereich eines Quantors $\exists x$ liegt. Präzis definiert durch Rekursion nach dem Aufbau von ϕ:

$$\begin{aligned} x \text{ frei in } t_1 \doteq t_2 &\Leftrightarrow x \text{ kommt in } t_1 \text{ oder in } t_2 \text{ vor} \\ x \text{ frei in } Rt_1 \cdots t_n &\Leftrightarrow x \text{ kommt in einem der } t_i \text{ vor.} \\ x \text{ frei in } \neg\psi &\Leftrightarrow x \text{ frei in } \psi \\ x \text{ frei in } (\psi_1 \wedge \psi_2) &\Leftrightarrow x \text{ frei in } \psi_1 \text{ oder } x \text{ frei in } \psi_2 \\ x \text{ frei in } \exists y\, \psi &\Leftrightarrow x \neq y \text{ und } x \text{ frei in } \psi \end{aligned}$$

Satz 2.2.2. *Wenn β und γ an allen Variablen, die frei in ϕ vorkommen, übereinstimmen, ist*

$$\mathfrak{A} \models \phi[\beta] \Leftrightarrow \mathfrak{A} \models \phi[\gamma].$$

Wenn wir eine Formel in der Form $\phi(x_1, \ldots, x_n)$ schreiben, meinen wir:

1. daß die x_i paarweise verschiedene Variable sind,
2. daß in ϕ nur Variable aus $\{x_1, \ldots, x_n\}$ frei vorkommen.

Wenn dann $a_1, \ldots, a_n$ Elemente der Struktur $\mathfrak{A}$ sind, ist wegen 2.2.2

$$\mathfrak{A} \models \phi[a_1, \ldots, a_n]$$

durch $\mathfrak{A} \models \phi[\beta]$ für eine Belegung β mit $\beta(x_i) = a_i$ wohldefiniert.

Definition. Eine *Aussage* ϕ ist eine Formel ohne freie Variable. Wir schreiben $\mathfrak{A} \models \phi$, wenn $\mathfrak{A} \models \phi[\beta]$ für ein (alle) β.

Sprechweisen:

- ϕ gilt in $\mathfrak{A}$.
- ϕ ist wahr in $\mathfrak{A}$.
- $\mathfrak{A}$ ist Modell von ϕ.
- $\mathfrak{A}$ erfüllt ϕ.

Zwei L–Strukturen $\mathfrak{A}$ und $\mathfrak{B}$ heißen *elementar äquivalent*, wenn in ihnen die gleichen Aussagen gelten.

Lemma 2.2.3. *Isomorphe Strukturen sind elementar äquivalent.*

Sei x eine Variable und s ein Term. Dann entsteht der Term

$$t\frac{s}{x}$$

aus t durch Ersetzen aller Vorkommen von x durch s.

Lemma 2.2.4 (Substitutionslemma für Terme)**.** *Sei x eine Variable, s ein Term und β eine Belegung mit Werten in der Struktur $\mathfrak{A}$. Dann ist für jeden Term T*

$$(t\frac{s}{x})^{\mathfrak{A}}[\beta] = t^{\mathfrak{A}}[\beta\frac{s^{\mathfrak{A}}[\beta]}{x}].$$

Sei $t = t(x_1, \ldots, x_n)$, und $s = s(x_1, \ldots, x_n)$. Dann kann man das Substitutionslemma schreiben als

$$t(s, x_2, \ldots, x_n)^{\mathfrak{A}}[a_1, \ldots, a_n] = t^{\mathfrak{A}}[s^{\mathfrak{A}}[a_1, \ldots, a_n], a_2, \ldots, a_n].$$

Die Formel

$$\phi\frac{s}{x}$$

entsteht aus ϕ durch Ersetzen aller freien Vorkommen von x durch s.

Die rekursive Definition von $\phi\frac{s}{x}$ ist

$$\begin{aligned}
(t_1 \doteq t_n)\frac{s}{x} &= t_1\frac{s}{x} \doteq t_2\frac{s}{x} \\
(Rt_1 \cdots t_n)\frac{s}{x} &= Rt_1\frac{s}{x} \cdots t_n\frac{s}{x} \\
(\neg\psi)\frac{s}{x} &= \neg(\psi\frac{s}{x}) \\
(\psi_1 \wedge \psi_2)\frac{s}{x} &= (\psi_1\frac{s}{x} \wedge \psi_2\frac{s}{x}) \\
(\exists y\psi)\frac{s}{x} &= \exists y(\psi\frac{s}{x}), \text{ wenn } x \neq y \\
&= \exists y\psi, \text{ wenn } x = y.
\end{aligned}$$

Definition. x heißt *frei für s in ϕ*, wenn kein freies Vorkommen von x in ϕ im Wirkungsbereich eines Quantors ist, der eine Variable von s bindet.

Rekursive Definition: x ist frei für s in ϕ, wenn x nicht frei in ϕ ist **oder** wenn x frei in ϕ ist und einer der folgenden Fälle zutrifft

$\phi = t_1 \doteq t_n$
$\phi = Rt_1 \cdots t_n$
$\phi = \neg\psi$ und x frei für s in ψ
$\phi = (\psi_1 \wedge \psi_2)$ und x frei für s in ψ_1 und ψ_2,
$\phi = \exists y\psi, x$ frei für s in ψ und y kommt nicht in s vor.

Lemma 2.2.5 (Substitutionslemma für Formeln)**.** *Sei x eine Variable, s ein Term und β eine Belegung mit Werten in der Struktur $\mathfrak{A}$. Wenn x frei für s in der Formel ϕ, ist*

$$\mathfrak{A} \models \phi\frac{s}{x}[\beta] \Longleftrightarrow \mathfrak{A} \models \phi[\beta\frac{s^{\mathfrak{A}}[\beta]}{x}],$$

Sei $\phi = \phi(x_1, \ldots, x_n)$ und $s = s(x_1, \ldots, x_n)$. Dann kann man das Substitutionslemma schreiben als

$$\mathfrak{A} \models \phi(s, x_2, \ldots, x_n)[a_1, \ldots, a_n] \Longleftrightarrow \mathfrak{A} \models \phi[s^{\mathfrak{A}}[a_1, \ldots, a_n], a_2, \ldots, a_n].$$

2.3 Allgemeingültige Formeln

Definition. Eine Formel ϕ heißt *allgemeingültig*, wenn sie für alle Belegungen β in allen Strukturen[1] $\mathfrak{A}$ gilt. Wir schreiben dafür

$$\models \phi.$$

$\phi(x_1, \ldots, x_n)$ ist genau dann allgemeingültig, wenn die Aussage $\forall x_1 \ldots x_n \phi(x_1, \ldots, x_n)$ allgemeingültig ist.

Definition. Eine (prädikatenlogische) *Tautologie* entsteht aus einer aussagenlogischen Tautologie durch Ersetzen der Aussagenvariablen durch L–Formeln.

Lemma 2.3.1. *Tautologien sind allgemeingültig.*

Lemma 2.3.2 (Axiome der Gleichheit)**.** *Die folgenden L–Aussagen sind allgemeingültig.*

$$\begin{array}{rl}
\textit{Reflexivität} & \forall x \;\; x \doteq x \\
\textit{Symmetrie} & \forall x, y \;\; x \doteq y \to y \doteq x \\
\textit{Transitivität} & \forall x, y, z \;\; x \doteq y \land y \doteq z \to x \doteq z \\
 & \forall x_1, \ldots, x_n, y_1, \ldots y_n \;\; x_1 \doteq y_1 \land \cdots \land x_n \doteq y_n \\
 & \qquad\qquad\qquad \longrightarrow f x_1 \ldots x_n \doteq f y_1 \ldots y_n \\
 & \forall x_1, \ldots, x_n, y_1, \ldots y_n \;\; x_1 \doteq y_1 \land \cdots \land x_n \doteq y_n \\
 & \qquad\qquad\qquad \longrightarrow (R x_1 \ldots x_n \leftrightarrow R y_1 \ldots y_n)
\end{array}$$

Dabei sind die f n–stellige Funktionszeichen und die R n–stellige Relationszeichen aus L.

Die ersten drei Aussagen drücken aus, daß = eine Äquivalenzrelation ist. Die beiden letzten bedeuten, daß = eine *Kongruenzrelation* ist.

Lemma 2.3.3 ($\exists$–Quantorenaxiome)**.** *ϕ sei eine L–Formel, t ein L–Term und x frei für t in ϕ. Dann ist*

$$\phi \frac{t}{x} \longrightarrow \exists x \phi$$

allgemeingültig.

Lemma 2.3.4 (Modus Ponens)**.** *Wenn ϕ und $(\phi \to \psi)$ allgemeingültig sind, dann auch ψ.*

Lemma 2.3.5 ($\exists$–Einführung)**.** *Wenn x nicht frei in ψ vorkommt, dann ist mit $\phi \to \psi$ auch $\exists x \phi \to \psi$ allgemeingültig.*[2]

---

[1] Wenn ϕ eine L–Formel ist, sind L–Strukturen gemeint. Der Begriff hängt von der Wahl von L nicht ab.

[2] Man spricht von *Existenzeinführung*.

2.4 Der Gödelsche Vollständigkeitssatz

Definition (Der Hilbertkalkül)**.** L sei eine Sprache. Eine L–Formel ist *beweisbar*, wenn sie

B1 eine Tautologie ist,

B2 ein Gleichheitsaxiom ist,

B3 ein $\exists$–Quantorenaxiom ist,

B4 sich mit Hilfe der Modus Ponens Regel aus zwei beweisbaren L–Formeln ergibt,

B5 oder wenn sie sich mit der Regel der $\exists$–Einführung aus einer beweisbaren L–Formel ergibt.

B1-B3 sind die Axiome , **B4-B5** die Schlußregeln des Hilbertkalküls.

Eine Formel ϕ ist also genau dann beweisbar, wenn sie eine *Beweis* hat, das heißt, eine Folge

$$\phi_0, \ldots, \phi_n = \phi$$

von L–Formeln ϕ_i, die entweder Axiome sind oder mit einer der beiden Schlußregel aus Formeln ϕ_j $(j < i)$ folgen. Wir schreiben:

$$\vdash_L \phi$$

Satz 2.4.1 (Gödelscher Vollständigkeitssatz)**.** *Ein Formel ist genau dann allgemeingültig, wenn sie beweisbar ist:*[3]

$$\models \phi \iff \vdash_L \phi$$

Die die Richtung $\Leftarrow$ folgt aus 2.3.1, 2.3.2, 2.3.3, 2.3.4 und 2.3.5. Zum Beweis der Umkehrung macht man Gebrauch von den folgenden *abgeleiteten* Axiome und Regeln.

Lemma 2.4.2. *1.* **(Aussagenlogik)** *Wenn ϕ_1,...,ϕ_n beweisbar sind und $(\phi_1 \wedge \cdots \wedge \phi_n) \to \psi$ eine Tautologie ist, ist auch ψ beweisbar.*

2. **($\forall$–Quantorenaxiome)** *Wenn x frei für t in ϕ, ist*

$$\vdash_L \forall x \phi \to \phi \frac{t}{x}.$$

3. **($\forall$–Einführung)** *Wenn x nicht frei in ϕ ist, dann folgt aus der Beweisbarkeit von $\phi \to \psi$ die Beweisbarkeit von $\phi \to \forall x \psi$. Insbesondere folgt aus der Beweisbarkeit von ψ die Beweisbarkeit von $\forall x \psi$.*

[3] Es folgt, daß $\vdash_L \phi$ von der Sprachumgebung unabhängig ist. Wir verwenden darum später die Notation $\vdash \phi$.

Beweisschema des Gödelschen Vollständigkeitssatzes Nehmen wir an $\phi(\bar{x})$ sei nicht beweisbar. Wir müssen eine Struktur $\mathfrak{A}$ und Elemente $\bar{a} \in A$ finden mit $\mathfrak{A} \models \neg\phi[\bar{a}]$.

Schritt 1) Wir wählen eine Folge $\bar{c}$ von neuen Konstanten. Dann ist $\phi(\bar{c})$ ebenfalls nicht beweisbar. (Das folgt aus 2.4.3).

Schritt 2) Wir erweitern $\{\neg\phi(\bar{c})\}$ zu einer *widerspruchsfreien Henkintheorie* mit Konstantenmenge C. (Die Definitionen folgen unten.)

Schritt 3) Wir erweitern T^+ zu einer *vollständigen* Henkintheorie T^*.

Schritt 4) Wir konstruieren ein Modell[4] $(\mathfrak{A}, a_c)_{c \in C}$ von T^*.

In $\mathfrak{A}$ gilt $\neg\phi[a_{\bar{c}}]$.

Lemma 2.4.3. *Sei $\phi(x_1, \ldots, x_n)$ eine L–Formel, C eine Menge von neuen Konstanten und $c_1, \ldots, c_n$ eine Folge von paarweise verschiedenen Elementen von C. Dann ist*

$$\vdash_{L \cup C} \phi(c_1, \ldots, c_n) \iff \vdash_L \phi(x_1, \ldots, x_n).$$

Definition. Eine L–*Theorie* ist eine Menge von L–Aussagen. Eine Aussage ϕ ist in T *beweisbar*,

$$T \vdash \phi,$$

wenn es Axiome $\psi_1, \ldots, \psi_n$ aus T gibt für die $\vdash_L \psi_1 \wedge \cdots \wedge \psi_n \to \phi$.[5]

Definition. Eine L–Theorie T heißt *widerspruchsfrei* oder *konsistent*, wenn $\bot$ nicht in T beweisbar ist.[6] T ist *vollständig*, wenn T widerspruchsfrei ist und $T \vdash \phi$ oder $T \vdash \neg\phi$ für alle L–Aussagen ϕ.

Definition. Sei C eine Menge von Konstanten. Eine $L \cup C$–Theorie T^+ heißt Henkintheorie mit Konstantenmenge C, wenn es zu jeder $L \cup C$–Formel $\phi(x)$ eine Konstante $c \in C$ mit

$$T^+ \vdash \exists x \phi(x) \to \phi(c)$$

gibt.

Definition. Ein *Modell* von T ist eine L–Struktur, in der alle Aussagen aus T gelten.

Der skizzierte Beweis zeigt die folgende Verschärfung von 2.4.1.

[4] Die Konstruktion liefert $A = \{a_c | c \in C\}$. Solche Modelle *aus Konstanten* sind durch T^* bis auf Isomorphie eindeutig bestimmt.

[5] Man beachte, daß es wegen 2.4.2.1 auf die Klammerung und Reihenfolge der ϕ_i nicht ankommt.

[6] Wir nehmen für $\bot$ irgendeine L–Aussage, deren Negation allgemeingültig ist. Zum Beispiel $\exists x \, \neg x \doteq x$.

Satz 2.4.4. *Eine Theorie hat genau dann ein Modell, wenn sie widerspruchsfrei ist.*

Definition. Die Aussage ϕ *folgt logisch* aus T,

$$T \models \phi,$$

wenn ϕ in allen Modellen von T gilt.

Folgerung 2.4.5.

$$T \vdash \phi \qquad \Leftrightarrow \qquad T \models \phi$$

Folgerung 2.4.6 (Kompaktheitssatz)**.** *Eine Theorie hat genau dann ein Modell, wenn jede endliche Teilmenge ein Modell hat.*

Den Kompaktheitssatz könnte man den ersten Hauptsatz der Modelltheorie nennen. Der zweite Hauptsatz wäre der Satz von Löwenheim-Skolem, aus dem Beweis von 2.4.4 folgt.

Folgerung 2.4.7 (Löwenheim-Skolem)**.** *Wenn eine Theorie mit höchstens abzählbarer Sprache eine Modell hat, hat sie ein abzählbares Modell*

Folgerung 2.4.8 (aus 2.4.1)**.** *Sei L eine endlichen Sprache, dann ist die Menge der allgemeingültigen L–Aussagen effektiv aufzählbar.*

Die Folgerung gilt auch für effektiv aufgezählte Sprachen.

2.5 Der Herbrandsche Satz und automatisches Beweisen

Definition. Eine Formel ist in pränexer Normalform, wenn alle Quantoren am Anfang der Formel stehen, wenn also die Formel die Gestalt

$$Q_1 x_1 Q_2 x_2 \ldots Q_n x_n \psi$$

hat, mit quantorenfreiem ψ und $Q_i \in \{\exists, \forall\}$.

Lemma 2.5.1 (Pränexe Normalform)**.** *Jede Formel läßt sich in eine äquivalente*[7] *pränexe Formel umwandeln.*

Definition. Eine universelle Formel hat die Form

$$\forall x_1 \ldots \forall x_n \psi,$$

wobei ψ eine quantorenfreie Formel ist. Existentielle Formeln beginnen mit Existenzquantoren.

Satz 2.5.2 (Skolem–Normalform)**.** *Zu jeder L–Aussage ϕ kann man eine Spracherweiterung L^* und einen universellen L^*–Satz ϕ^* angeben, derart, daß ϕ in einer L–Struktur $\mathfrak{A}$ genau dann gilt, wenn sich $\mathfrak{A}$ zu einem Modell von ϕ^* expandieren läßt. ϕ ist also genau dann erfüllbar, wenn ϕ^* erfüllbar ist.*

Man nennt die im Beweis von 2.5.2 neu eingeführten Konstanten und Funktionszeichen in L^* *Skolemfunktionen.*

Folgerung 2.5.3 (Herbrand–Normalform)**.** *Zu jeder L–Aussage ϕ kann man eine Spracherweiterung L^* und einen existentiellen L^*–Satz ϕ_* angeben, der genau dann allgemeingültig ist, wenn ϕ allgemeingültig ist.*

Satz 2.5.4 (Satz von Herbrand)**.** *Sei $\psi(x_1, \ldots, x_n)$ eine quantorenfreie*[8] *Formel in einer Sprache L, die mindestens eine Konstante enthält. Dann ist*

$$\phi = \exists x_1 \ldots \exists x_n \psi(x_1, \ldots, x_n)$$

genau dann allgemeingültig, wenn es konstante Terme[9]

$$t_1^1, t_2^1, \ldots t_n^1 \ldots t_1^N, t_2^N, \ldots t_n^N$$

gibt, für die die quantorenfreie Aussage

$$\bigvee_{i=1}^{N} \psi(t^i) = \psi(t_1^1, t_2^1, \ldots t_n^1) \vee \cdots \vee \psi(t_1^N, t_2^N, \ldots t_n^N)$$

allgemeingültig ist.

[7] ϕ und ψ sind *äquivalent*, wenn $\phi \leftrightarrow \psi$ allgemeingültig ist.
[8] Der Satz gilt auch für existentielle ψ.
[9] Ein konstanter Term ist ein Term ohne Variable.

Definition.

$$S^1(x_1,\ldots,x_n) = \big(s^1_1(x_1,\ldots,x_n),\ldots,s^1_k(x_1,\ldots,x_n)\big)$$

und

$$S^2(x_1,\ldots,x_n) = \big(s^2_1(x_1,\ldots,x_n),\ldots,s^2_k(x_1,\ldots,x_n)\big)$$

seien zwei gleichlange Folgen von Termen. Eine Termfolge $T = (t_1,\ldots,t_n)$ *unifiziert* S^1 und S^2, wenn

$$S^1(t_1,\ldots,t_n) = S^2(t_1,\ldots,t_n).$$

Satz 2.5.5 (Unifikation)**.** *Wenn S^1 und S^2 unifizierbar sind, gibt es eine* universelle *unifizierende Termfolge $U(y_1,\ldots,y_m)$. Das heißt, daß eine Termfolge T genau dann S^1 und S^2 unifiziert, wenn es Terme $r_1,\ldots,r_m$ gibt, sodaß*

$$T = U(r_1,\ldots,r_m).$$

Man kann U durch ein einfaches Verfahren finden, das gleichzeitig entscheidet, ob S^1 und S^2 unifizierbar sind.

Lemma 2.5.6. *Sei $F(A_1,\ldots,A_n)$ eine aussagenlogische Formel, in den paarweise verschiedenen Variablen A_i. Für jedes i sei α_i ein Primformel der Form*[10] *$Rt_1\ldots t_m$. Die α_i seien paarweise verschieden. Dann ist $F(\alpha_1,\ldots,\alpha_n)$ genau dann allgemeingütlig, wenn $F(A_1,\ldots,A_n)$ allgemeingültig ist.*

Folgerung 2.5.7. *Sei $\psi(x_1,\ldots,x_n)$ eine quantorenfreie L–Formel ohne Gleichheitszeichen und N eine natürliche Zahl*[11]*. Man kann effektiv entscheiden, ob es konstante Terme*

$$t^1_1, t^1_2, \ldots t^1_n \ldots t^N_1, t^N_2, \ldots t^N_n$$

gibt, für die

$$\bigvee_{i=1}^{N} \psi(t^i) = \psi(t^1_1, t^1_2, \ldots t^1_n) \vee \cdots \vee \psi(t^N_1, t^N_2, \ldots t^N_n)$$

allgemeingültig ist. □

Proposition 2.5.8. *Wir finden in 2.5.3 immer ein ϕ_* ohne Gleichheitszeichen.*

---

[10] Das Lemma gilt für beliebige Primformeln α_i.

[11] Der Fall $N = 1$ impliziert sofort die Folgerung für beliebiges N. Wir haben diese Formulierung gewählt wegen des Zusammenhangs mit Satz 2.5.4.

Kapitel 3

Unentscheidbarkeit der Prädikatenkalküls

Wir zeigen in diesem Kapitel, daß es kein Verfahren gibt, das bei vorgelegter Aussage ϕ entscheidet, ob ϕ beweisbar ist.

3.1 Registermaschinen

Eine Registermaschine M arbeitet mit endlichen vielen Registern $\mathcal{R}_0, \ldots, \mathcal{R}_{R-1}$, in denen Wörter aus einem endlichen Alphabet $\mathcal{A} = \{a_1, \ldots, a_L\}$ stehen. Die Maschine kann den letzten Buchstaben dieser Wörter lesen, den letzten Buchstaben streichen oder einen Buchstaben anhängen[1]. Das Programm von M ist eine Folge von Befehlen der folgenden Art.

`PUSH`(r,l)	Wenn $l = 0$, wird der letzte Buchstabe des Worts in $\mathcal{R}_r$ gestrichen. Sonst wird das Wort in $\mathcal{R}_r$ um den Buchstaben a_l verlängert. Danach wird der nächste Befehl ausgeführt.
`GOTO`$(r,c_0,\ldots,c_L)$	Liest das Wort in $\mathcal{R}_r$. Wenn das Wort leer ist, führe als nächstes den Befehl mit der Nummer c_0 aus. Wenn der letzte Buchstabe a_l ist, führe als nächstes den Befehl Nummer c_l aus.
`STOP`	Die Maschine stoppt.

Der letzte Befehl des Programms soll immer `STOP` sein

[1] Die Register sind also „stacks“.

M berechnet auf folgende Weise ein *partielle* Funktion

$$\mathrm{F}_M : \mathcal{A}^* \to \mathcal{A}^*.$$

Der Input x steht im Register $\mathcal{R}_0$, die anderen Register sind leer. Die Maschine führt, mit dem ersten Befehl beginnend, das Programm solange aus, bis der Befehl `STOP` ausgeführt wird. Wenn M niemals stoppt, ist $\mathrm{F}_M(x)$ undefiniert.

Satz 3.1.1 (Churchs These)**.** *Alle berechenbaren Funktionen*[2] $\mathcal{A}^* \to \mathcal{A}^*$ *haben die Form* F_M *für eine geeignete Registermaschine* M*.*

Folgerung 3.1.2 (Unlösbarkeit des Halteproblems)**.** *Sei* $\mathcal{A}$ *ein endliches Alphabet. Es gibt kein effektives Verfahren, das für alle Maschinen* M *und Wörter* x *entscheidet, ob* M *mit Input* x *stoppt.*

Folgerung 3.1.3. *Sei* $\mathcal{A}$ *ein endliches Alphabet. Es gibt kein effektives Verfahren, das für alle Maschinen* M *entscheidet, ob* M *(mit leerem Input) stoppt.*

Der Beweis von Folgerung 3.1.2 ist das gleiche Diagonalargument wie im Beweis des nächsten Satzes.

Satz 3.1.4 (Cantor)**.** *Die Menge alle unendlichen* 0–1*–Folgen ist überabzählbar.*

[2]Das gilt auch für „partielle berechenbare“ Funktionen.

3.2 Codierung von Registermaschinen

Satz 3.2.1. *Zu jeder Registermaschine M läßt sich ein Aussage ϕ^M angeben, die genau dann erfüllbar ist, wenn M (mit leerem Input) nicht stoppt.*

Man beachte, daß sowohl die Menge aller M, die stoppen, als auch die Menge aller ϕ, die nicht erfüllbar sind, effektiv aufzählbar sind.

Beweisskizze. Wir nehmen der Einfachheit halber an, daß $L = 1$, das heißt, daß $\mathcal{A} = \{a_1\}$ nur ein Zeichen enthält[3]. Die Elemente von $\mathcal{A}^*$ identifizieren wir mit natürlichen Zahlen. Sei $b_0, \dots, b_N$ das Programm von M, R die Zahl der Register. Wir können annehmen,daß b_N der einzige Stopbefehl ist.

Sei

$$L^M = \{0, f, z_0, \dots, z_N\},$$

wobei 0 eine Konstante, f ein einstelliges Funktionszeichen und die z_c R–stellige Relationszeichen. Betrachte die folgende L^M–Struktur

$$\mathfrak{M}^0 = \{\mathbb{N}, 0, (x \mapsto x + 1), Z^0_0, \dots, Z^0_N\},$$

wobei $Z^0_c(n_0, \dots, n_{R-1})$ genau dann, wenn M einen Zustand erreicht, in dem b_c der aktuelle Befehl ist und in Register $\mathcal{R}_r$ die Zahl n_r steht.

Man findet eine L^M–Aussage μ^M mit folgenden Eigenschaften:

- $\mathfrak{M}^0$ ist ein Modell von μ^M.
- Wenn $\mathfrak{A} = \{A, O, F, Z_0, \dots, Z_N\}$ ein Modell von μ^M ist, gilt

$$Z^0_c(n_0, \dots, n_{R-1}) \Rightarrow Z_c(F^{n_0}(O), \dots, F^{n_{R-1}}(O)).$$

Man setzt nun $\phi^M = \mu^M \wedge \forall x_0, \dots, x_{R-1} \neg z_N(x_0, \dots, x_{R-1})$. □

Folgerung 3.2.2. *Es gibt kein Verfahren, das bei vorgelegter Aussage ϕ entscheidet, ob ϕ allgemeingültig ist.*

Folgerung 3.2.3 (aus dem Beweis)**.** *Es gibt eine endliche Sprache L, für die sich nicht entscheiden läßt, ob eine gegebene L–Aussage allgemeingültig ist.*

Beweisskizze. Betrachte ein M, für das sich nicht entscheiden läßt, ob M bei gegebenem Input x stoppt. $L = L^M$ sei wie oben gewählt. Konstruiere für jedes x eine Aussage ϕ^x, die genau dann erfüllbar ist, wenn M bei Input x nicht stoppt. □

Satz 3.2.4. *Es gibt eine endliche Sprache L, für die sich nicht entscheiden läßt, ob eine gegebene L–Aussage ein endliches Modell hat.*

Beweisskizze. Sei M eine Maschine wie im Beweis von 3.2.3. Wir können annehmen, daß der Inhalt des Registers $\mathcal{R}_0$ während des Laufs immer größer wird. Wenn man μ^x, ähnlich wie im Beweis von 3.2.1, richtig wählt, hat μ^x genau dann ein endliches Modell, wenn M bei Input x stoppt. □

[3]Sonst muß man L Funktionszeichen verwenden.

3.3 Turingmaschinen

Wir geben einen alternativen Beweis von 3.2.2 mit Hilfe von Turingmaschinen. Eine Turingmaschine M verwendet als Speicher ein Band von Zellen, auf denen sich ein Lese/Schreibkopf bewegt. Die Zellen sind leer oder enthalten einen Buchstaben aus einem Alphabet $\mathcal{A} = \{a_1, \ldots, a_L\}$. Das Programm von M ist eine Folge von Befehlen der folgenden Art.

`WRITE`(l)	Wenn $l = 0$, wird die aktuelle Zelle gelöscht. Sonst wird der Buchstabe a_l in die Zelle geschrieben. Danach wird der nächste Befehl ausgeführt.
`LEFT`	Der Kopf bewegt sich einen Schritt nach links. Danach wird der nächste Befehl ausgeführt.
`RIGHT`	Der Kopf bewegt sich einen Schritt nach rechts. Danach wird der nächste Befehl ausgeführt.
`GOTO(`c_0`,...,`c_L`)`	Wenn die aktuelle Zelle leer ist, führe als nächstes den Befehl mit der Nummer c_0 aus. Wenn die Zelle den Buchstaben a_l enthält, führe als nächstes den Befehl Nummer c_l aus.
`STOP`	Die Maschine stoppt.

Der letzte Befehl des Programms soll immer `STOP` sein

Satz 3.3.1. *Registermaschinen lassen sich durch Turingmaschinen simulieren und umgekehrt.*

Folgerung 3.3.2 (Unlösbarkeit des Halteproblems für Turingmaschinen)**.** *Sei $\mathcal{A}$ ein endliches Alphabet. Es gibt kein effektives Verfahren, das für alle Turingaschinen M entscheidet, ob M mit leerem Anfangsband stoppt.*

Satz 3.3.3. *Zu jeder Turingmaschine M läßt sich ein Aussage ψ^M angeben, die genau dann erfüllbar ist, wenn M nicht stoppt.*

Beweisskizze. Wir können annehmen, daß $L = 1$. Sei $b_0, \ldots, b_N$ das Programm von M, b_N der einzige Stopbefehl. Sei

$$L^M = \{0, <, z, h, d_0, \ldots, d_N\},$$

wobei 0 eine Konstante, $<$ und die z zweistellige Relationen sind, h ein einstelliges Funktionszeichen und die d_c einstellige Relationszeichen. Betrachte die folgende L^M–Struktur

$$\mathfrak{M}^0 = \{\mathbb{Z}, 0, <, Z^0, H^0, D_0^0, \ldots, D_N^0\},$$

wobei

- $Z^0(s, n)$ genau dann, wenn nach s Schritten von M in der Zelle $n \in \mathbb{Z}$ der Buchstabe a_1 steht.
- $H^0(s)$ ist die Position des Kopfes nach s Schritten, wenn $s \geq 0$, sonst $=0$.
- $D_c^0(s)$, wenn nach s Schritten b_c der aktuelle Befehl ist.

Wir stellen uns dabei vor, daß die Maschine nach einem Stop-befehl weiterläuft und „auf der Stelle tritt".

Man findet eine L^M–Aussage μ^M mit folgenden Eigenschaften:

- $\mathfrak{M}^0$ ist ein Modell von μ^M.
- Wenn $\mathfrak{A} = \{A, \bar{0}, <, Z, H, D_0, \ldots, D_N\}$ ein Modell von μ^M ist, ist $(M, <)$ eine lineare Ordnung, in der jedes Element einen unmittelbaren Vorgänger und Nachfolger hat. Sei für jedes $n \in \mathbb{Z}$ das Element $\bar{n}$ der n–fache Nachfolger[4] von $\bar{0}$. Die Unterstruktur $\{\bar{n} \mid n \in \mathbb{Z}\}$ ist isomorph zu $\mathfrak{M}^0$.

Man setzt nun $\psi^M = \mu^M \wedge \forall x \neg z_N(x)$ $\square$

Wie im letzten Abschnitt folgt daraus 3.2.2, die Unentscheidbarkeit des Prädikatenkalküls.

[4] der $-n$–fache Vorgänger, wenn $n < 0$

Änderungen

Version 1.0 (2007-02-14) Erste vollständige Version.

Version 1.1 (2007-04-04) Im Abschnitt 1.1 waren die leere Konjunktion und Disjunktion falsch definiert. Druckfehler in Satz 1.5.1.

Version 1.2 (2007-10-12) *konstanter Term* definiert.

Version 1.3 (2009-01-25) Druckfehler in Definition konjunktive (disjunktive) Normalform. Lemma 1.4.2 geändert. In Abschnitt 1.3 wurden an einer Stelle inf und sup verwechselt.

Index