Logik für Studierende der Informatik

Markus Junker (Mathematisches Institut, Universität Freiburg)

Wintersemester 2017/18 – Version von 27. Februar 2018

Inhaltsverzeichnis

0 Kurze Einleitung

Logik wird traditionell häufig als die *Lehre vom korrekten Schließen* bezeichnet. Untersuchungsobjekte der Logik sind Argumente, die typischerweise die Form

$$\begin{array}{l} \textit{Prämisse } 1 \\ \quad\vdots \\ \textit{Prämisse } n \\ \hline \textit{Schlussfolgerung} \end{array}$$

annehmen. Untersucht wird, ob sich die Schlussfolgerung „logisch korrekt“ aus den Prämissen ergibt. Dabei bezieht sich logische Korrektheit üblicherweise nur auf die *Form* der Sätze unter Abstraktion von ihrem konkreten Inhalt. Logische Schlüsse lassen sich daher in einer formalen Sprache wiedergeben, weshalb Logik auch eine Theorie formaler Sprachen begründet. Logik ist daher aus mehreren Gründen für die Informatik interessant:

- Logik ist Grundlage aller Wissenschaft, insofern wissenschaftliche Argumentation logischen Standards genügen muss („Logik als Propädeutik“).
- Der Logik als eigener Disziplin lag auch immer schon der Gedanke der Berechenbarkeit logischer Schlüsse nahe, weshalb aus der Logik zu Beginn des 20. Jahrhunderts mit einer Theorie der Berechenbarkeit die theoretische Informatik entstanden ist, deutlich vor der Konstruktion der ersten Computer („Logik als historische Grundlage“).
- Die Sprache der Logik ist dazu geeignet, die Arbeitsweise von Programmen zu beschreiben, da eine Berechnung aus Eingaben einer Schlussfolgerung aus Prämissen entspricht.
- Die formalen Sprachen und die Methoden der Logik können helfen, Probleme aus der Informatik zu beschreiben oder zu lösen.

Die folgenden Namen aus der Geschichte der Logik sollten einem vielleicht vertraut sein:

– Aristoteles (384–322): Begründer der formalen Logik (*Syllogistik*)

– Ramon Llull (1232–1316): erste Idee einer „logischen Maschine“

– Gottfried Wilhelm Leibniz (1646–1716): Binärsystem, binäre Rechenmaschine, Idee einer „universellen Maschine“

– George Boole (1815–1864), Charles Sanders Peirce (1839–1914), Gottlob Frege (1848–1925): moderne symbolische/formalisierte Logik

– Emil Post (1897–1954), Alonzo Church (1903–1995), Stephen Kleene (1909–1994), Alan Turing (1912–1954): Begründer der Berechenbarkeitstheorie/theoretischen Informatik

– Kurt Gödel (1906–1978): bedeutende Ergebnisse mit dem Vollständigkeitssatz und den Unvollständigkeitssätzen

Hinweise zum Gebrauch des Skripts

Wichtige Begriffe sind unterstrichen. In *kursiver Schrift* (*oder in* gerader Schrift *innerhalb von kursivem Text*) sind Sprechweisen, Betonungen und nur vorübergehend gebrauchte oder weniger wichtige Begriffe markiert. Der Gebrauch der Farben wird jeweils bei Einführung bzw. ausführlich im Anhang erklärt.

1 Aussagenlogik

1.1 Syntax der Aussagenlogik

Aussagenlogische Formeln sind alle Zeichenfolgen (Zeichenketten), die nach den unten erläuterten Regeln aus den unten aufgeführten Zeichen (Symbolen) gebildet werden können. Der besseren Kenntlichkeit halber werde ich diese Zeichen blau schreiben (wie z. B. $\to$), wenn Sie in ihrer Funktion als einzelne Zeichen, aus denen Zeichenketten zusammengesetzt sind, angesprochen sind, und ebenso Zeichenketten, die keine aussagenlogischen Formeln sind. Aussagenlogische Formeln werde ich dagegen rot schreiben, wie z. B. $(A_0 \to A_1)$. Dabei kann es sein, dass die in Rot abgedruckte Zeichenfolge bereits die aussagenlogische Formel ist, aber auch, dass sie eine Abkürzung oder eine Variable für eine aussagenlogische Formel ist. [1]

Zeichen:

Aussagenvariablen	Junktoren		Klammern
A_0	nullstellig:	$\top$ und $\bot$	(
A_1	einstellig:	$\neg$	)
A_2	zweistellig:	$\wedge$, $\vee$, $\to$ und $\leftrightarrow$	
$\vdots$			

Es gibt unendlich viele Aussagenvariablen. Jede soll dabei als ein einziges Zeiches gelten, also z. B. A_{398652} als *ein* Symbol. Ein Ausdruck wie A_i wird als Variable für Aussagenvariablen verwendet, ohne selbst eine Aussagenvariable zu sein. Solche Variablen für Symbole schreibe ich ebenfalls blau, so wie Variablen für Formeln rot geschrieben werden.

Die Junktoren heißen der Reihe nach: *Verum*, *Falsum*, *Negations-*, *Konjunktions-*, *Disjunktions-*, *Implikations-* und *Äquivalenzjunktor*. In einer Formel werden die letzten fünf Junktoren oft gelesen als: *nicht – und – oder – wenn … dann – genau dann, wenn.*

Gleichzeitig mit den aussagenlogischen Formeln F wird ihre Länge $\lg(F)$ definiert und ihre Schachtelungstiefe $\mathrm{tf}(F)$, kurz Tiefe genannt. Die Länge ist dabei die Länge als Zeichenfolge.

Regeln:

- Jede Aussagenvariable bildet eine aussagenlogische Formel der Länge 1 und der Tiefe 0.
- $\top$ und $\bot$ sind aussagenlogische Formeln der Länge 1 und der Tiefe 1.
- Wenn F_1 und F_2 aussagenlogische Formeln sind, dann sind auch aussagenlogische Formeln: [2]

$\neg F_1$	mit Länge $\lg(F_1)+1$ und Tiefe $\mathrm{tf}(F_1)+1$
$(F_1 \wedge F_2)$ $(F_1 \vee F_2)$ $(F_1 \to F_2)$ $(F_1 \leftrightarrow F_2)$	mit Länge $\lg(F_1)+\lg(F_2)+3$ und Tiefe $\max\{\mathrm{tf}(F_1),\mathrm{tf}(F_2)\}+1$

[1] In meiner Farbverwendung überschreibt in der Regel eine komplexere Definition die einfachere: Wenn z. B. in der Definition einer Formel ein Symbol vorkommt, wird dieses nicht blau, sondern die ganze Formel rot sein.

[2] Mit $\neg F_1$ ist die Zeichenkette gemeint, die man erhält, indem man das Zeichen $\neg$ vor die mit F_1 bezeichnete Zeichenkette setzt, etc. Nicht gemeint ist die Zeichenkette, die aus den Symbolen $\neg$ und F_1 besteht! Variablen für aussagenlogische Formeln stehen stets für die Zeichenkette, welche die Formel bildet.

Die Menge der aussagenlogischen Formeln wird auch aussagenlogische Sprache genannt. Im Folgenden schreibe ich im oft kurz *Formel* für „aussagenlogische Formel“.

Die Notation $F(A_{i_1},\ldots,A_{i_k})$ für eine aussagenlogische Formel F soll bedeuten, dass $A_{i_1},\ldots,A_{i_k}$ paarweise verschieden sind und dass sich die in der Zeichenkette F vorkommenden Aussagenvariablen darunter befinden (aber nicht alle müssen vorkommen).

Beispiele: A_{398652} und $\bot$ sind aussagenlogische Formeln der Länge 1;
$(A_{398652}\to\bot)$ ist eine aussagenlogische Formel der Länge 5 und der Tiefe 2;
$\neg(A_{398652}\to\bot)$ ist eine Formel der Länge 6 und der Tiefe 3.
$(\neg A_{398652}\to\bot)$ ist eine andere Formel der Länge 6, aber der Tiefe 2.
$(\neg(A_{398652}\to\bot)\vee(\neg A_{398652}\to\bot))$ ist eine Formel der Länge 15 und der Tiefe 4.
Die Zeichenkette $A_{398652}\to\bot$ ist keine aussagenlogische Formel, ebensowenig $((A_{398652}\to\bot))$.

Aussagenlogische Formeln sind *induktiv* über ihre Tiefe (oder Länge) definiert. Formeln der Länge 1 heißen auch atomare Formeln, alle anderen zusammengesetzte Formeln. Per *Induktion über den Aufbau der Formeln* (also über Tiefe oder Länge) kann man Eigenschaften von Formeln definieren oder beweisen. Dies soll an zwei Beispielen geschehen.

(A) Eine Teilformel einer aussagenlogischen Formel F ist eine aussagenlogische Formel, die im Aufbauprozess von F auftritt. Genauer:

Definition 1.1.1 *Die Menge* $\mathrm{Tf}(F)$ *der* Teilformeln *von* F *und die Menge* $\mathrm{eTf}(F)$ *der* echten Teilformeln *von* F *sind per Induktion über den Aufbau von Formeln definiert durch:*

$$\begin{aligned}
&\mathrm{Tf}(F) := \mathrm{eTf}(F)\cup\{F\}\\
&\mathrm{eTf}(A_i) = \mathrm{eTf}(\top) = \mathrm{eTf}(\bot) := \emptyset\\
&\mathrm{eTf}(\neg F) := \mathrm{Tf}(F)\\
&\mathrm{eTf}\big((F_1\circ F_2)\big) := \mathrm{Tf}(F_1)\cup\mathrm{Tf}(F_2)\quad \textit{für jedes } \circ\in\{\wedge,\vee,\to,\leftrightarrow\}
\end{aligned}$$

(B) Aussagenlogische Formeln sind eindeutig lesbar, d. h. der Aufbauprozess einer Formel ist eindeutig aus ihr ablesbar. Dies wird durch Lemma 1.1.3 beschrieben. Wir brauchen zunächst eine Vorbereitung: Eine *Klammerung* sei hier eine Zeichenkette aus den Symbolen (und), die induktiv nach den folgenden Regeln gebildet werden kann:

- Die leere Zeichenfolge ist eine Klammerung.
- Wenn κ_1 und κ_2 Klammerungen sind, dann auch die Zeichenfolge $(\kappa_1\kappa_2)$.

Ist $\kappa = z_1\ldots z_n$ mit $z_i\in\{(,)\}$ eine Klammerung, dann ist $n=\lg(\kappa)$ die *Länge* von κ als Zeichenkette, und die *Klammerungstiefe* $\mathrm{Kt}(\kappa,i)$ von κ an der Stelle $i\in\{0,\ldots,n\}$ sei die Anzahl der öffnenden minus die Anzahl der schließenden Klammern in $z_1,\ldots,z_i$.

Beispiel:

Klammerung κ:		(	(	(	)	)	(	)	)	
$i=$	0	1	2	3	4	5	6	7	8	
$\mathrm{Kt}(\kappa,i)=$	0	1	2	3	2	1	2	1	0	

Klammerungen sind so definiert, dass man aus einer aussagenlogischen Formel F eine Klammerung $\kappa(F)$ erhält, wenn man aus F alle Zeichen bis auf die Klammern entfernt.

Lemma 1.1.2 *Für Klammerungen* κ *der Länge* n *gilt* $\mathrm{Kt}(\kappa,i)\geqslant 0$ *für alle* $i=0,\ldots,n$ *und* $\mathrm{Kt}(\kappa,i)=0 \iff i\in\{0,n\}$.
Für $n>0$ *und* $\kappa=(\kappa_1\kappa_2)$ *gilt* $\mathrm{Kt}(\kappa,i)=1 \iff i\in\{1,\lg(\kappa_1)+1,n-1\}$.

BEWEIS: Für die leere Klammerung und die Klammerung ( ) der Länge 2 sind die Eigenschaften offensichtlich. Sei also $\kappa = (\kappa_1\kappa_2)$ mit Klammerungen κ_1, κ_2 der Länge l bzw. $n-2-l$. Dann ist $\mathrm{Kt}(\kappa, i) = 1 + \mathrm{Kt}(\kappa_1, i-1)$ für $0 < i \leqslant l+1$ und $\mathrm{Kt}(\kappa, i) = 1 + \mathrm{Kt}(\kappa_2, i-(l+1))$ für $l+1 \leqslant i < n$. Per Induktion über die Länge der Klammerung folgt nun alles leicht. □

Lemma 1.1.3 (Eindeutige Lesbarkeit aussagenlogischer Formeln)
Eine zusammengesetzte Formel F ist entweder von der Form $\neg F_1$ oder von der Form $(F_1 \circ F_2)$. Dabei ist die Zerlegung jeweils eindeutig, d. h. falls $\neg F_1 = \neg F_1'$, dann ist $F_1 = F_1'$ und falls $(F_1 \circ F_2) = (F_1' * F_2')$ mit $\circ, * \in \{\wedge, \vee, \to, \leftrightarrow\}$, dann ist $\circ = *$, $F_1 = F_1'$ und $F_2 = F_2'$.

BEWEIS: Da F nach den Zusammensetzungsregeln aussagenlogischer Formeln entweder mit $\neg$ oder mit (beginnen muss und da man $\neg F_1$ im ersten Fall durch Weglassen des Negationsjunktors erhält, ist alles bis auf den letzten Teil offensichtlich. Sei also $(F_1 \circ F_2) = (F_1' * F_2')$ wie oben. Nun betrachtet man die zugrundeliegende Klammerung $\kappa = \kappa((F_1 \circ F_2)) = \kappa((F_1' * F_2'))$. Klar ist wiederum: F_1 beginnt mit (genau dann, wenn F_1' mit (beginnt, und F_2' endet mit) genau dann, wenn F_2' mit) endet. Wenn beides der Fall ist, zerlegt sich κ nach Lemma 1.1.2 in $(\kappa_1\kappa_2)$ mit $\kappa_1 \neq \emptyset \neq \kappa_2$, und die eindeutige Trennstelle von κ_1 und κ_2 gibt die Position von $\circ$ und von $*$ an. Ansonsten ist $\kappa = (\kappa')$ und die Klammerung κ' ist entweder die leere Klammerung – nämlich genau dann, wenn F_1 und F_2 atomar sind – oder $\kappa' = \kappa(F_1) = \kappa(F_1')$ – nämlich genau dann, wenn F_1 und F_1' mit (beginnen und F_2 und F_2' nicht mit) enden – oder $\kappa' = \kappa(F_2) = \kappa(F_2')$ im umgekehrten Fall. In jedem Fall stehen $\circ$ und $*$ an der gleichen Position und sind daher insbesondere auch gleich. □

In ähnlicher Art kann man die beiden folgenden Lemmata beweisen:

Lemma 1.1.4 *Kein echtes Anfangsstück einer aussagenlogischen Formel ist selbst eine aussagenlogische Formel.*

Lemma 1.1.5 (Substitution von Teilformeln)
Sind F' und F aussagenlogische Formeln und hat F die Gestalt $K^\frown F'^\frown K'$ für Zeichenketten K und K' dann ist F' eine Teilformel von F.[3]
Ist G' eine beliebige aussagenlogische Formel, dann ist auch $K^\frown G'^\frown K'$ eine Formel.

Beispiel: In der Formel $F = (\neg(A_0 \to \bot) \vee (A_2 \leftrightarrow A_0))$ ist $F' = (A_0 \to \bot)$ eine Teilformel. Ersetzt man diese in F durch z. B. $G' = ((\neg A_0 \to A_1) \vee \bot)$, so erhält man die Formel

$$(\neg((\neg A_0 \to A_1) \vee \bot) \vee (A_2 \leftrightarrow A_0)).$$

Bemerkung: (a) Die hier präsentierte Syntax der Aussagenlogik ist *eine mögliche Variante* einer aussagenlogische Sprache. In der Literatur findet man andere Varianten, die sich

- in den Symbolen für die Aussagenvariablen und Junktoren,
- in der Auswahl der Junktoren
- oder in der Klammerungsweise

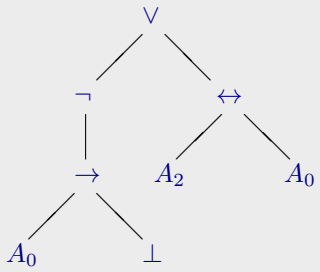
unterscheiden. Hier folgt die Schreibweise von Formeln mit zweistelligen Junktoren der Konvention z. B. bei einer Addition wie $3 + 2$ das Pluszeichen zwischen die Argumente zu setzen.

[3] Die *Konkatenation* $K^\frown K'$ bezeichnet die Zeichenkette, die durch Hintereinanderhängen von K und K' entsteht.

Eine grundsätzlich andere Möglichkeit orientiert sich an der Schreibweise $f(x,y)$ abstrakter Funktionen. Statt z. B. $(\neg A_{398652} \to \bot)$ schreibt man dann $\to(\neg(A_{398652}), \bot)$ bzw. klammer- und kommafrei $\to \neg A_{398652} \bot$ (polnische Notation), oder $A_{398652} \neg \bot \to$ (umgekehrte polnische Notation). Die polnische Notation erreicht die eindeutige Lesbarkeit ohne Klammern und erlaubt eine einheitliche Darstellung aller Junktoren, weshalb ich sie ab und zu in Beweisen verwendet werde. Sie hat aber den Nachteil einer für Menschen schlechteren Lesbarkeit.

(b) Eine aussagenlogische Formel ist eigentlich ein orientierter binärer Baum, dessen Blätter mit Aussagenvariablen, Verum und Falsum und dessen innere Knoten mit höherstelligen Junktoren etikettiert sind.

Beispiel: Die Formel $(\neg(A_0 \to \bot) \vee (A_2 \leftrightarrow A_0))$ entspricht dem Baum:



Die Anzahl der Knoten ist die Länge der Formel in polnischer Notation (also ohne Klammern); die Höhe des Baums ist im Wesentlichen die Tiefe der Formel (evtl. -1, falls die längsten Pfade alle mit $\top$ oder $\bot$ enden). Die eindeutige Lesbarkeit von Formeln bedeutet, dass man eine sequentielle Darstellung des zweidimensionalen Baumes gefunden hat, aus der sich der Baum rekonstruieren lässt.

Die Teilformeln entsprechen den maximalen Teilbäumen unterhalb von einem Knoten (der zur neuen Wurzel wird). Die in Lemma 1.1.5 bechriebene Substitution einer Teilformel durch eine andere Formel läuft darauf hinaus, dass man solch einen Teilbaum entfernt und durch einen anderen Baum einer Formel ersetzt.

1.2 Semantik der klassischen zweiwertigen Aussagenlogik

Die Menge der Wahrheitswerte sei $\{0,1\}$ als angeordnete Menge mit $0 < 1$. Der Wahrheitswert 0 heißt auch *falsch*, der Wahrheitswert 1 *wahr*. Bei Bedarf wird die Menge der Wahrheitswerte mit dem Körper $\mathbb{F}_2$ identifiziert, so dass die arithmetischen Operationen $+, -$ und $\cdot$ benutzt werden können. Eine Belegung der Aussagenvariablen mit Wahrheitswerten, kurz: Belegung, ist eine Abbildung $\beta : \{A_i \mid i \in \mathbb{N}\} \to \{0,1\}$.

Jede Belegung β wird mittels der nächsten Festlegung induktiv zu einer Funktion fortgesetzt, die jeder aussagenlogischen Formel F einen Wahrheitswert $\beta(F)$ zuordnet. („Auswertungen“ von Zeichen und Formeln schreibe ich grün.) Dazu wird zunächst jedem n-stelligen Junktor $*$ folgendermaßen eine n-stellige Funktion $\overline{*} : \{0,1\}^n \to \{0,1\}$ zugeordnet:

$$\overline{\top} := 1 \qquad \overline{\neg}(w) := 1 - w \qquad \overline{\wedge}(v,w) := \min\{v,w\} \qquad \overline{\to}(v,w) := \begin{cases} 1 & \text{falls } v \leqslant w \\ 0 & \text{falls } v > w \end{cases}$$

$$\overline{\bot} := 0 \qquad\qquad \overline{\vee}(v,w) := \max\{v,w\} \qquad \overline{\leftrightarrow}(v,w) := \begin{cases} 1 & \text{falls } v = w \\ 0 & \text{falls } v \neq w \end{cases}$$

Wenn nun eine aussagenlogische Formel F mithilfe eines n-stelligen Junktors $*$ aus Formeln $F_1,\ldots,F_n$ zusammengesetzt ist, so setzt man

$$\beta(F) := \overline{*}\big(\beta(F_1),\ldots,\beta(F_n)\big).$$

In polnischer Notation hätte man also $\beta(*F_1\ldots F_n) = \overline{*}\big(\beta(F_1),\ldots,\beta(F_n)\big)$.

Als Tabellen dargestellt haben die Junktoren also die folgende Wahrheitswertfunktionalität:

F_0	F_1	$(F_0\wedge F_1)$	$(F_0\vee F_1)$	$(F_0\to F_1)$	$(F_0\leftrightarrow F_1)$
0	0	0	0	1	1
0	1	0	1	1	0
1	0	0	1	0	0
1	1	1	1	1	1

F	$\neg F$
0	1
1	0

$\top$	$\bot$
1	0

Das Prinzip, dass sich der Wahrheitswert einer zusammengesetzten Formel F unabhängig von der konkreten Gestalt der Teilformeln nur aus dem führenden Junktor und den Wahrheitswerten der Teilformeln berechnet, heißt *Kompositionalitätsprinzip*. Es beinhaltet auch die *Kontextfreiheit*, die bedeutet, dass die Berechnung unabhängig davon ist, in welcher umfassenderen Formel F als Teilformel vorkommen mag.

Lemma 1.2.1 *Wenn β_1,β_2 zwei Belegungen sind, die auf allen in einer Formel F vorkommenden Aussagenvariablen übereinstimmen, dann gilt $\beta_1(F)=\beta_2(F)$.*

Beweis: Offensichtlich. Formaler Beweis über den Aufbau der Formel. □

Die Funktion $(\beta(A_i))_{i\in I}\mapsto\beta(F)$ nennt man auch den Wahrheitswertverlauf von F. Aus dem Lemma folgt, dass für eine Formel $F(A_{i_1},\ldots,A_{i_n})$ bereits eine *partielle Belegung* $\beta' : \{A_{i_1},\ldots,A_{i_n}\}\to\{0,1\}$ der in F vorkommenden Aussagenvariablen einen eindeutigen Wahrheitswert $\beta'(F)$ bestimmt. Der Wahrheitswertverlauf von F ist daher bei n vorkommenden Aussagenvariablen durch 2^n partielle Belegungen bestimmt und kann übersichtlich in einer *Wahrheitstafel* (oder *Wahrheitstabelle*) dargestellt werden.

Beispiel: Für $F=((\neg A_0\to A_1)\vee\bot)$ bekommt man die Wahrheitstafel:

(*ausführliche Version*)

A_0	A_1	$\neg A_0$	$(\neg A_0\to A_1)$	$\bot$	$((\neg A_0\to A_1)\vee\bot)$
0	0	1	0	0	0
0	1	1	1	0	1
1	0	0	1	0	1
1	1	0	1	0	1

(*kompakte Version*)

$((\neg$	A_0	$\to$	$A_1)$	$\vee$	$\bot)$
1	0	0	0	0	0
1	0	1	1	1	0
0	1	1	0	1	0
0	1	1	1	1	0

Definition 1.2.2 **(a)** *Eine aussagenlogische Formel F heißt Tautologie oder allgemeingültig, in Zeichen: $\vdash F$, falls $\beta(F)=1$ für alle Belegungen β.*

F heißt erfüllbar oder konsistent, falls es eine Belegung β gibt mit $\beta(F)=1$.

(b) *Zwei aussagenlogische Formeln F_1, F_2 heißen (logisch) äquivalent zueinander, in Zeichen: $F_1\sim F_2$, falls $\beta(F_1)=\beta(F_2)$ für alle Belegungen β.*

(c) *Eine aussagenlogische Formel* F *wird von einer Menge von Formeln* $\{F_i \mid i \in I\}$ *impliziert (oder folgt (logisch) aus dieser Menge), in Zeichen:* $\{F_i \mid i \in I\} \vdash F$*, falls* $\beta(F) = 1$ *für jede Belegung* β*, die für alle* $i \in I$ *die Bedingung* $\beta(F_i) = 1$ *erfüllt.*

(d) *Eine Menge von Formeln* $\{F_i \mid i \in I\}$ *heißt widerspruchsfrei oder erfüllbar, falls es eine Belegung* β *gibt mit* $\beta(F_i) = 1$ *für alle* $i \in I$*, und andernsfalls widersprüchlich.*

Für $\{F_1, \ldots, F_n\} \vdash F$ schreibt man auch kurz $F_1, \ldots, F_n \vdash F$.

Man beachte, dass sich aus der Definition das *ex-falso-quodlibet*-Prinzip ergibt: Jede beliebige Formel F folgt aus einer widersprüchlichen Formelmenge! Insbesondere gilt $\bot \vdash F$.

Lemma 1.2.3 (Zusammenhänge)

(a) $\vdash F \iff \emptyset \vdash F \iff \top \vdash F \iff F \sim \top \iff \neg F$ *ist nicht erfüllbar.*

(b) $F_1 \sim F_2 \iff \vdash (F_1 \leftrightarrow F_2) \iff (F_1 \vdash F_2$ *und* $F_2 \vdash F_1)$

(c) $F_1, \ldots, F_n \vdash F \iff ((\cdot\cdot\,(F_1 \wedge F_2) \wedge \cdots) \wedge F_n) \vdash F$
$\iff \vdash (((\cdot\cdot\,(F_1 \wedge F_2) \wedge \cdots) \wedge F_n) \to F)$

(d) $\{F_1, \ldots, F_n\}$ *ist erfüllbar* $\iff ((\cdot\cdot\,(F_1 \wedge F_2) \wedge \cdots) \wedge F_n)$ *ist erfüllbar*

$\{F_1, \ldots, F_n\}$ *ist nicht erfüllbar* $\iff ((\cdot\cdot\,(F_1 \wedge F_2) \wedge \cdots) \wedge F_n) \sim \bot$
$\iff F_1, \ldots, F_n \vdash \bot$

Beweis: Übung (folgt unmittelbar aus den Definitionen). □

Die Gültigkeit einer logischen Äquivalenz oder Folgerung bzw. die Tatsache, dass eine Formel eine Tautologie ist, nenne ich *logisches Gesetz* (oder *Regel* – in der Aussagenlogik verwende ich beide Begriffe synonym).

1.3 Gesetze der klassischen zweiwertigen Aussagenlogik

Aus Lemma 1.2.3 folgt, dass alles, was sich über endliche viele Formeln mit den in Definition 1.2.2 definierten Begriffen aussagen lässt, in Form einer logischen Äquivalenz ausdrücken lässt. Da es unendlich viele Formeln in n fest gewählten Aussagenvariablen gibt, dafür aber nur 2^n mögliche Wahrheitswertverläufe, gibt es viele Äquvalenzen zwischen Formeln. Solch eine Äquivalenz kann man als eine „Umformungsregel“ auffassen. Alle Äquvalenzen lassen sich auf eine Reihe von elementaren Umformungen zurückführen.

Erinnerung: Wenn F' eine Teilformel von $F = K\hat{}F'\hat{}K'$ ist und G' eine weitere Formel, dann ist auch $G = K\hat{}G'\hat{}K'$ eine Formel (Lemma 1.1.5). Folglich ist auch die Zeichenkette, die aus F entsteht, indem jedes Vorkommen einer festen Aussagenvariable A_i in F gleichzeitig durch eine Formel H ersetzt wird, wieder eine Formel. Diese wird mit $F\frac{H}{A_i}$ bezeichnet.

Achtung: H darf selbst wieder die Aussagenvariable A_i enthalten!

Beispiel: In der Formel $F = (\neg(A_0 \to \bot) \vee (A_2 \leftrightarrow A_0))$ wird A_0 einmal durch $(A_1 \to A_2)$ und einmal durch $\neg A_0$ ersetzt. Dies ergibt die Formeln

$$F\frac{(A_1 \to A_2)}{A_0} = (\neg((A_1 \to A_2) \to \bot) \vee (A_2 \leftrightarrow (A_1 \to A_2)))$$
$$F\frac{\neg A_0}{A_0} = (\neg(\neg A_0 \to \bot) \vee (A_2 \leftrightarrow \neg A_0))$$

Lemma 1.3.1 (Aussagenlogisches Substitutionsprinzipien)
Sei $F \sim G$. Dann ist $F\frac{H}{A_i} \sim G\frac{H}{A_i}$ (uniforme Substitution) und $K\hat{}F\hat{}K' \sim K\hat{}G\hat{}K'$, sofern dies Formeln sind (äquivalente Substitution). Analog für $\vdash$ anstelle von $\sim$.

Beweis: Beide Prinzipien folgen sofort aus der Kompositionalität, da es nur auf die Wahrheitswerte von A_i bzw. F und G ankommt. □

Beispiel: Aus $(A_0 \vee \neg A_0) \sim \top$ folgt mit $A_0 \sim \neg\neg A_0$ und äquivalenter Substitution des zweiten Vorkommens von A_0 die Äquivalenz $(A_0 \vee \neg\neg\neg A_0) \sim \top$, und mit uniformer Substitution von A_0 durch $(A_1 \to A_2)$ folgt $((A_1 \to A_2) \vee \neg(A_1 \to A_2)) \sim \top$.

Satz 1.3.2 *Es gelten folgende elementaren Regeln, deren Gültigkeit man leicht anhand von Wahrheitstafeln nachprüft. Dabei stehen A, B, C als Variable für Aussagenvariablen (bzw. wegen des Prinzips der äquivalenten Substitution auch für Formeln).*

$\top$-/$\bot$-Regeln:

$\neg\top \sim \bot$	$\neg\bot \sim \top$	
$(A \wedge \top) \sim A$	$(A \vee \top) \sim \top$	$(\top \to A) \sim A$
$(A \wedge \bot) \sim \bot$	$(A \vee \bot) \sim A$	$(A \to \bot) \sim \neg A$

$\neg$-Regeln:

$\neg\neg A \sim A$	*Doppelnegationsregel*
$(A \vee \neg A) \sim \top$	*Prinzip des ausgeschlossenen Dritten*
$(A \wedge \neg A) \sim \bot$	*Prinzip des ausgeschlossenen Widerspruchs*

$\wedge$-/$\vee$-Regeln:

$(A \wedge A) \sim A$	$(A \vee A) \sim A$	*Idemp.*
$(A \wedge B) \sim (B \wedge A)$	$(A \vee B) \sim (B \vee A)$	*Komm.*
$((A \wedge B) \wedge C) \sim (A \wedge (B \wedge C))$	$((A \vee B) \vee C) \sim (A \vee (B \vee C))$	*Ass.*
$((A \wedge B) \vee C) \sim ((A \vee C) \wedge (B \vee C))$	$((A \vee B) \wedge C) \sim ((A \wedge C) \vee (B \wedge C))$	*Distr.*

Regeln von *de Morgan*:

$\neg(A \wedge B) \sim (\neg A \vee \neg B)$	$\neg(A \vee B) \sim (\neg A \wedge \neg B)$

Definition von $\to$ bzw. $\leftrightarrow$:

$(A \to B) \sim (\neg A \vee B)$	$(A \leftrightarrow B) \sim ((A \to B) \wedge (B \to A))$

Die $\wedge$-/$\vee$-Regeln heißen der Reihe nach: *Idempotenz*, *Kommutativität* und *Assoziativität von $\wedge$ bzw. $\vee$* sowie *Distributivität von $\vee$ über $\wedge$* bzw. umgekehrt.

Nützlich zu kennen sind außerdem folgende Gesetze:

Absorption	$((A \wedge B) \vee A) \sim A$	$((A \vee B) \wedge A) \sim A$
Kontraposition	$(A \to B) \sim (\neg B \to \neg A)$	$(A \leftrightarrow B) \sim (\neg B \leftrightarrow \neg A)$
„Currying“	$((A \wedge B) \to C) \sim (A \to (B \to C))$	
Transitivität von $\to$	$(A \to B), (B \to C) \vdash (A \to C)$	
(*modus ponens*	$B, (B \to C) \vdash C$ mit $A = \top$)	
(*modus tollens*	$(A \to B), \neg B \vdash \neg A$ mit $C = \bot$)	
Links-Distributivität	$(A \to (B \wedge C)) \sim ((A \to B) \wedge (A \to C))$	
von $\to$ *über* $\wedge$ *bzw.* $\vee$	$(A \to (B \vee C)) \sim ((A \to B) \vee (A \to C))$	
Rechts-Antidistributivität	$((A \wedge B) \to C) \sim ((A \to C) \vee (B \to C))$	
von $\to$ *über* $\wedge$ *bzw.* $\vee$	$((A \vee B) \to C) \sim ((A \to C) \wedge (B \to C))$	

Abkürzungen: Im weiteren Verlauf werden folgende abkürzende Schreibweisen verwendet:

$$(F_1 \wedge F_2 \wedge \cdots \wedge F_n) \quad \text{oder} \quad \bigwedge_{i=1}^{n} F_i \quad \text{für die Formel} \quad ((\cdot\cdot (F_1 \wedge F_2) \wedge \cdots) \wedge F_n)$$

$$(F_1 \vee F_2 \vee \cdots \vee F_n) \quad \text{oder} \quad \bigvee_{i=1}^{n} F_i \quad \text{für die Formel} \quad ((\cdot\cdot (F_1 \vee F_2) \vee \cdots) \vee F_n)$$

Für $n = 1$ sei dabei $\bigwedge_{i=1}^{1} F_i = \bigvee_{i=1}^{1} F_i = F_1$; für $n = 0$ sei $\bigwedge_{i \in \emptyset} F_i = \top$ und $\bigvee_{i \in \emptyset} F_i = \bot$.

Diese Sonderfälle sind sinnvoll, weil nun z. B. die Äquivalenz

$$\bigwedge_{i=1}^{n} F_i \sim (\bigwedge_{i=1}^{k} F_i \wedge \bigwedge_{i=k+1}^{n} F_i)$$

für alle $k = 0, \ldots, n$ gilt, analog für die Disjunktion.

Häufig werden aussagenlogische Formeln nur bis auf logische Äquivalenz betrachtet. Dann spielt die Reihenfolge und die Klammerungsreihenfolge innerhalb einer Konjunktion bzw. Disjunktion keine Rolle, da sich bei anderer Reihenfolge logisch äquivalente Formeln ergeben. In diesem Sinn kann man auch Mengennotationen wie $\bigwedge\{F_i \mid i \in I\}$ etc. einführen. dann gilt allgemeiner für Indexmengen I und I' zum Beispiel:

$$\bigwedge_{i \in I \cup I'} F_i \sim (\bigwedge_{i \in I} F_i \wedge \bigwedge_{i \in I'} F_i).$$

Definition 1.3.3 *Ein Literal ist eine aussgenlogische Formel, die entweder eine Aussagenvariable A_i oder eine negierte Aussagenvariable $\neg A_i$ ist.*

Eine Formel ist in disjunktiver Normalform (DNF), wenn sie eine Disjunktion sogenannter Konjunktionsterme ist, d. h. mit Literalen L_{ij} und $k, n_1, \ldots, n_k \in \mathbb{N}$ von der Form

$$((L_{11} \wedge \cdots \wedge L_{1n_1}) \vee \cdots \vee (L_{k1} \wedge \cdots \wedge L_{kn_k})).$$

Eine Formel ist in konjunktiver Normalform (KNF), wenn sie eine Konjunktion sogenannter Disjunktionsterme oder Klauseln ist, d. h. (mit L_{ij} etc. wie oben) von der Form

$$((L_{11} \vee \cdots \vee L_{1n_1}) \wedge \cdots \wedge (L_{k1} \vee \cdots \vee L_{kn_k})).$$

Insbesondere sind also $\top$ und $\bot$ in DNF und in KNF (Fall $k = 0$ bzw. $k = 1$ und $n_1 = 0$).

KNF und DNF sind „dual“ zueinander [4], insbesondere wird die Negation einer Formeln in DNF durch Anwenden der Formeln von *de Morgan* und Elimination von Doppelnegationen zu einer Formel in KNF und umgekeht.

Satz 1.3.4 *Jede Formel ist logisch äquivalent zu einer Formel in disjunktiver Normalform und logisch äquivalent zu einer Formel in konjunktiver Normalform.*

Beispiel: Der folgende erste Beweis soll zunächst an einem Beispiel veranschaulich werden. Für die Formel F mit dem unten stehenden Wahrheitswewrtverlauf soll eine äquivalenten Formel in DNF gefunden werden. Zunächst wird für jede 1 im Wahrheitswertverlauf von F eine Formel konstruiert, die genau an dieser Stelle den Wahrheitswert 1 hat und sonst überall 0. Dazu modifiziert man die Konjunktion der vorkommenden Aussagenvariablen durch Negationen an den notwendigen Stellen:

A_0	A_1	A_2	F	$(A_0 \wedge A_1 \wedge A_2)$	$(A_0 \wedge \neg A_1 \wedge A_2)$	$(\neg A_0 \wedge \neg A_1 \wedge A_2)$
0	0	0	0	0	0	0
0	0	1	1	0	0	1
0	1	0	0	0	0	0
0	1	1	0	0	0	0
1	0	0	0	0	0	0
1	0	1	1	0	1	0
1	1	0	0	0	0	0
1	1	1	1	1	0	0

Nun ist F äquivalent zur Disjunktion dieser Formeln, also

$$F \sim ((A_0 \wedge A_1 \wedge A_2) \vee (A_0 \wedge \neg A_1 \wedge A_2) \vee (\neg A_0 \wedge \neg A_1 \wedge A_2)).$$

Erster Beweis durch Kodieren der Wahrheitstafel (am Beispiel der DNF): Ohne Einschränkung kommen in F genau die Aussagenvariablen $A_0, \ldots, A_n$ vor. Für Belegungen β definiert man das Symbol $(\neg)_\beta A_i$ als für folgende Formel stehend:

$$(\neg)_\beta A_i := \begin{cases} A_i & \text{falls } \beta(A_i) = 1 \\ \neg A_i & \text{falls } \beta(A_i) = 0 \end{cases}$$

Für eine Belegung β' gilt nun genau dann $\beta'\left(\bigwedge_{i=0}^{n}(\neg)_\beta A_i\right) = 1$, wenn β' und β auf $A_0, \ldots, A_n$ übereinstimmen. Es folgt:

$$F \sim \bigvee_{\beta(F)=1} \bigwedge_{i=0}^{n} (\neg)_\beta A_i$$

Durch die „duale Konstruktion“ dazu bekommt man die KNF als $F \sim \bigwedge_{\beta(F)=0} \bigvee_{i=0}^{n} \neg(\neg)_\beta A_i$.

Alternativ erhält man diese Formel dadurch, dass man $\neg F$ in DNF bringt, dann diese DNF negiert und durch Hineinziehen der Negation in KNF umformt. □

Die im diesem Beweis konstruierte disjunktive Normalform hat die Eigenschaft, dass in jedem Konjunktionsterm jede der in F vorkommenden Aussagenvariablen genau einmal vorkommt.

[4] Näheres zum Konzept der Dualität siehe Abschnitt 1.4.

Sie ist bis auf Reihenfolge der Konjunktionsterme, Reihenfolge der Literale innerhalb der Konjunktionsterme und Klammerungsreihenfolge eindeutig. Durch Konventionen für die Reihenfolgen (z. B. Linksklammerung, Aussagenvariable innerhalb der Konjunktionsterme nach Indizes aufsteigend geordnet und Belegungen lexikographisch) erhält man eine eindeutige DNF, die kanonische disjunktive Normalform. (Analog für die KNF.)

Skizze des zweiten Beweises: Durch die elementaren Regeln aus Satz 1.3.2 kann man eine Formel F mit folgendem Verfahren sukzessive in kanonische DNF umformen:

1. Ersetze alle Junktoren $\leftrightarrow$ und dann alle Junktoren $\rightarrow$ durch ihre Definition.
2. Vereinfache alle Vorkommen von $\top$ und $\bot$ mit Hilfe der $\top$-/$\bot$-Regeln, bis nur noch eine Formel $\top$ oder $\bot$ übrig bleibt oder kein $\top$ oder $\bot$ mehr vorkommt.
3. Ziehe mit den *de Morgan*'schen Regeln alle Negationen „nach innen“ bis unmittelbar vor die Aussagenvariablen.
4. Wende solange das Distributivgesetz (von $\wedge$ über $\vee$) an, bis die Formel DNF hat.
5. Durch Kommutativität und Assoziativität können Konjunktionsterme beliebig sortiert werden, durch Idempotenz doppelte Vorkommen von Literalen in einem Konjunktionsterm eliminiert werden.
6. Ersetze gleichzeitiges Vorkommen von A_i und $\neg A_i$ in einem Konjunktionsterm durch $\bot$ (Prinzip des ausgeschlossenen Widerspruchs) und führe wieder den 2. Schritt durch. Dabei bleibt die Formel in DNF.
7. Ersetz ggf. einen Konjunktionsterm K durch $(K \wedge \top) \sim (K \wedge (A_i \vee \neg A_i))$ und verwende erneut das Distributivgesetz, um wieder DNF zu erreichen, bis in allen Konjunktionstermen alle Aussagenvariablen von F vorkommen.

Man muss sich nun noch davon überzeugen, dass das Verfahren stoppt und das gewünschte Ergebnis liefert. (Ein analoges Verfahren funktioniert für die KNF.) □

Beispiel: (a) $(A_1 \rightarrow A_0)$ ist logisch äquivalent zu $(A_0 \vee \neg A_1)$, was bereits eine Formel in DNF und in kanonischer KNF ist. Die kanonische DNF ist $((\neg A_0 \wedge \neg A_1) \vee (A_0 \wedge \neg A_1) \vee (A_0 \wedge A_1))$.

(b) Für die *dreistellige Paritätsformel* $F = \neg((A_0 \leftrightarrow A_1) \leftrightarrow A_2)$ ist

die Wahrheitstafel:

A_0	A_1	A_2	F
0	0	0	1
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	0

die kanonische DNF:

$$(((((\neg A_0 \wedge \neg A_1) \wedge \neg A_2) \vee ((\neg A_0 \wedge A_1) \wedge A_2)) \vee ((A_0 \wedge \neg A_1) \wedge A_2)) \vee ((A_0 \wedge A_1) \wedge \neg A_2))$$

und die kanonische KNF:

$$(((((A_0 \vee A_1) \vee \neg A_2) \wedge ((A_0 \vee \neg A_1) \vee A_2)) \wedge ((\neg A_0 \vee A_1) \vee A_2)) \wedge ((\neg A_0 \vee \neg A_1) \vee \neg A_2))$$

Die Konjunktions- bzw. Disjunktionsterme entsprechen der Reihe nach den Zeilen mit Wahrheitswert 1 bzw. 0, wobei bei der DNF die Negationen bei den Aussagenvariablen mit Wahrheitswert 0, bei der KNF dagegen bei denen mit Wahrheitswert 1 stehen.

(c) Ein Beispiel für eine KNF durch Umformungen findet man auf Seite 20.

Folgerung 1.3.5 **(a)** *Alle logischen Äquivalenzen folgen aus den elementaren Regeln aus Satz 1.3.2 und den Substitutionsprinzipien aus Lemma 1.3.1.*

(b) *Zu jedem n und jeder Funktion $f : \{0,1\}^n \to \{0,1\}$ gibt es eine aussagenlogische Formel $F(A_0, \ldots, A_{n-1})$ mit f „als Wahrheitswertverlauf“, d. h. für alle Belegungen β gilt $\beta(F) = f\big(\beta(A_0), \ldots, \beta(A_{n-1})\big)$.*

Beweis: (a) Äquivalente Formeln haben die gleiche kanonische DNF. Man kann sie also nach dem zweiten Beweis von Satz 1.3.4 mit den elementaren Regeln über die kanonische DNF ineinander überführen. (b) folgt unmitelbar aus den Konstruktion der DNF aus einem Wahrheitswertverlauf im ersten Beweis von Satz 1.3.4. □

Definition 1.3.6 *Eine Menge von Junktoren $\mathcal{J}$ heißt vollständiges Junktorensystem, falls es zu jedem n und jeder n-stelligen Funktion $f : \{0,1\}^n \to \{0,1\}$ eine aussagenlogische Formel mit diesem Wahrheitswertverlauf gibt (in dem in 1.3.5 (b) präzisierten Sinn), in der nur Junktoren aus $\mathcal{J}$ vorkommen.*

Insbesondere ist jede aussagenlogische Formel logisch äquivalent zu einer Formel, in der alle vorkommenden Junktoren aus dem vollständigem Junktorensystem $\mathcal{J}$ stammen.

Satz 1.3.7 *$\{\neg, \wedge\}$ und $\{\neg, \vee\}$ bilden vollständige Junktorensysteme.*

Beweis: Aus dem ersten Beweis für die Existenz der DNF folgt sofort, dass $\{\neg, \vee, \wedge, \bot, \top\}$ ein vollständiges Junktorensystem ist. Aus $\bot \sim \neg\top$, $\top \sim (A_0 \vee \neg A_0)$ und den substitutierten Regeln von *de Morgan*, also z. B. $(A_0 \wedge A_1) \sim \neg(\neg A_0 \vee \neg A_1)$, folgt dann der Satz. □

Man kann die Anforderung an ein vollständiges Junktorensystem $\mathcal{J}$ dahingehend verstärken, dass jede aussagenlogische Formel F zu einer Formel äquivalent sein soll, in der nur Junktoren aus $\mathcal{J}$ vorkommen und keine weiteren Aussagenvariablen als die in F vorkommenden. Dann braucht man zusätzlich zu $\{\neg, \wedge\}$ bzw. $\{\neg, \vee\}$ noch jeweils $\top$ oder $\bot$.

Beispiel: $\{\to, \bot\}$ ist ein weiteres vollständiges Junktorensystems im stärkeren Sinne.

Man kann das Konzept auch auf „neue“ Junktoren ausdehnen, die man zur Sprache hinzunimmt, und erhält mit NOR $(A_0 \downarrow A_1) :\sim \neg(A_0 \vee A_1)$ und NAND $(A_0 \uparrow A_1) :\sim \neg(A_0 \wedge A_1)$ vollständige Junktorensysteme $\{\downarrow\}$ und $\{\uparrow\}$ im Sinne der ursprünglichen Definition.[5] Dies sind die einzigen vollständigen Junktorensysteme, die aus einem einzelnen Junktor der Stelligkeit höchstens 2 bestehen.

1.4 Boole’sche Algebren

Logische Äquivalenz ist eine Äquivalenzrelation auf der Menge der aussagenlogischen Formeln. Die Äquivalenzklasse einer Formel F bezeichne ich mit $F/_\sim$. (Ein Repräsentantensystem der Äquivalenzklassen ist z. B. durch die Formeln in kanonischer disjunktiver Normalformen mit jeweils minimaler Anzahl an vorkommenden Aussagenvariablen gegeben.)

Aus den Substitutionsregeln folgt, dass man die Junktoren als Verknüpfungen auf den Äquivalenzklassen definieren kann; z. B. ist $(F/_\sim) \wedge (G/_\sim) := (F \wedge G)/_\sim$ wohldefiniert und $\wedge$ wird

[5] NAND wird auch $|$ geschrieben und *Sheffer-Strich* genannt; NOR heißt auch *Peirce-Funtion oder -Operator*.

somit zu einer kommutativen und assoziativen zweistelligen Verknüpfung auf den Äquivalenzklassen, ebenso $\vee$. Die Menge der $\mathcal{F}_\infty$ der Äquivalenzklassen aussagenlogischer Formeln zusammen mit den Operationen $\wedge$ und $\vee$ ist eine Struktur, die Tarski-Lindenbaum-Algebra heißt und ein Beispiel einer *Boole'schen Algebra* ist. Üblicherweise nimmt man noch die aus der Negation definierte einstellige Verknüpfung $\neg\,(F/_\sim) := \neg F/_\sim$ und die Äquivalenzklassen von $\top$ und $\bot$ als Konstanten hinzu. Schränkt man die Grundmenge ein auf die Äquivalenzklassen von Formeln, in denen nur die Aussagenvariablen $A_0,\ldots,A_{n-1}$ vorkommen, erhält man die Tarski-Lindenbaum-Algebra $\mathcal{F}_n$.[6]

Definition 1.4.1 *Eine Boole'sche Algebra* $\mathfrak{B} = (B; \sqcap, \sqcup, {}^{\mathrm{c}}, 1, 0)$ *ist eine Struktur mit zwei zweistelligen Verknüpfungen* $\sqcap$ *und* $\sqcup$*, einer einstelligen Verknüpfung* $^{\mathrm{c}}$ *und zwei Konstanten* 1 *und* 0*, für die gelten:*

Idemp., Komm., Ass.:	$\sqcap$ *und* $\sqcup$ *sind idempotent, kommutativ und assoziativ*	
Distributivgesetze:	$a \sqcup (b \sqcap c) = (a \sqcup b) \sqcap (a \sqcup c)$	$a \sqcap (b \sqcup c) = (a \sqcap b) \sqcup (a \sqcap c)$
Absorptionsgesetze:	$a \sqcup (a \sqcap b) = a$	$a \sqcap (a \sqcup b) = a$
de Morgan'sche Regeln:	$(a \sqcap b)^{\mathrm{c}} = a^{\mathrm{c}} \sqcup b^{\mathrm{c}}$	$(a \sqcup b)^{\mathrm{c}} = a^{\mathrm{c}} \sqcap b^{\mathrm{c}}$
Doppelnegation:	$a^{\mathrm{cc}} = a$	
Extremalgesetze:	$a \sqcap 1 = a \sqcup 0 = a$	$a \sqcap 0 = 0,\ \ a \sqcup 1 = 1$
Komplementgesetze:	$a \sqcap a^{\mathrm{c}} = 0$, *insbesondere* $1^{\mathrm{c}} = 0$	$a \sqcup a^{\mathrm{c}} = 1$, *insbesondere* $0^{\mathrm{c}} = 1$

Dies ist eine traditionelle, aber nicht unabhängige Liste von Axiomen. Zum Beispiel kann man die *de Morgan*'schen Regeln und die Absorptionsgesetze aus den restlichen Axiomen herleiten.

Beispiel: Herleitung des ersten Absorptionsgesetzes aus den anderen Axiomen:

$a \sqcup (a \sqcap b)$	$= (a \sqcap 1) \sqcup (a \sqcap b)$	Extremalgesetz
	$= (a \sqcap (b \sqcup b^{\mathrm{c}})) \sqcup (a \sqcap b)$	Komplementgesetz
	$= ((a \sqcap b) \sqcup (a \sqcap b^{\mathrm{c}})) \sqcup (a \sqcap b)$	Distributivgesetz
	$= ((a \sqcap b) \sqcup (a \sqcap b)) \sqcup (a \sqcap b^{\mathrm{c}})$	Kommutativ- und Assoziativgesetze
	$= (a \sqcap b) \sqcup (a \sqcap b^{\mathrm{c}})$	Idempotenzgesetz
	$= a \sqcap (b \sqcup b^{\mathrm{c}})$	Distributivgesetz
	$= a \sqcap 1$	Extremalgesetz
	$= a$	Extremalgesetz

Eine Struktur mit zwei idempotenten, kommutativen und assoziativen Verknüpfungen $\sqcap$ und $\sqcup$, die die Absorptionsgesetze erfüllen, heißt Verband. Verbände kann man durch sogenannte Hasse-Diagramme darstellen (wie z. B. auf Seite 16). Gelten die Distributivgesetze, heißt der Verband distributiver Verband; gibt es Elemente 0 und 1, die die Extremalgesetze erfüllen, spricht man von einem beschränkten Verband. In einem Verband wird durch

$$a \sqsubseteq b :\iff a \sqcup b = b \iff a \sqcap b = a$$

eine partielle Ordnung definiert. Darin ist $a \sqcap b$ das *Infimum* zweier Elemente a und b (also das größte Element x, das $x \sqsubseteq a$ und $x \sqsubseteq b$ erfüllt), und $a \sqcup b$ das *Supremum* (also das kleinste Element x, das $a \sqsubseteq x$ und $b \sqsubseteq x$ erfüllt). Ist der Verband beschränkt, dann ist 0 das kleinste und 1 das größte Element der Ordnung. Umgekehrt ist jede partielle Ordnung mit

[6] Man kann sich überlegen, dass $|\mathcal{F}_n| = 2^{2^n}$.

diesen Eigenschaften ein beschränkter Verband. Ist der Verband auch distributiv und existieren Komplemente (d. h. zu jedem a ein Element a^{c}, so dass die Komplementgesetze gelten), dann ist die partielle Ordnung eine Boole'esche Algebra.

$\mathcal{F}_\infty$ und $\mathcal{F}_n$ sind mit den Verkmüpfungen $\wedge, \vee, \neg$ und den Konstanten $\top/_\sim$ und $\bot/_\sim$ Boole'sche Algebren. Die partielle Ordnung wird durch die logische Folgerung bzw. den Junktor $\to$ beschrieben, genauer

$$F/_\sim \sqsubseteq G/_\sim \iff F \vdash G \iff \vdash (F \to G).$$

Andere Beispiele Boole'scher Algebren sind die Potenzmengenalgebren. Zu einer Menge M sei $\mathfrak{P}(M)$ die Potenzmenge von M, also die Menge aller Teilmengen von M; und für $X \subseteq M$ sei $X^{\mathrm{c}} := M \setminus X$ das mengentheoretische Komplement in M. Dann ist $(\mathfrak{P}(M); \cap, \cup, {}^{\mathrm{c}}, M, \emptyset)$. eine Boole'sche Algebra.

Eine Unteralgebra einer Boole'schen Algebra ist eine Teilmenge U, die 0 und 1 enthält und unter den Operationen $\sqcap$, $\sqcup$ und ${}^{\mathrm{c}}$ abgeschlossen ist. U ist dann mit den eingeschränkten Operationen selbst eine Boole'sche Algebra. Ein Homomorphismus zwischen Boole'schen Algebren $\mathfrak{B}_1$ und $\mathfrak{B}_2$ ist eine Abbildung $h : B_1 \to B_2$, die mit allen Konstanten und Operationen verträglich ist, für die also gilt:[7]

$$h(a \sqcap_{\mathfrak{B}_1} b) = h(a) \sqcap_{\mathfrak{B}_2} h(b) \quad \text{und} \quad h(a \sqcup_{\mathfrak{B}_1} b) = h(a) \sqcup_{\mathfrak{B}_2} h(b)$$
$$h(a^{\mathrm{c}_{\mathfrak{B}_1}}) = h(a)^{\mathrm{c}_{\mathfrak{B}_2}} \quad , \quad h(0_{\mathfrak{B}_1}) = 0_{\mathfrak{B}_2} \quad \text{und} \quad h(1_{\mathfrak{B}_1}) = 1_{\mathfrak{B}_2}$$

Ein *Isomorphismus* zwischen Boole'schen Algebren ist ein bijektiver Homomorphismus.

Die Menge der Wahrheitswerte $\{0, 1\}$ mit $0 < 1$ ist eine Boole'sche Algebra (die isomorph zu $\mathfrak{P}(\emptyset)$ und zu $\mathcal{F}_0$ ist). Die Auswertung von Formeln ist gerade so definiert, dass eine Belegung β einen Homomorphismus Boole'scher Algebren $\beta : \mathcal{F}_\infty \to \{0, 1\}$, $F/_\sim \mapsto \beta(F)$ ergibt. Da eine Formel bis auf logische Äquivalenz durch ihren Wahrheitwertverlauf bestimmt ist, kann man sie mit der Menge der Belegungen, die sie wahr macht, identifizieren. Dies funktioniert sehr viel allgemeiner:

Satz 1.4.2 (Darstellungssatz von Stone, 1936)
Jede Boole'sche Algebra ist isomorph zu einer Unteralgebra einer Potenzmengenalgbra.

Beweisskizze: Sei $\mathfrak{B}$ eine beliebige Boole'sche Algebra. Man betrachtet

$$M := \big\{ h : \mathfrak{B} \to \{0, 1\} \;\big|\; h \text{ ist Homomorphismus Boole'scher Algebren} \big\}.$$

Ein Element von $\mathfrak{B}$ wird nun mit der Menge der Homomorphismen identifiziert, die das Element auf 1 abbilden:

$$S : \mathfrak{B} \to \mathfrak{P}(M), \quad S(b) := \big\{ h : \mathfrak{B} \to \{0, 1\} \;\big|\; h(b) = 1 \big\}$$

Die so definierte Abbildung S ist ein injektiver Homomorphismus: Die Homomorphie-Eigenschaften sind einfach nachzuweisen; z. B. sieht man sofort $S(0) = \emptyset$ und $S(1) = M$. Die Injektivität erfordert für den allgemeinen Fall etwas Mathematik; für die Tarski-Lindenbaum-Algebren folgt sie dagegen sofort aus der Definition der logischen Äquivalenz, da sich zwei nicht äquivalente Formeln in mindestens einer Belegung unterscheiden müssen. □

Die endlichen Tarksi-Lindenbaum-Algebren $\mathcal{F}_n$ sind sogar isomorph zu Potenzmengenalgebren, nämlich zur Potenzmenge der partiellen Belegungen der vorkommenden Aussagenvariablen. (Einer Formel entspricht die Menge der Belegungen, die sie wahr macht.)

[7] Der Index an den Operationen zeigt jeweils an, in welcher Struktur sie berechnet wird.

Aus dem Stone'schen Satz ergibt sich somit die Möglichkeit, die Gültigkeit logischer Gesetze durch Mengen-Diagramme (z. B. Venn-Diagramme) nachzuweisen.

Dualität

Definition 1.4.3 *Zu einer Boole'schen Algebra* $\mathfrak{B} = (B; \sqcap, \sqcup, ^{\mathrm{c}}, 1, 0)$ *ist die duale Algebra definiert als* $\mathfrak{B}^* = (B; \sqcup, \sqcap, ^{\mathrm{c}}, 0, 1)$. *In der Betrachtungsweise als partielle Ordnung ist* $\mathfrak{B}^* = (B; \sqsupseteq)$ *die umgedrehte Ordnung zu* $\mathfrak{B} = (B; \sqsubseteq)$, *also* $x \sqsubseteq_{\mathfrak{B}^*} y \iff y \sqsubseteq_{\mathfrak{B}} x$.

Satz 1.4.4 $\mathfrak{B}$ *und* $\mathfrak{B}^*$ *sind isomorph zueinander via* $b \mapsto b^{\mathrm{c}}$.

Beweis: Aus der Doppelnegationsregel $b^{\mathrm{cc}} = b$ folgt, dass $b \mapsto b^{\mathrm{c}}$ eine Bijektion ist. Zusammen mit den *de Morgan*'schen Regeln, $0^{\mathrm{c}} = 1$ und $1^{\mathrm{c}} = 0$ sieht man die Homomorphie. □

In der Sichtweise partieller Ordnungen ist dieser Satz nichts anderes als die Kontrapositionsregel. Die Axiome der Boole'schen Algebren bestehen bis auf die Doppelnegationsregel aus Paaren von zueinander dualen Axiomen.

Definition 1.4.5 *Sei* F *eine Formel, in der die Junktoren* $\to$ *und* $\leftrightarrow$ *nicht vorkommen. Die duale Formel* F^* *entsteht aus* F, *indem simultan alle Vorkommen von* $\wedge$ *durch* $\vee$ *ersetzt werden und umgekehrt und alle Vorkommen von* $\top$ *durch* $\bot$ *ersetzt werden und umgekehrt.*

Beispiel: Die duale Formel zu $(\neg(A_0 \wedge \bot) \vee (\neg A_2 \vee A_0))$ ist $(\neg(A_0 \vee \top) \wedge (\neg A_2 \wedge A_0))$.

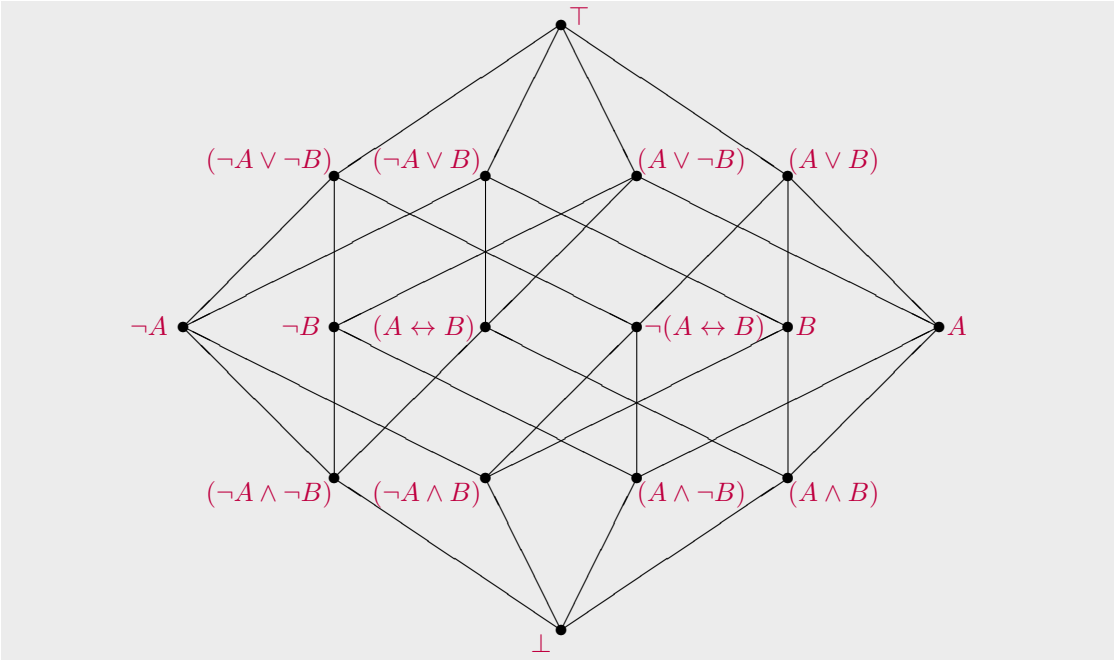
Folgerung 1.4.6 $F \vdash G \iff G^* \vdash F^*$ *und* $F \sim G \iff F^* \sim G^*$.

Beweis: Falls $F \vdash G$, so folgt mit der Kontrapositionsregel $\neg G \vdash \neg F$. Kommen $\to$ und $\leftrightarrow$ nicht vor, kann man mittels der *de Morgan*'schen Regeln die Negation nach innen ziehen; dabei wird F fast zu F^*, nur dass zusätzlich vor jeder Aussagenvariable ein Negationsjunktor steht. Also hat man (wenn $A_1, \ldots, A_n$ die vorkommenden Aussagenvariablen sind).

$$G^* \frac{\neg A_0}{A_0} \ldots \frac{\neg A_n}{A_n} \vdash F^* \frac{\neg A_0}{A_0} \ldots \frac{\neg A_n}{A_n},$$

wobei diese Verallgemeinerung der Schreibweise aus Lemma 1.3.1 für eine *simultane* uniforme Substitution der Aussagenvariablen durch ihre Negationen steht). Durch uniforme (Re-)Substitution (A_i wird erneut durch $\neg A_i$ substitutiert und die Doppelnegationsregel angewandt) erhält man schließlich $G^* \vdash F^*$. Die Aussage über äquivalente Formeln $F \sim G$ folgt daraus. □

Zur Illustration sei das Hasse-Diagramm der Tarski-Lindenbaum-Algebra $\mathcal{F}_2$ angegeben (mit ausgewählten Repräsentanten und der besseren Lesbarkeit willen mit A und B anstelle von A_0 und A_1). Die Isomorphie zwischen der Algebra und ihrer dualen Algebra sieht man hier als Punktsymmetrie am Mittelpunkt.



1.5 Erfüllbarkeit und Resolution

Typische Fragen zu aussagenlogischen Formeln lassen sich als Erfüllbarkeitsprobleme formulieren: Zum Beispiel sind F und G genau dann äquivalent, wenn $\neg(F \leftrightarrow G)$ nicht erfüllbar ist. Es gibt zwei Versionen des Erfüllbarkeitsproblems:

- Das *Entscheidungsproblem*: Ist eine gegebene Formel erfüllbar?
- Das *Such- oder Konstruktionsproblem*: Finde für eine gegebene (erfüllbare) Formel eine erfüllende Belegung.

Beide Probleme sind lösbar durch das Aufstellen von Wahrheitstafeln. Allerdings muss man bei n Aussagenvariablen im schlimmsten Fall 2^n Belegungen ausrechnen, hat also ein exponentielles Wachstum, was bereits bei etwa 100 Aussagenvariablen praktisch nicht mehr durchführbar ist. Es stellt sich daher die Frage, ob es bessere Algorithmen gibt als Wahrheitstafeln auszurechnen. Der Satz von Cook (siehe Satz 3.2.3) besagt, dass dies in einer prinzipiellen Weise nicht möglich ist. Allerdings gibt es gute Algorithmen für Formeln in besonderer Gestalt (z. B. Formeln in DNF, Horn-Formeln), und es gibt Algorithmen, die häufig besser als Wahrheitstafeln sind (Baum- oder Tableauverfahren, Resolution).

Baumkalkül / Tableau-Methode

In Wahrheitstafeln wird der Wahrheitswertverlauf einer Formel „von unten“ ausgerechnet, d. h. entlang des induktiven Aufbaus der Formel von den Aussagenvariablen ausgehend zu immer komplexeren Teilformeln. Alternativ kann man ihn „von oben“ bestimmen, d. h. unter Betrachtung des zuletzt hinzugekommenen Junktors untersuchen, unter welchen Bedingungen die Gesamtformel einen bestimmten Wahrheitswert bekommen kann. Es gibt verschiedene Varianten in der Darstellung dieser Methode, die als *Baumkalküle* oder *Tableau-Methoden* bekannt sind.

Am Beispiel der Konjunktion sei die Grundidee aufgezeigt: $(F \wedge G)$ ist genau dann *wahr*, wenn F und G *wahr* sind. Der Fall, dass $(F \wedge G)$ *wahr* ist, also den Wahrheitswert 1 bekommt – kurz

$(F \wedge G)$: 1 geschrieben –, löst sich daher in F: 1 und G: 1 auf. Der Fall, dass $(F \wedge G)$ *falsch* ist, entspricht *drei* Belegungen, die aber durch folgende *zwei* nicht-ausschließende Fälle abgedeckt sind: F: 0 oder G: 0. Schematisch stellt man dies dar durch:

$$\begin{array}{l} (F \wedge G) : 1 \\ \quad F : 1 \\ \quad G : 1 \end{array} \qquad\qquad \begin{array}{ccc} & (F \wedge G) : 0 & \\ \swarrow & & \searrow \\ F : 0 & & G : 0 \end{array}$$

Man kann sich nun leicht davon überzeugen, dass man jeden Junktor analog auflösen kann in entweder einen oder zwei Fälle:

$$\begin{array}{lllll} \neg F : 1 & (F \wedge G) : 1 & (F \vee G) : 1 & (F \to G) : 1 & (F \leftrightarrow G) : 1 \\ \quad F : 0 & \quad F : 1 & \swarrow \quad \searrow & \swarrow \quad \searrow & \swarrow \quad \searrow \\ & \quad G : 1 & F : 1 \quad G : 1 & F : 0 \quad G : 1 & F : 1 \quad F : 0 \\ & & & & G : 1 \quad G : 0 \\ \neg F : 0 & (F \wedge G) : 0 & (F \vee G) : 0 & (F \to G) : 0 & (F \leftrightarrow G) : 0 \\ \quad F : 1 & \swarrow \quad \searrow & \quad F : 0 & \quad F : 1 & \swarrow \quad \searrow \\ & F : 0 \quad G : 0 & \quad G : 0 & \quad G : 0 & F : 1 \quad F : 0 \\ & & & & G : 0 \quad G : 1 \end{array}$$

Indem man alle Junktoren der Formel auf diese Weise sukzessive auflöst, bekommt man einen binären Baum.[8] Man sagt, dass ein Pfad in diesem Baum *schließt*, wenn er widersprüchliche Information enthält: entweder $A_i : 1$ und $A_i : 0$ für eine Aussagenvariable A_i, oder $\top : 0$ oder $\bot : 1$. Die nicht-schließenden Pfade liefern dann genau die Belegungen, welche die Ausgangsbedingung $F : 1$ oder $F : 0$ erfüllen. Beim Entscheidungsproblem der Erfüllbarkeit startet man also mit $F : 1$ und kann abbrechen, sobald man einen nicht-schließenden Pfad gefunden hat.

Beispiel: Ist die Formel $(\neg(\neg A_0 \to (A_2 \vee A_1)) \leftrightarrow ((A_0 \wedge A_1) \to \neg\neg A_2))$ erfüllbar?

$$\begin{array}{ll} & (\neg(\neg A_0 \to (A_2 \vee A_1)) \overset{①}{\leftrightarrow} ((A_0 \wedge A_1) \to \neg\neg A_2)) : 1 \\ ① & \swarrow \qquad\qquad \searrow \\ & \overset{②}{\neg}(\neg A_0 \to (A_2 \vee A_1)) : 0 \qquad \neg(\neg A_0 \to (A_2 \vee A_1)) : 1 \\ & ((A_0 \wedge A_1) \overset{③}{\to} \neg\neg A_2) : 0 \qquad ((A_0 \wedge A_1) \to \neg\neg A_2) : 1 \\ ② & (\neg A_0 \overset{⑦}{\to} (A_2 \vee A_1)) : 1 \qquad \vdots \\ ③ & (A_0 \overset{⑥}{\wedge} A_1) : 1 \qquad ⑩ \text{ etc.} \\ & \overset{④}{\neg\neg} A_2 : 0 \qquad \vdots \\ ④ & \overset{⑤}{\neg} A_2 : 1 \qquad \textit{Wenn man auch diesen Teil} \\ ⑤ & A_2 : 0 \qquad \textit{des Baumes entwickelt,} \\ ⑥ & A_0 : 1 \qquad \textit{bekommt man sämtliche} \\ & A_1 : 1 \qquad \textit{erfüllenden Belegungen.} \\ ⑦ & \swarrow \qquad \searrow \\ & \overset{⑧}{\neg} A_0 : 0 \qquad (A_2 \overset{⑨}{\vee} A_1) : 1 \\ ⑧ & A_0 : 1 \qquad ⑨ \quad \swarrow \qquad \searrow \\ & \textit{erfüllbar} \qquad A_2 : 1 \qquad\qquad A_1 : 1 \\ & \qquad\quad \textit{schließt} \qquad\quad \textit{erfüllbar} \end{array}$$

Die beiden nicht-schließenden Pfade sind durch die Belegung $A_0 \mapsto 1, A_1 \mapsto 1, A_2 \mapsto 0$ erfüllt.

[8] Die Reihenfolge, in der man die Formeln in einem Pfad abarbeitet, ist beliebig. Geschickt ist es, wie im Beispiel zunächst die nicht-verzweigenden Fälle zu behandeln, um einen schmalen Baum zu bekommen.

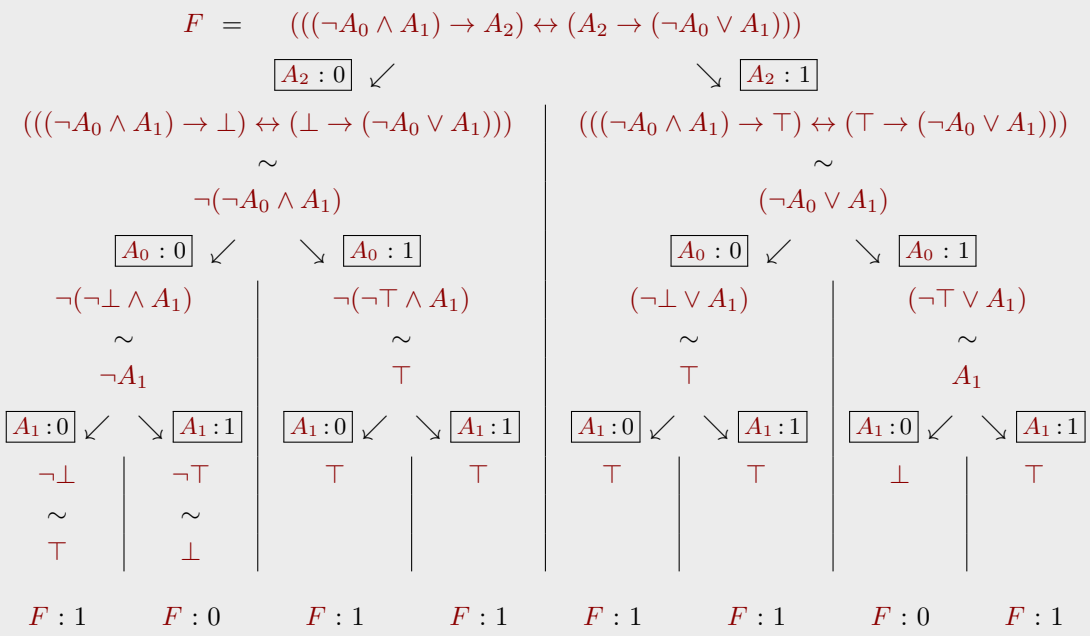
Die Methode von Quine

Die Methode von Quine ist eine weitere Möglichkeit, den Wahrheitsverlauf einer Formel zu bestimmen. Dazu legt man zunächst den Wahrheitswert einer (ggf. geeignet gewählten) Aussagenvariable A_i fest und ersetzt A_i in der Formel durch $\bot$ für den Fall „$A_i : 0$“ und durch $\top$ für den Fall „$A_i : 1$“. Die um die fehlenden Junktoren ergänzten $\top$-/$\bot$-Regeln lassen dann eine schnelle Vereinfachung der Formel zu:

$\neg\top \sim \bot$	$(A \wedge \top) \sim A$	$(A \vee \top) \sim \top$	$(A \to \top) \sim \top$	$(\top \to A) \sim A$	$(A \leftrightarrow \top) \sim A$
$\neg\bot \sim \top$	$(A \wedge \bot) \sim \bot$	$(A \vee \bot) \sim A$	$(A \to \bot) \sim \neg A$	$(\bot \to A) \sim \top$	$(A \leftrightarrow \bot) \sim \neg A$

In beiden Fällen fährt man dann mit der nächsten Aussagenvariable fort und erhält so einen binären Baum, dessen Zweige den Belegungen entsprechen und an deren Blättern die Formel $\top$ bzw. $\bot$ steht, falls die Formel F unter der entsprechenden Belegung wahr bzw. falsch wird.

Beispiel: Die Verkürzungen der Formeln nach den Einsetzungen von $\top$ bzw. $\bot$ sind hier nicht schrittweise ausgeführt, sondern nur das Endergebnis angegeben.



Formeln in DNF und KNF

Einfach sind die Erfüllbarkeitsprobleme für Formeln in disjunktiver Normalform: $F = \bigvee_i K_i$ ist genau dann erfüllbar, wenn $F = \top$ oder wenn ein Konjunktionsterm $K_i = (L_1 \wedge \cdots \wedge L_m)$ existiert, der nicht $\bot$ ist und zu keinem Literal $L_j = A_k$ auch das Literal $\neg A_k$ enthält. Eine erfüllende Belegung erhält man dann z. B. durch $\beta(A_i) = 0$, falls $\neg A_i$ eines der Literale in K_i ist, und $\beta(A_i) = 1$ sonst. Allerdings ist die Umwandlung einer gegebenen Formel in disjunktive Normalform im Allgemeinen zeitaufwendig, da die DNF exponentiell größere Länge als die Ausgangsformel haben kann. Dies gilt zwar auch für die Umwandlung einer Formel in konjunktive Normalform, allerdings gibt es in diesem Fall die Möglichkeit, durch zusätzliche Aussagenvariablen die Problematik zu umgehen:

Lemma 1.5.1 *Zu jeder Formel F gibt es eine Formel F^+ mit folgenden Eigenschaften:*

- *F und F^+ sind erfüllbarkeitsäquivalent, d. h. F ist genau dann erfüllbar, wenn F^+ es ist;*
- *F^+ ist in KNF mit maximal drei Literalen pro Klausel;*
- *es gibt eine (von F unabhängige) Konstante $C \in \mathbb{N}$ mit $\lg(F^+) \leqslant C \cdot \lg(F)$.*

Beweis: Für jede Teilformel F' wählt man eine Aussagenvariable $A_{F'}$, und zwar $A_{F'} = A_i$, wenn $F' = A_i$, und andernfalls eine jeweils „neue“ Aussagenvariable, also ein A_i, das in F nicht vorkommt und verschieden ist von allen $A_{F''}$ für $F' \neq F''$.

Angenommen $F' = *F_1 \dots F_n$ für einen n-stelligen Junktor $*$ mit $n \in \{0, 1, 2\}$ – hier ausnahmsweise der einheitlichen Behandlung halber in polnischer Notation geschrieben. Setze dann

$$\mathrm{E}_{F'} := (A_{F'} \leftrightarrow *A_{F_1} \dots A_{F_n}).$$

Für $F' = (G \wedge H)$ ist zum Beispiel $\mathrm{E}_{F'} = (A_{F'} \leftrightarrow (A_G \wedge A_H))$, für $F' = \neg G$ ist $\mathrm{E}_{F'} = (A_{F'} \leftrightarrow \neg A_G)$ und für $F' = \bot$ ist $\mathrm{E}_{F'} = (A_{F'} \leftrightarrow \bot)$.

Eine partielle Belegung β der in F vorkommenden Aussagenvariablen setzt sich nun durch $\beta'(A_{F'}) := \beta(F')$ auf eindeutige Weise zu einer Belegung der neuen Aussagenvariablen fort, die alle Äquivalenzen $\mathrm{E}_{F'}$ wahr macht. Insbesondere ist β genau dann eine F erfüllende Belegung, wenn $\beta'(A_F) = 1$. Also ist F genau dann erfüllbar, wenn es die Menge $\{A_F\} \cup \{\mathrm{E}_{F'} \mid F' \in \mathrm{Tf}(F)\}$ ist bzw. deren Konjunktion.

Jede Formel $\mathrm{E}_{F'}$ lässt sich nun in KNF schreiben, wobei maximal drei Literale pro Klausel vorkommen können, da jede der Formeln maximal drei Aussagenvariablen enthält. F^+ ist dann die Konjunktion von A_F und aller in KNF gebrachten Formeln $\mathrm{E}_{F'}$. Die Anzahl der Teilformeln von F ist beschränkt durch die Anzahl der Vorkommen von Junktoren, die wiederum durch die Länge $\lg(F)$ beschränkt ist. Die Länge aller KNF's von Formeln mit drei Aussagenvariablen ist ebenfalls beschränkt. Damit ergibt sich insgesamt, dass F^+ gegenüber F in der Länge maximal um einen konstanten Faktor zunimmt. (Genauere Analyse zeigt, dass man etwa $C = 20$ erreicht, mit Klammersparregeln etwa $C = 15$.) □

Beispiel: Die Formel $F = (A_0 \vee \neg(\neg A_1 \to A_0))$ mit Teilformeln $F' = \neg(\neg A_1 \to A_0)$ und $F'' = (\neg A_1 \to A_0)$ und $F''' = \neg A_1$ wird ersetzt durch die erfüllbarkeitsäquivalente Formel

$$\big(A_F \wedge (A_F \leftrightarrow (A_0 \vee A_{F'})) \wedge (A_{F'} \leftrightarrow \neg A_{F''}) \wedge (A_{F''} \leftrightarrow (A_{F'''} \to A_0)) \wedge (A_{F'''} \leftrightarrow \neg A_1)\big)$$

Beispielhaft sei angegeben, wie die Teilformel $\mathrm{E}_{F''}$ in KNF aussieht (als Illustration des zweiten Beweises von Satz 1.3.4 durch Umformungen):

$$\begin{aligned}\mathrm{E}_{F''} = (A_{F''} \leftrightarrow (A_{F'''} \to A_0)) &\sim ((A_{F''} \to (A_{F'''} \to A_0)) \wedge ((A_{F'''} \to A_0) \to A_{F''})) \\ &\sim ((\neg A_{F''} \vee \neg A_{F'''} \vee A_0) \wedge (\neg(\neg A_{F'''} \vee A_0) \vee A_{F''})) \\ &\sim ((\neg A_{F''} \vee \neg A_{F'''} \vee A_0) \wedge (A_{F''} \vee A_{F'''}) \wedge (A_{F''} \vee \neg A_0))\end{aligned}$$

Da es für die Formeln $\mathrm{E}_{F'}$ aus dem Beweis nur endlich viele Möglichkeiten gibt, braucht man diese Umformungen natürlich nicht jedesmal vorzunehmen, sondern kann den Übergang von der Form der Formel zur Form der KNF fest einprogrammieren.

Man sieht auch, dass man sich bei der Konstruktion der erfüllbarkeitsäquivalenten Formel in KNF systematisch zwei Arten von Schritte ersparen kann, nämlich die Einführung von eigenen Aussagenvariablen für die gesamte Formel und für negierte Aussagenvariablen. Im Beispiel kann man $(A_F \wedge (A_F \leftrightarrow (A_0 \vee A_{F'})))$ zu $(A_0 \vee A_{F'})$ verkürzen und die negierte Aussagenvariable

$\neg A_1$ anstelle von $A_{F'''}$ in den vorletzten Term der Konjunktion einsetzen. Dadurch erhält man eine ebenso leicht in KNF umformbare, erfüllbarkeitsäquivalente Formel

$$\big((A_0 \vee A_{F'}) \wedge (A_{F'} \leftrightarrow \neg A_{F''}) \wedge (A_{F''} \leftrightarrow (\neg A_1 \rightarrow A_0))\big).$$

Resolutionsmethode

Für Formeln in KNF gibt es die *Resolutionsmethode* zum Testen von Erfüllbarkeit.

Zur Erinnerung (Definition 1.3.3): Eine *Klausel* ist eine Disjunktion von Literalen. Aus Gründen notationeller Einfachheit soll im Folgenden eine Klausel $(L_1 \vee \cdots \vee L_k)$ mit der Menge der Literale $\{L_1, \ldots, L_k\}$ identifiziert werden.[9] Eine solche Klausel ist also durch eine Belegung β erfüllt, wenn β die Disjunktion der Literale wahrmacht. Da dies der bisherigen Definition von Erfüllbarkeit von Formelmengen widerspricht, schreibe ich zur Kenntlichmachung Klauseln als $\{L_1, \ldots, L_k\}$. Für eine Menge von Klauseln $\{C_1, \ldots, C_n\}$ hingegen soll Erfüllbarkeit durch β wie früher bedeuten, dass β jede der Klauseln erfüllt. Damit steht eine *Klauselmenge* hinsichtlich Erfüllbarkeit also für eine Formel in KNF.

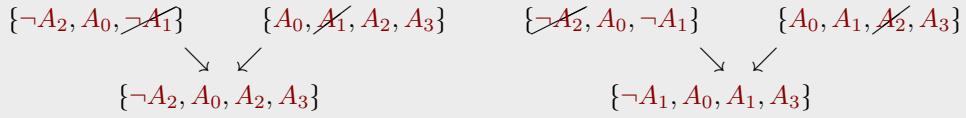
Beispiel: Die Formel $((\neg A_2 \vee \neg A_1 \vee A_0) \wedge (A_2 \vee A_1) \wedge (A_2 \vee \neg A_0))$ wird zur Klauselmenge

$$\big\{\{\neg A_2, \neg A_1, A_0\}, \{A_2, A_1\}, \{A_2, \neg A_0\}\big\}$$

Die *leere Klausel* $\{\}$ enthält kein Literal, entspricht also der „leeren Disjunktion“, d. h. der Formel $\bot$, und ist somit nicht erfüllbar. Eine Menge an Klauseln, die die leere Klausel enthält – insbesondere auch die Menge $\{\{\}\}$, die nur die leere Klausel enthält – ist also ebenfalls nicht erfüllbar. Die *leere Menge an Klauseln* $\{\}$ entspricht dagegen der „leeren Konjunktion“, d. h. der Formel $\top$, und ist erfüllbar.

Definition 1.5.2 *Sind $C_1 = \{L_1, \ldots, L_k, A_i\}$ und $C_2 = \{L'_1, \ldots, L'_l, \neg A_i\}$ Klauseln, dann heißt die Klausel $R = \{L_1, \ldots, L_k, L'_1, \ldots, L'_l\}$ eine Resolvente von C_1 und C_2. Man sagt, dass R durch Resolution aus C_1 und C_2 entsteht.*

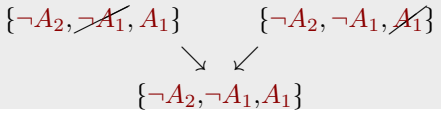
Beispiel: **(a)** Die Klauseln $C_1 = \{\neg A_2, A_0, \neg A_1\}$ und $C_2 = \{A_0, A_1, A_2, A_3\}$ haben zwei Resolventen: zum einen $\{\neg A_2, A_0, A_2, A_3\}$ und zum andern $\{\neg A_1, A_0, A_1, A_3\}$.



Dagegen ist $\{A_0, A_3\}$ keine Resolvente von C_1 und C_2, da immer nur eine Aussagenvariable gleichzeitig resolviert werden darf.

(b) Die Klauseln $C_3 = \{\neg A_2, A_1, A_0\}$ und $C_4 = \{A_3, \neg A_2\}$ haben keine Resolvente miteinander.

(c) Die Klausel $C_5 = \{\neg A_2, \neg A_1, A_1\}$ bildet mit sich selbst eine Resolvente, die aber nichts Neues ergibt, weswegen dieser Fall nicht betrachtet werden muss:



[9] Allerdings kann man daraus die Klausel nur bis auf logische Äquivalenz zurückgewinnen, da doppelte Vorkommen von Literalen, ihre Reihenfolge und Klammerungsreihenfolge verloren gehen.

Eine endliche Klauselmenge enthält nur endlich viele Aussagenvariablen, aus denen sich nur endlich viele verschiedene Klauseln ergeben können. Daher kann man eine endliche Klauselmenge in endlich vielen Schritten unter Resolution abschließen, bis man die kleinste Ober-Klauselmenge erhält, aus der sich keine neuen Resolventen mehr ergeben.

Satz 1.5.3 (Resolutionsmethode) *Eine endliche Klauselmenge*[10] *ist genau dann nicht erfüllbar, wenn sich durch sukzessive Resolution die leere Klausel ergibt.*

Beweis: „⇐“: Eine Belegung β erfüllt mit $C_1 = \{L_1, \ldots, L_k, A_i\}$ und $C_2 = \{L'_1, \ldots, L'_l, \neg A_i\}$ auch jede Resolvente $R = \{L_1, \ldots, L_k, L'_1, \ldots, L'_l\}$. Denn entweder $\beta(A_i) = 1$ und β muss eines der Literale $L'_1, \ldots, L'_l$ wahr machen, um C_2 zu erfüllen, oder $\beta(A_i) = 0$ und β muss eines der Literale $L_1, \ldots, L_k$ wahr machen, um C_1 zu erfüllen. In jedem Fall ist R erfüllt. Daraus folgt, dass eine erfüllbare Klauselmenge einen erfüllbaren Abschluss unter Resolution hat, der somit nicht die leere Klausel enthalten kann.

„⇒“: Angenommen $\mathcal{C}$ ist eine unter Resolution abgeschlossene Klauselmenge, die nicht die leere Klausel enthält. Ohne Einschränkung seien $A_0, \ldots, A_{n-1}$ die in $\mathcal{C}$ vorkommenden Aussagenvariablen. Per Induktion über n wird gezeigt, dass $\mathcal{C}$ erfüllbar ist.

Für $n = 0$ gibt es nur die leere Klauselmenge $\mathcal{C} = \{\,\}$, die nicht die leere Klausel enthält. $\mathcal{C}$ wird durch alle Belegungen erfüllt.

Im Induktionsschritt $n \to n + 1$ betrachtet man die Aussagenvariable A_n. Es können nicht beide Klauseln $\{A_n\}$ und $\{\neg A_n\}$ in $\mathcal{C}$ vorkommen, denn sonst bekäme man die leere Klausel als Resolvente. Angenommen also $\{\neg A_n\} \notin \mathcal{C}$ (der andere Fall ist analog). Dann modifiziert man $\mathcal{C}$ folgendermaßen zu einer Klauselmenge $\mathcal{C}'$ in den Aussagenvariablen $A_0, \ldots, A_{n-1}$:

- Wenn die Klausel $C \in \mathcal{C}$ das Literal A_n enthält, entfällt C.
- Wenn die Klausel $C \in \mathcal{C}$ das Literal $\neg A_n$ enthält, wird dieses aus C entfernt und $C' := C \setminus \{\neg A_n\}$ zu einer Klausel in $\mathcal{C}'$.
- Alle anderen Klauseln werden unverändert von $\mathcal{C}$ nach $\mathcal{C}'$ übernommen.

Man vergewissert sich unschwer, dass $\mathcal{C}'$ eine unter Resolution abgeschlossene Klauselmenge ist, die zudem nicht die leere Klausel enthält, da $\{\neg A_n\} \notin \mathcal{C}$. Per Induktion ist $\mathcal{C}'$ durch ein β' erfüllbar. Belegungen β mit $\beta(A_n) = 1$, die auf $A_0, \ldots, A_{n-1}$ mit β' übereinstimmen, erfüllen dann $\mathcal{C}$, denn durch $\beta(A_n) = 1$ werden alle weggelassenen Klauseln erfüllt; die Erfüllung der modifizierten Klauseln durch β' ändert sich nicht durch Hinzunahme von $\neg A_n$. □

Der Beweis zeigt auch, wie man aus dem Verfahren eine erfüllende Belegung gewinnen kann. Das Resolutionsverfahren wird in der Praxis benutzt und liefert häufig gute Ergebnisse; der Abschluss unter Resolution kann aber auch zu exponentiell vielen Resolventen führen.

Beispiel 1: Die Menge folgender Klauseln soll auf Erfüllbarkeit getestet werden:

$$C_1 = \{\neg A_0, A_1\},\ C_2 = \{A_1, A_2\},\ C_3 = \{A_0, \neg A_2\},\ C_4 = \{\neg A_0, \neg A_1, \neg A_2\},\ C_5 = \{\neg A_0, A_3\}$$

Im ersten Schritt bekommt man durch Resolution von C_1 und C_3 die neue Klausel $C_6 = \{A_1, \neg A_2\}$ und durch Resolution von C_1 und C_4 die neue Klausel $C_7 = \{\neg A_0, \neg A_2\}$. Die Klauseln

[10] Aus dem Kompaktheitssatz 1.5.4 folgt, dass der Satz ebenso für unendliche Klauselmengen gilt. Der Abschluss unter Resolution lässt sich dann aber i. Allg. nicht mehr in endlich vielen Schritten erreichen.

C_1 und C_2 bzw. C_5 lassen keine Resolution zu. Aus C_2 und C_3 erhält man $C_8 = \{A_0, A_1\}$; aus C_2 und C_4 die beiden Klauseln $C_9 = \{\neg A_0, A_2, \neg A_2\}$ und $C_{10} = \{\neg A_0, A_1, \neg A_1\}$. Aus C_3 und C_4 bekommt man die Klausel $C_{11} = \{\neg A_1, \neg A_2\}$ und aus C_3 und C_5 die Klausel $C_{12} = \{\neg A_2, A_3\}$; C_2 bzw. C_4 und C_5 lassen wieder keine Resolution zu.

Klauseln wie C_9 und C_{10}, die eine Aussagenvariable und ihr Negat erhalten, sind immer erfüllbar und können nie dazu beitragen, dass die leere Klausel entsteht. Man kann die daher in der Betrachtung weglassen.

Im zweiten Schritt ergeben sich neue Resolutionen: aus C_1 und C_8 oder auch aus C_2 und C_6 erhält man $C_{13} = \{A_1\}$. C_1 und C_{11} lassen ebenfalls eine Resolution zu, diese ergibt aber wieder C_7. An neuen, relevanten Klauseln ergibt sich noch $\{\neg A_0, A_1\}$, $\{A_1, A_3\}$ und $\{\neg A_2\}$.

Im dritten Schritt findet man als neue relevante Klausel $\{\neg A_0, \neg A_2, \neg A_3\}$ und im vierten Schritt $\{A_1, \neg A_2, \neg A_3\}$. Danach erhält man durch Resolution keine neuen Klauseln mehr, die nicht eine Aussagenvariable und ihr Negat erhalten. Im Wesentlichen hat man damit also den Abschluss der Klauselmenge unter Resolution gefunden (bis auf für Erfüllbarkeit irrelevante Klauseln). Da die leere Klausel nicht dabei ist, ist die Ausgangsklauselmenge erfüllbar.

Man kann auch alle erfüllenden Belegungen gewinnen: Aus der Einerklausel $\{A_1\}$ sieht man, dass A_1 wahr sein muss. Löscht man alle Klauseln mit einem positiven Vorkommen von A_1 und löscht alle Vorkommen von $\neg A_1$ aus den verbleibenden Klauseln, erhält man:

$$\{A_0, \neg A_2\}\,,\ \{\neg A_0, \neg A_2\}\,,\ \{\neg A_0, A_3\}\,,\ \{\neg A_2\}\,,\ \{\neg A_0, \neg A_2, \neg A_3\}$$

Man sieht erneut an der Einerklausel $\{\neg A_2\}$, dass A_2 falsch ein muss. Löscht man alle Klauseln mit einem Vorkommen von $\neg A_2$ und löscht alle Vorkommen von A_2 aus den verbleibenden Klauseln, erhält man:

$$\{\neg A_0, A_3\}$$

A_0 kann nun sowohl wahr als auch falsch sein. Setzt man A_0 wahr, bleibt nach dem Reduktionsverfahren die Klausel $\{A_3\}$ und A_3 muss ebenfalls wahr sein. Setzt man A_0 falsch, bleibt nach dem Reduktionsverfahren keine Klausel übrig und A_3 kann einen beliebigen Wahrheitswert annehmen.

Beispiel 2: Aus der Menge der Klauseln

$$\{\neg A_0, A_1\}\,,\ \{\neg A_0, A_2\}\,,\ \{A_0, A_3\}\,,\ \{A_0, A_1\}\,,\ \{\neg A_1, \neg A_2\}\,,\ \{\neg A_1, \neg A_3\}$$

bekommt man durch Resolution aus der 1. und 4. Klausel $\{A_1\}$, aus der 3. und 6. $\{A_0, \neg A_1\}$ und aus der 2. und 5. $\{\neg A_0, \neg A_1\}$. Diese beiden letzten lassen sich zu $\{\neg A_1\}$ resolvieren; mit $\{A_1\}$ ergibt sich dann die leere Klausel. Also ist die Klauselmenge nicht erfüllbar.

Es gibt viele Varianten und Verfeinerungen der Resolutionsmethode. Man kann zum Beispiel zunächst unter allen Resolution mit einelementigen Klauseln abschließen (was von der Komplexität her überschaubar bleibt) und dann die Größe steigern.

Einen Spezialfall stellen Horn-Formeln da: Eine Horn-Formel ist eine Formel in KNF in der jede Klausel eine Horn-Klausel ist, d. h. maximal ein positives Literal enthält. Für Horn-Formeln gibt es lineare Algorithmen zum Testen von Erfüllbarkeit und zur Konstruktion von erfüllenden Belegungen, z. B. Varianten der Resolutionsmethode oder den *Markierungsalgorithmus*.

Beispiel für Umformung in KNF und Resolution

Mit der in den vorangehenden beiden Abschnitten präsentierten Methode soll die Gültigkeit des *modus tollens* $(A_0 \to A_1), \neg A_1 \vdash \neg A_0$ überprüft werden, d. h. es soll die Unerfüllbarkeit gezeigt werden von der Formel

$$\neg(((A_0 \to A_1) \wedge \neg A_1) \to \neg A_0).$$

Schnelle Umformung in KNF ergibt mit dem Zwischenschritt der Einführung zusätzlicher Aussagenvariablen für Teilformeln zunächst

$$(\neg A_4 \wedge (A_4 \leftrightarrow (A_3 \to \neg A_0)) \wedge (A_3 \leftrightarrow (A_2 \wedge \neg A_1)) \wedge (A_2 \leftrightarrow (A_0 \to A_1)))$$

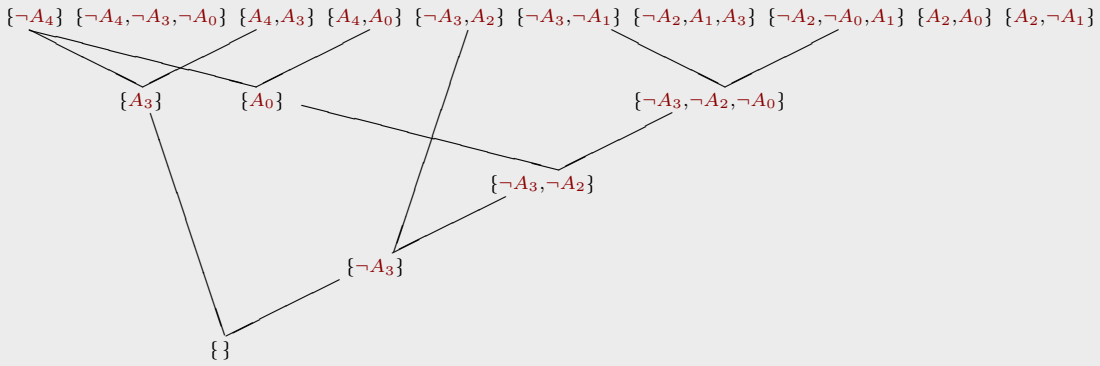
und schließlich

$$\begin{aligned}\big(\neg A_4 \quad &\wedge \quad (\neg A_4 \vee \neg A_3 \vee \neg A_0) \wedge (A_4 \vee A_3) \wedge (A_4 \vee A_0) \\ &\wedge \quad (\neg A_3 \vee A_2) \wedge (\neg A_3 \vee \neg A_1) \wedge (\neg A_2 \vee A_1 \vee A_3) \\ &\wedge \quad (\neg A_2 \vee \neg A_0 \vee A_1) \wedge (A_2 \vee A_0) \wedge (A_2 \vee \neg A_1)\big),\end{aligned}$$

bzw. als Menge von Klauseln

$$\big\{\{\neg A_4\}, \{\neg A_4, \neg A_3, \neg A_0\}, \{A_4, A_3\}, \{A_4, A_0\}, \{\neg A_3, A_2\}, \\ \{\neg A_3, \neg A_1\}, \{\neg A_2, A_1, A_3\}, \{\neg A_2, \neg A_0, A_1\}, \{A_2, A_0\}, \{A_2, \neg A_1\}\big\}.$$

Mit systematischer Resolution bekommt man daraus die leere Klausel. Hier ist nur ein Auszug zielführender Resolutionen gezeigt:



Duale Resolution

(Inhalt fehlt noch; ist auf Seite 52 angedeutet.)

Der Kompaktheitssatz

Satz 1.5.4 (Kompaktheits- oder Endlichkeitssatz)
Sei T eine unendliche Formelmenge.[11] *T ist genau dann erfüllbar, wenn T endlich erfüllbar ist, d. h. wenn jede endliche Teilmenge von T erfüllbar ist.*

[11] In der Definition der aussagenlogischen Sprache wurden nur abzählbar unendlich viele Aussagenvariablen zugelassen; daher können Formelmengen bestenfalls abzählbar unendlich werden. Man kann die Aussagenlogik problemlos mit einer größeren Anzahl an Aussagenvariablen gestalten. Der Kompaktheitssatz gilt dann auch für überabzählbare Mengen – im Wesentlichen mit dem gleichen Beweis: Man braucht dazu allerdings ein wenig Handwerkszeug zum tranfiniten Aufzählen überabzählbarer Mengen.

BEWEIS: Es ist offensichtlich, dass aus Erfüllbarkeit endliche Erfüllbarkeit folgt. Sei also T endlich erfüllbar. Per Induktion konstruieren wir endlich erfüllbare Formelmengen $T_0 \subseteq \cdots \subseteq T_n \subseteq T_{n+1} \subseteq \ldots$, die sukzessive Literale zu allen Aussagenvariablen enthalten:

$$T_0 := T \quad \text{und} \quad T_{n+1} := \begin{cases} T_n \cup \{A_n\} & \text{falls endlich erfüllbar,} \\ T_n \cup \{\neg A_n\} & \text{sonst.} \end{cases}$$

Behauptung: T_{n+1} ist auch im zweiten Fall endlich erfüllbar. Angenommen sowohl $T_n \cup \{A_n\}$ als auch $T_n \cup \{\neg A_n\}$ wären nicht endlich erfüllbar. Dann gibt es endliche Teilmengen S^+ und S^- von T, so dass $S^+ \cup \{A_n\}$ und $S^- \cup \{\neg A_n\}$ nicht erfüllbar sind. Mit $S := S^+ \cup S^-$ sind dann auch $S \cup \{A_n\}$ und $S \cup \{\neg A_n\}$ nicht erfüllbar. Da T_n aber endlich erfüllbar ist, gibt es eine Belegung β, welche S erfüllt. Je nachdem, ob $\beta(A_i) = 1$ oder $= 0$ gilt, erfüllt β aber die erste oder die zweite Menge: Widerspruch.

Es folgt, dass auch $T_\infty := \bigcup_{n\in\mathbb{N}} T_n$ endlich erfüllbar ist, da jede endliche Teilmenge von T_∞ bereits in einem T_n liegt.

Definiere nun $\beta(A_i) := \begin{cases} 1, & \text{falls } A_i \in T_\infty, \quad \text{und setze dann } L_i := A_i, \\ 0, & \text{falls } \neg A_i \in T_\infty, \quad \text{und setze dann } L_i := \neg A_i. \end{cases}$

Behauptung: β erfüllt T_∞. Denn sei $F \in T_\infty$ und m so, dass $F = F(A_0, \ldots, A_m)$. Dann ist $\{F, L_0, \ldots, L_m\}$ als endliche Teilmenge von T_∞ erfüllbar. Jede erfüllende Belegung β' ist aber auf $A_0,\ldots,A_m$ festgelegt (da nämlich $\beta'(A_i) = 1 \iff A_i = L_i$) und stimmt darauf mit β überein. Insbesondere erfüllt β also F. □

1.6 Semantik der intuitionistischen Aussagenlogik

Die intuitionistische Aussagenlogik ist eine Logik, die die gleiche Syntax wie die klassische zweiwertige Aussagenlogik benutzt, aber eine andere Semantik. Entwickelt wurde der Intuitionismus von L. E. J. Brouwer zu Beginn des 20. Jahrhunderts als eine mathematik-philosophische Position im sogenannten Grundlagenstreit der Mathematik. Die intuitionistische Logik wurde von seinem Schüler Arend Heyting formalisiert.

Grob gesprochen werden in der Semantik die Wahrheitswerte *wahr* und *falsch* durch explizite Beweisbarkeit bzw. Widerlegbarkeit ersetzt. Dadurch entfällt das Prinzip des ausgeschlossenen Dritten, da es Aussagen geben kann, die weder beweisbar noch widerlegbar sind. Der Intuitionismus stellt eine – je nach Sichtweise – schwächere oder stärkere Logik als die klassische Logik dar: Alle intuitionistischen Äquivalenzen sind auch klassisch gültig, aber nicht umgekehrt. Es gelten also *weniger* Gesetze, dadurch hat die intuitionistische Logik aber eine *größere* Differenzierungskraft.

Für die Informatik ist die intuitionistische Logik insofern von Bedeutung, als sie in einem gewissen Sinn das Berechenbarkeitsverhalten beschreibt: Beweisbarkeit in inutitionistischer Logik entspricht Berechenbarkeit, Widerlegbarkeit dem Nachweis, dass etwas nicht berechenbar ist. Präzise gemacht wird dies durch die *Curry-Howard-Korrespondenz* (auch: Curry-Howard-Isomorphismus), die eine Übersetzung zwischen Beweisen in intuitionistischer Logik und Programmen angibt.

Ich beschränkt mich in meiner kurzen Darstellung der intuitionistischen Logik hier auf das Junktorensystem $\{\bot, \to, \wedge, \vee\}$, da auch intuitionistisch die folgenden Äquivalenzen gelten, die

daher als Definitionen für die fehlenden Junktoren $\leftrightarrow$, $\neg$ und $\top$ genommen werden können:

$$\begin{aligned}(A_0 \leftrightarrow A_1) &\sim ((A_0 \to A_1) \wedge (A_1 \to A_0))\\ \neg A_0 &\sim (A_0 \to \bot)\\ \top &\sim \neg\bot \sim (\bot \to \bot)\end{aligned}$$

Von diesen vier Junktoren kann allerdings kein weiterer mehr weggelassen werden. Die Regeln zur Formelbildung sind ansonsten die gleichen wie in der klassischen Aussagenlogik.

Definition 1.6.1 *Ein Modell für die intuitionistische Logik (in diesem Abschnitt auch kurz: Modell) ist eine partiell geordnete Menge* $(W, \leqslant)$.[12]

Die Elemente von W nennt man *Zustände* (oder, aus der Modallogik kommend: *mögliche Welten*), die Relation $\leqslant$ heißt oft *Zugangsrelation*. Für $w_1 \leqslant w_2$ sagt man: „w_2 *kommt nach* w_1“ oder „w_2 *ist später als* w_1“ (bzw. „w_1 *sieht* w_2“).

Definition 1.6.2 *Eine Belegung in einem Modell für die intuitionistische Logik ist eine Abbildung* $\beta : W \times \{A_i \mid i \in \mathbb{N}\} \to \{0,1\}$*, die monoton bezüglich* $\leqslant$ *ist, d. h. falls* $w_1 \leqslant w_2$*, so muss* $\beta(w_1, A_i) \leqslant \beta(w_2, A_i)$ *gelten.*

Konkret heißt dies, dass ein Wert 1 in allen späteren Zuständen erhalten bleibt, während sich ein Wert 0 in einem späteren Zustand zu 1 ändern kann. Vorstellen kann man sich ein $w \in W$ mit $\beta(w, A_i) = 1$ als einen Zeitpunkt, zu dem man einen Beweis (bzw. eine Berechnung) für A_i hat – der in späteren Zuständen nicht verloren geht – während $\beta(w, A_i) = 0$ bedeutet, dass man (noch) keinen Beweis (bzw. keine Berechung) für A_i hat. Im Gegensatz zur klassischen Logik bedeutet die Zuweisung des Wertes 0 zu A_i hier also nicht, dass A_i nicht gilt oder falsch ist, sondern nur, dass nicht bekannt ist, ob A_i gilt.

Wie in der klassischen Logik wird die Belegung nun zu einer Funktion fortgesetzt, die in jedem Zustand $w \in W$ allen aussagenlogischen Formeln F einen Wert $\beta(w, F) \in \{0,1\}$ zuordnet.

$$\begin{aligned}\beta(w, \bot) &:= 0\\ \beta(w, (F \wedge G)) &:= \min\{\beta(w, F), \beta(w, G)\}\\ \beta(w, (F \vee G)) &:= \max\{\beta(w, F), \beta(w, G)\}\\ \beta(w, (F \to G)) &:= \begin{cases} 1 & \beta(w', F) \leqslant \beta(w', G) \text{ für alle } w' \in W \text{ mit } w \leqslant w'\\ 0 & \text{sonst}\end{cases}\end{aligned}$$

Lemma 1.6.3 *Die Fortsetzungen von Belegungen sind für alle Formeln monoton, d. h. für alle Formeln* F *folgt aus* $w_1 \leqslant w_2$ *auch* $\beta(w_1, F) \leqslant \beta(w_2, F)$.

Beweis: Per Induktion über den Aufbau der Formeln:

Monotonie gilt per Definition der Belegung für Aussagenvariablen und offensichtlich für $\bot$. Man weiß oder überlegt sich leicht, dass Maximum und Minimum monotoner Funktionen wieder monoton sind, also bleibt die Eigenschaft bei Konjunktionen und Disjunktionen erhalten.

[12] Dies bedeutet, dass $\leqslant$ eine reflexive, antisymmetrische und transitive binäre Relation ist.

Schließlich ist die Belegung für die Implikation so definiert, dass sie monoton wird: Wenn $w \leqslant w''$ und $\beta(w, (F \to G)) = 1$, dann ist per Definition $\beta(w', F) \leqslant \beta(w', G)$ für alle w' mit $w \leqslant w'$, insbesondere also auch für alle w' mit $w'' \leqslant w'$, da $\leqslant$ transitiv ist. □

Definition 1.6.4 **(a)** *Eine aussagenlogische Formel F heißt intuitionistische Tautologie, falls $\beta(w, F) = 1$ für alle Modelle $(W, \leqslant)$, alle Belegungen β und alle $w \in W$.*

(b) *Zwei aussagenlogische Formeln F_1, F_2 heißen intuitionistisch äquivalent zueinander, falls $\beta(w, F_1) = \beta(w, F_2)$ für alle Modelle $(W, \leqslant)$, alle Belegungen β und alle $w \in W$.*

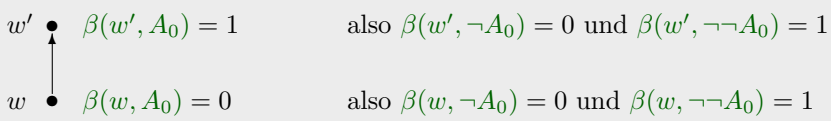
(c) *Eine aussagenlogische Formel F folgt intuitionistisch aus einer Menge $\{F_i \mid i \in I\}$ von Formeln, falls für jedes Modell $(W, \leqslant)$, jede Belegung β und jedes $w \in W$, in dem für alle $i \in I$ die Bedingung $\beta(w, F_i) = 1$ erfüllt ist, auch $\beta(w, F) = 1$ gilt.*

Man schreibt als Zeichen $\vdash_{\text{int}}$ bzw. $\sim_{\text{int}}$ und sieht leicht aus der Definition, dass entsprechenden Äquivalenzen wie in der klasischen Logik gelten, also beispielsweise:

$$\begin{aligned} \vdash_{\text{int}} F &\iff F \sim_{\text{int}} \top \\ F_1 \sim_{\text{int}} F_2 &\iff \vdash_{\text{int}} (F_1 \leftrightarrow F_2) \\ F_1 \vdash_{\text{int}} F_2 &\iff \vdash_{\text{int}} (F_1 \to F_2) \end{aligned}$$

Endliche oder diskrete Modelle $(W, \leqslant)$ zeichnet man gerne als gerichtete Graphen, wobei die Kanten in Richtung der Zugangsrelation weisen, also zu den späteren Zuständen hin, und nur Kanten zu den unmittelbaren echten Nachfolgern hin gezeichnet werden.

Das einfachste nicht-triviale Modell zeigt bereits zwei wesentliche Unterschiede zwischen der intuitionistischen und der klassischen Aussagenlogik:



Man sieht zum einen, dass wegen $\beta(w, (A_0 \vee \neg A_0)) = 0$ das Prinzip des ausgeschlossenen Dritten verletzt ist. Zum andern sieht man ebenfalls im Zustand w, dass A_0 und $\neg\neg A_0$ nicht äquivalent sind, also die Doppelnegationsregel nicht gilt. Es gilt lediglich $A_0 \vdash_{\text{int}} \neg\neg A_0$. Auch die *de Morgan*'schen Regeln gelten nur teilweise.

Man kann sich aber überlegen, dass die *Tripelnegationsregel* gilt, also $\neg A_0 \sim_{\text{int}} \neg\neg\neg A_0$.[13]

Mengendarstellung und Heyting-Algebren

Eine Teilmenge von $(W, \leqslant)$ heiße $\leqslant$-*abgeschlossen*, falls sie mit jedem Element auch alle größeren enthält. Lemma 1.6.3 zeigt, dass in einem Modell $(W, \leqslant)$ mit Belegung β jede Formel F eine $\leqslant$-abgeschlossene Menge $[\![F]\!] := \{w \in W \mid \beta(w, F) = 1\}$ bestimmt. Es gilt dann

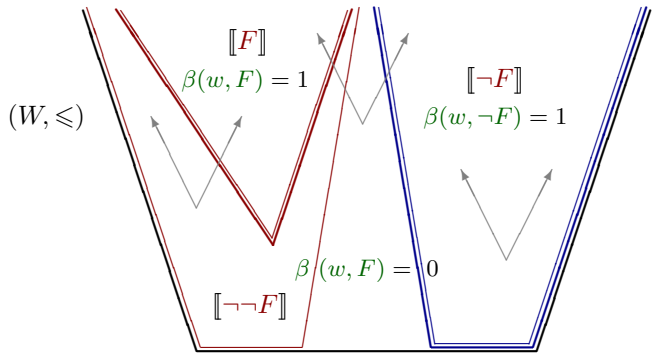
$$[\![\bot]\!] = \emptyset$$
$$[\![(F_1 \wedge F_2)]\!] = [\![F_1]\!] \cap [\![F_2]\!]$$

[13] Beweisidee: Aus $A_0 \vdash_{\text{int}} \neg\neg A_0$ bekommt man zum einen durch Substitution $\neg A_0 \vdash_{\text{int}} \neg\neg\neg A_0$, zum andern durch Kontraposition $\neg\neg\neg A_0 \vdash_{\text{int}} \neg A_0$. Kontraposition funktioniert intuitionistisch aber nur in eine Richtung: $(A_0 \to A_1) \vdash_{\text{int}} (\neg A_1 \to \neg A_0)$.

$[\![(F_1 \vee F_2)]\!] = [\![F_1]\!] \cup [\![F_2]\!]$

$[\![(F_1 \to F_2)]\!]$ ist die maximale $\leqslant$-abgeschlossene Teilmenge von $\big(W \setminus [\![F_1]\!]\big) \cup [\![F_2]\!]$

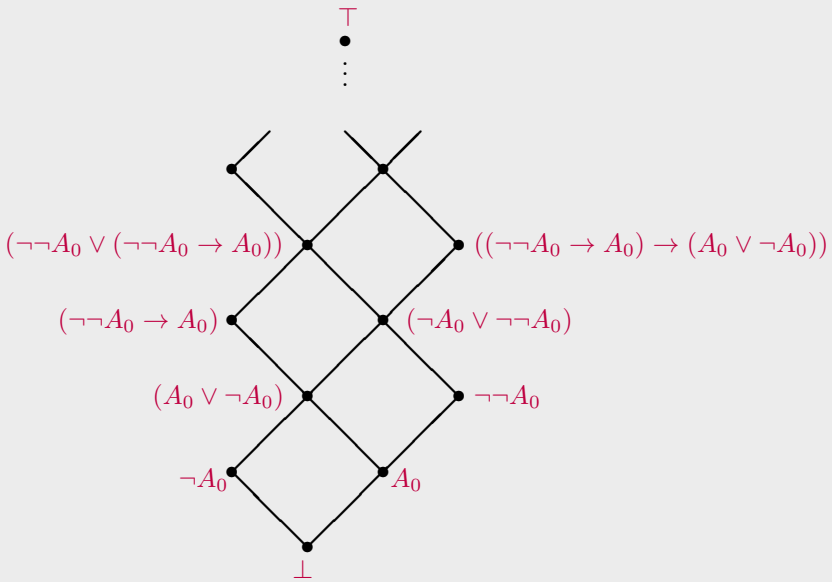
Damit ergibt sich, dass $[\![\neg F]\!]$ die größte $\leqslant$-abgeschlossene Teilmenge im Komplement von $[\![F]\!]$ ist, im folgenden Bild andeutungsweise illustriert (spätere Zustände stehen weiter oben, zum Teil durch Pfeile angedeutet):



Mit dieser Überlegung kann man sich z. B. die Gültigkeit der Tripelnegationsregel (und andere Regeln) plausibel machen: $[\![\neg F]\!]$ und $[\![\neg\neg F]\!]$ sind bereits maximal komplementäre Teilmengen unter den $\leqslant$-abgeschlossenen Mengen, also kann $[\![\neg\neg\neg F]\!]$ nicht noch größer als $[\![\neg F]\!]$ werden.

Man kann nun Modelle $(W, \leqslant)$ mit Belegungen β konstruieren, für die die Abbildung $F \mapsto [\![F]\!]$ injektiv ist. Dadurch bekommt man eine Darstellung von Formeln durch Mengen analog zum Satz von Stone. Die Tarski-Lindenbaum-Algebra für die intuitionistische Aussagenlogik ist allerdings keine Boole'sche Algebra mehr, sondern eine sogenannte Heyting-Algebra: ein distributiver beschränkter Verband mit Pseudokomplementen anstelle von Komplementen. [14]

Im Gegensatz zur klassischen Logik gibt es bereits mit einer Aussagenvariablen unendlich viele Formeln, die intuitionistisch paarweise nicht äquivalent zueinander sind. Das Heyting-Pendant zur Boole'schen Algebra $\mathcal{F}_1$ ist also unendlich groß und sieht folgendermaßen aus:



[14] c ist Pseudokompement von a, wenn c das eindeutige größte Element mit $a \sqcap c = 0$ ist. Ein Komplement erfüllt zusätzlich noch $a \sqcup c = 1$.

In der klassischen Logik bleiben bis auf Äquivalenz nur $\bot$, A_0, $\neg A_0$ und $\top$ übrig: Alle Formeln oberhalb von $(A_0 \vee \neg A_0)$ sind äquivalent zu $\top$.

Entscheidbarkeit

Man kann zeigen, dass auch für die intuitionistische Aussagenlogik die Fragen nach Erfüllbarkeit, Äquivalenz oder logischer Folgerung entscheidbar sind, allerdings kann man nicht mehr einfach Wahrheitstafeln ausrechnen. Essentiell sind folgende Eigenschaften (die hier nicht bewiesen werden können), am Beispiel des Erfüllbarkeits- bzw. Tautologieproblems:

Satz 1.6.5 **(a)** *Es gibt ein endliches System von Regeln, aus denen sich (mit Substitution) alle intuitionistischen Tautologien (und nur Tautologien) ableiten lassen.*

(b) *Für jede Formel, die keine intuitionistische Tautologie ist. gibt es bereits ein endliches Modell, in der sie falsch ist.*

Das Entscheidungsverfahren für intuitionistische Tautologien funktioniert nun so: Man lässt zwei Maschinen gleichzeitig laufen. Die erste erzeugt aus dem endlichen Regelsystem systematisch alle Tautologien. Falls eine Formel gegebene Formel F eine Tautologie ist, wird diese Maschine irgendwann eine Ableitung dieser Tautologie aus den Regeln aufzeigen. Die zweite Maschine erzeugt systematisch alle endlichen Modelle (z. B. der Größe nach) und testet für alle Belegungen, ob F in allen Zuständen den Wert 1 bekommt. Falls F keine Tautologie ist, wird diese Maschine irgendwann ein Gegenbeispiel konstruiert haben.

Auch intuitionistisch gilt, dass eine Formel F genau dann eine Tautologie ist, wenn $\neg F$ nicht erfüllbar ist (und $\neg F$ ist genau dann eine Tautologie, wenn F nicht erfüllbar ist). Mit dem Tautologieproblem ist also auch das Erfällbarkeitsproblem entschieden.

Im Gegensatz zur klassischen Aussagenlogik folgt die Entscheidbarkeit nicht allein aus Teil (a) des Satzes, da keine explizite kanonische Normalform für intuitionistische Formeln bekannt ist.

2 Prädikatenlogik (erster Stufe)

Die *Prädikatenlogik* ist eine Erweiterung der Aussagenlogik, in der es zum einen möglich ist, Aussagenvariablen durch genaue Aussagen über eine gegebene Struktur zu ersetzen, und man zum andern durch Quantifikationen eine zusätzliche Ausdrucksstärke erhält. Quantifikationen werden durch *Quantoren* ausgedrückt und sagen etwas über die Anzahl der Elemente aus, auf die eine Formel zutrifft. Manche nennen die Prädikatenlogik daher auch *Quantorenlogik*. Bei der hier beschriebenen Logik handelt es sich genauer um die *Prädikatenlogik erster Stufe*, die nur Quantifikationen über Elemente zulässt. In der *Prädikatenlogik zweiter Stufe* kann man auch über Teilmengen quantifizieren, in der dritten Stufe über Mengen von Teilmengen, etc.

2.1 Syntax der Prädikatenlogik

Im Gegensatz zur Aussagenlogik gibt es in der Prädikatenlogik je nach ins Auge gefasster Anwendung verschiedene Sprachen, die einen allen gemeinsamen festen Anteil und einen für eine Sprache $\mathcal{L}$ spezifischen Anteil haben.

Zeichen – fester Anteil:

Individuenvariablen	Quantoren	Junktoren	Klammern
v_0	$\forall$	$\top$	(
v_1	$\exists$	$\bot$	)
v_2		$\neg$	
$\vdots$	Gleichheitszeichen	$\wedge$	
	$\doteq$	$\vee$	
		$\rightarrow$	
		$\leftrightarrow$	

Zeichen – $\mathcal{L}$-spezifischer Anteil:

Funktionszeichen	Relationszeichen
f_i für $i \in I$	R_j für $j \in J$

Jedes Funktions- und jedes Relationszeichen hat in $\mathcal{L}$ eine feste Stelligkeit $\in \mathbb{N}$

$\forall$ heißt Allquantor oder universeller Quantor, $\exists$ heißt Existenzquantor.

Die Indexmengen I und J der Funktions- bzw. Relationszeichen können endlich oder unendlich sein (insbesondere auch leer) und bestehen üblicherweise aus natürlichen Zahlen. Falls z. B. $I = \{0, 13, 35\}$, soll $\mathcal{L}$ also genau die Funktionszeichen f_0, f_{13} und f_{35} enthalten. Die Stelligkeit ordnet jedem Funktions- und jedem Relationszeichen eine natürliche Zahl zu, die aus dem Zeichen heraus nicht erkennbar ist (aber aus der Verwendung des Zeichens) und bei der Angabe einer Sprache $\mathcal{L}$ festgelegt werden muss.

Nullstellige Funktionszeichen können mit Konstanten identifiziert werden und werden dann gerne c_0, c_1, … geschrieben (Erläuterung siehe Seite 33). Nullstellige Relationszeichen können mit Aussagenvariablen identifiziert werden und können dann auch, wie in der Aussagenlogik, A_0, A_1, … geschrieben werden.[15] Einstellige Relationszeichen heißen auch Prädikate[16] und haben der Prädikatenlogik ihren Namen gegeben. In Anwendungen wird man häufig andere Namen für die Funktions- und Rekationszeichen wählen, etwa f, g, h oder auch $+, \circ, ^{-1}$ für Funktionszeichen; c, d, e oder auch $0, 1$ für Konstanten; R, S, T, A, B oder auch $\leqslant, \subseteq$ für Relationszeichen.

[15] Die meisten Autoren lassen keine 0-stelligen Relationszeichen zu, was den Nachteil hat, dass die Prädikatenlogik dann keine Erweiterung der Aussagenlogik ist.

[16] in der Verwendungsweise als „(besondere) Eigenschaft“ wie in Prädikatswein, Prädikatsexamen,…

Das Gleichheitszeichen wird mit einem Punkt notiert (Ziegler'sche Konvention), um es von dem metasprachlichen Gleichheitszeichen zu unterscheiden.

Beispiel: Eine für die Beschreibung der natürlichen Zahlen $\mathbb{N}$ geeignete Sprache ist $\mathcal{L}_{\mathbb{N}} = \{<, +, \cdot, 0, 1\}$, wobei $<$ ein zweistelliges Relatiosnzeichen ist, $+$ und $\cdot$ zweistellige Funktionszeichen und 0 und 1 nullstelliges Funktionszeichen (Konstanten).

Für Gruppen kann man die Sprache $\mathcal{L} = \{\circ, {}^{-1}, e\}$ verwenden, in der $\circ$ zweistelliges, ${}^{-1}$ einstelliges und e nullstelliges Funktionszeichen ist.

Nun werden zunächst *Terme*, dann *atomare Formeln* und schließlich beliebige *Formeln* in der prädikatenlogischen Sprache $\mathcal{L}$ definiert. Terme sind Ausdrücke, die für Elemente einer Struktur stehen werden und bekommen eine eigene violette Farbe. Formeln wird man als Aussagen über Strukturen interpretieren können; sie behalten die rote Formelfarbe der Aussagenlogik. Beide Definitionen nutzen eine ähnliche Induktion wie die Definition aussagenlogischer Formeln. Sowohl Terme als auch Formeln sind Zeichenketten und haben damit automatisch eine definierte Länge. Analog zur Aussagenlogik kann man für sie auch die Schachtelungstiefe definieren.

Definition 2.1.1 *$\mathcal{L}$-Terme sind alle Zeichenketten, die man nach folgenden Regeln erhält:*

- *Jede Individuenvariable ist ein $\mathcal{L}$-Term v_i (der Länge 1 und Tiefe 0).*
- *Für $\mathcal{L}$-Terme $\tau_1,\ldots,\tau_n$ und jedes n-stellige Funktionszeichen f_i in $\mathcal{L}$ ist $f_i\tau_1\ldots\tau_n$ ein $\mathcal{L}$-Term (der Länge $1+\lg(\tau_1)+\cdots+\lg(\tau_n)$ und Tiefe $1+\max\{\mathrm{tf}(\tau_1),\ldots,\mathrm{tf}(\tau_n)\}$).*[17]

Insbesondere ist (mit $n=0$) jede Konstante c_i der Sprache $\mathcal{L}$ ein $\mathcal{L}$-Term.

Terme sind hier also in der (klammerfreien) polnischen Notation geschrieben. Manche Autoren verwenden dagegen die übliche Funktionsschreibweise $f_i(\tau_1,\ldots,\tau_n)$ für Terme. In Anwendungen wird für gebräuchliche Funktionszeichen wie $+, \circ, {}^{-1}$ meist die traditionelle Schreibweise benutzt, also (v_0+v_1) statt $+v_0v_1$ oder (v_0^{-1}) statt ${}^{-1}v_0$; man braucht dann aber u. U. Klammern, um die eindeutige Lesbarkeit sicherzustellen.

Lemma 2.1.2 (eindeutige Lesbarkeit von Termen) *$\mathcal{L}$-Terme sind eindeutig lesbar, d. h. wenn $f_i\tau_1\ldots\tau_n = f_j\tau'_1\ldots\tau'_m$ $\mathcal{L}$-Terme sind, dann gilt $f_i = f_j$, $n=m$ und $\tau_i=\tau'_i$.*

Beweis: Übung. Vergleiche die eindeutige Lesbarkeit aussagenlogischer Formeln. □

Definition 2.1.3 *$\mathcal{L}$-Formeln sind alle Zeichenketten, die atomare $\mathcal{L}$-Formeln sind oder nach den unten folgenden Zusammensetzungsregeln konstruiert werden können.*

Atomare $\mathcal{L}$-Formeln sind:

- *$\top$ und $\bot$;*
- *$R_j\,\tau_1\ldots\tau_n$, wenn R_j ein n-stelliges Relationszeichen in $\mathcal{L}$ ist und $\tau_1,\ldots,\tau_n$ $\mathcal{L}$-Terme sind;*
- *$\tau_1 \doteq \tau_2$, wenn τ_1 und τ_2 $\mathcal{L}$-Terme sind.*

Zusammensetzungsregeln: Für $\mathcal{L}$-Formeln φ_1 und φ_2 sind auch die folgenden Zeichenketten $\mathcal{L}$-Formeln (für jede Individuenvariable v_i)

- [Junktorenregeln] $\neg\varphi_1$, $(\varphi_1\wedge\varphi_2)$, $(\varphi_1\vee\varphi_2)$, $(\varphi_1\to\varphi_2)$ *und* $(\varphi_1\leftrightarrow\varphi_2)$
- [Quantorenregeln] $\forall v_i\,\varphi_1$ *und* $\exists v_i\,\varphi_1$

[17] $f_i\tau_1\ldots\tau_n$ ist wieder zu verstehen als die Zeichenkette, die aus dem Zeichen f_i und den aneinandergereihten Zeichenketten für τ_1 bis τ_n besteht.

Auch hier wird für gebräuchliche (in der Regel binäre) Relationszeichen wie $\leqslant$ meist die traditionelle Schreibeweise benutzt, also $v_0 \leqslant v_1$ statt $\leqslant v_0 v_1$, so wie es in den obigen Regeln insbesondere auch für das Gleichheitszeichen vorgesehen ist.

Lemma 2.1.4 *$\mathcal{L}$-Formeln sind eindeutig lesbar.*

Beweis: Dies folgt im Wesentlichen aus früheren Beobachtungen: Aus der eindeutigen Lesbarkeit der Terme folgt auch, dass eine atomare Formel $R_j\,\tau_1 \ldots \tau_n$ „eindeutig lesbar“ ist (da der Term $f\,\tau_1 \ldots \tau_n$ mit einem n-stelligen Funktionszeichen f es ist). Wenn atomare Formeln $R_j\,\tau_1 \ldots \tau_n$ oder $\tau_1 \doteq \tau_2$ als Teilformeln einer Formel φ auftreten, sind sie durch den Formelanfang oder das Formelende von φ, Klammern, Junktoren oder Quantoren (mit zugehöriger Individuenvariable) eindeutig abgegrenzt (und können nicht unmittelbar aneinanderstoßen). Man kann sie also durch Aussagenvariablen ersetzen und erhält fast eine aussagenlogische Formel: Es bleiben ggf. noch Quantoren, die man mit ihren zugehörigen Individuenvariablen wie neue einstellige Junktoren ansehen kann. Der Beweis der eindeutigen Lesbarkeit aussagenlogischer Formeln funktioniert ebensogut mit zusätzlichen null-, ein- oder zweistelligen Junktoren und zeigt dann auch hier die eindeutige Lesbarkeit. □

Analog zur Aussagenlogik kann man *Teilformeln* von $\mathcal{L}$-Formeln definieren und zeigen, dass eine $\mathcal{L}$-Formel φ', die als zusammenhängende Zeichenkette in einer $\mathcal{L}$-Formel φ vorkommt, bereits eine Teilformel von φ ist.

Definition 2.1.5 *Der* Wirkungsbereich *(des Vorkommens) eines Quantors $\exists v_i$ bzw. $\forall v_i$ in einer $\mathcal{L}$-Formel φ ist diejenige Teilformel φ', vor der der Quantor unmittelbar steht.* [18]

Anders ausgedrückt: Beim Aufbauprozess der Formel wurde der Quantor durch Anwenden der Quantorregel auf φ' eingeführt.

Konvention: In der Kombination mit einem Quantor, also in der Symbolfolge $\exists v_i$ bzw. $\forall v_i$, soll das Zeichen v_i nicht als ein Vorkommen der Individuenvariablen v_i in einer Formel zählen. Als Vorkommen einer Individuenvariablen zählen nur die Vorkommen in einem Term. Die Variable unmittelbar hinter einem Quantor sollte man besser als eine Art Zeiger verstehen, der angibt, auf welche Individuenvariablen in seinem Wirkungsbereich sich der Quantor bezieht.

Definition 2.1.6 *Ein Vorkommen einer Individuenvariablen v_i in einer $\mathcal{L}$-Formel φ heißt frei, wenn es sich nicht im Wirkungsbereich eines Quantors $\exists v_i$ bzw. $\forall v_i$ befindet.*

Ein Vorkommen einer Individuenvariablen v_i heißt gebunden durch einen Quantor $\exists v_i$ bzw. $\forall v_i$, wenn es sich im Wirkungsbereich φ' dieses Quantors [19] *befindet und in φ' frei ist.*

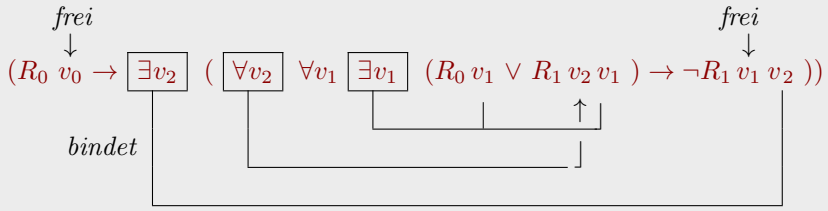
Die freien Variablen einer Formel φ sind diejenigen Individuenvariablen, die ein freies Vorkommen in φ haben.

*Ein $\mathcal{L}$-Term ohne Individuenvariablen heißt geschlossener $\mathcal{L}$-Term, eine $\mathcal{L}$-Formel ohne freie Variable heißt $\mathcal{L}$-Aussage (oder $\mathcal{L}$-*Satz *oder* geschlossene $\mathcal{L}$-Formel*).*

[18] Wenn in einer Formel ein Quantor wie $\exists v_i$ mehrfach vorkommt, haben diese Vorkommen jeweils verschiedene Wirkungsbereiche. Man spricht aber üblicherweise vom „Wirkungsbereich eines Quantors“ und meint damit ein konkretes Vorkommen des Quantors in der betreffenden Formel.

[19] Man beachte die Konvention aus der vorherigen Fußnote!

Beispiel: Sei R_0 einstelliges und R_1 zweistelliges Relationszeichen.



Notation: Für $n \geqslant 1$ soll die Schreibweise $\tau(v_{i_1}, \ldots, v_{i_n})$ für einen $\mathcal{L}$-Term τ bzw. $\varphi(v_{i_1}, \ldots, v_{i_n})$ für eine $\mathcal{L}$-Formel φ bedeuten, dass die Individuenvariablen $v_{i_1}, \ldots, v_{i_n}$ paarweise verschieden sind und dass alle in τ vorkommenden bzw. alle in φ *frei* vorkommenden Individuenvariablen enthalten sind in der Menge $\{v_{i_1}, \ldots, v_{i_n}\}$.

Es gibt keine speziellen Notationen dafür, dass ein Term geschlossen und eine Formel eine Aussage ist.

2.2 $\mathcal{L}$-Strukturen

Sei $\mathcal{L}$ eine prädikatenlogische Sprache.

Definition 2.2.1 *Eine $\mathcal{L}$-Struktur $\mathfrak{M}$ besteht aus*

- *einer nicht-leeren Menge M, dem Universum der Struktur;*
- *einer Funktion $f_i^{\mathfrak{M}} : M^n \to M$ für jedes n-stellige Funktionszeichen $f_i \in \mathcal{L}$ und $n \in \mathbb{N}$;*
- *einer Teilmenge $R_j^{\mathfrak{M}} \subseteq M^n$ für jedes n-stellige Relationszeichen $R_j \in \mathcal{L}$ und $n \in \mathbb{N}$.*

Man sagt, dass ein n-stelliges Funktionszeichen f_i durch die n-stellige Funktion $f_i^{\mathfrak{M}}$ interpretiert wird bzw. dass die Funktion $f_i^{\mathfrak{M}}$ Interpretation des Funktionszeichens f_i ist. Analog wird ein n-stelliges Relationszeichen R_j durch eine n-stellige Relation $R_j^{\mathfrak{M}}$ interpretiert. Eigentlich ist solch eine Relation R eine Eigenschaft von n-Tupeln: Sie trifft auf ein n-Tupel zu oder nicht (und man schreibt dann $R\, m_1 \ldots m_n$ für „R *trifft auf* $m_1 \ldots m_n$ *zu*“). Häufig aber werden (wie hier auch) Relationen mit ihren Graphen identifiziert, d. h. eine Relation wird als die Menge der n-Tupel aufgefasst, auf die sie zutrifft. Demgemäß schreibt man auch $(m_1, \ldots, m_n) \in R$.

Im Fall $n = 0$ ist $M^0 = \{\emptyset\}$. Eine 0-stellige Funktion $f : M^0 \to M$ ist also eine konstante Funktion und kann mit ihrem Bild $f(\emptyset) \in M$ identifiziert werden; ein 0-stelliges Funktionszeichen kann daher als Konstante verstanden werden. Ein 0-stellige Relation R trifft entweder auf das einzige 0-Tupel $\emptyset$ zu oder trifft nicht zu, je nachdem ob der Graph von R die ein-elementige Menge $\{\emptyset\}$ oder die leere Menge $\emptyset$ ist. Ein 0-stelliges Relationzeichen kann daher mit einer Aussagenvariablen gleichgesetzt werden: Trifft die interpretierende 0-stellige Relation auf $\emptyset$ zu, entspricht dies dem Wahrheitswert 1, trifft sie nicht zu, dem Wahrheitswert 0.

Beispiel: Sei $\mathcal{L}_1 = \{f_0, f_1\}$ mit einem zweistelligen Funktionszeichen f_0 und einem nullstelligen Funktionszeichen (d. h. einer Konstanten) f_1. Dann ist $\mathfrak{M}_1 := (\mathbb{Z}; +, 0)$ eine $\mathcal{L}_1$-Struktur mit Universum $\mathbb{Z}$ und den Interpretationen $f_0^{\mathfrak{M}_1} = +$ und $f_1^{\mathfrak{M}_1} = 0$. Analog kann man jede Gruppe und jedes Monoid auf natürliche Weise als $\mathcal{L}_1$-Struktur auffassen.

In der Definition von Strukturen ist aber keinerlei Gültigkeit von irgendwelchen Gesetzen gefordert Auch $\mathfrak{M}_2 := (\mathbb{R} \setminus \mathbb{Q}; g, -\frac{\pi}{17})$ ist eine $\mathcal{L}_1$-Struktur, wobei $f_0^{\mathfrak{M}_2} = g$ die Funktion sein soll, die (a, b) auf $|a|^b$ abbildet, wenn dies eine irrationale Zahl ist, und auf $\sqrt{2}$ sonst; und $f_1^{\mathfrak{M}_2} = -\frac{\pi}{17}$.

Umgekehrt kann man die additive Gruppe der ganzen Zahlen als $\mathcal{L}$-Struktur für verschiedene Sprachen $\mathcal{L}$ auffassen, z. B. nur als $\mathcal{L}_0$-Struktur mit $\mathcal{L}_0 = \{f_0\}$ oder als $\mathcal{L}_2$-Struktur mit $\mathcal{L}_2 = \{f_0, f_1, f_2\}$ und einem zusätzlichen einstelligen Funktionszeichen f_2, das durch die Inversenabbildung $z \mapsto -z$ interpretiert wird.

Unterstrukturen und Homomorphismen:

Sei $\mathfrak{M}$ eine $\mathcal{L}$-Struktur ist und $U \subseteq M$ eine nicht-leere Teilmenge, die unter allen Funktionen $f_i^{\mathfrak{M}}$ für $f_i \in \mathcal{L}$ abgeschlossen ist. Dann definiert man die $\mathcal{L}$-Unterstruktur $\mathfrak{U}$ mit Universum U durch:

- $f_i^{\mathfrak{U}} := f_i^{\mathfrak{M}} \restriction_{U^n}$ für jedes n-stellige Funktionszeichen $f_i \in \mathcal{L}$;
- $R_j^{\mathfrak{U}} := R_j^{\mathfrak{M}} \cap U^n$ für jedes n-stellige Relationszeichen $R_j \in \mathcal{L}$.

Eine Abbildung $\alpha : M \to N$ zwischen $\mathcal{L}$-Strukturen $\mathfrak{M}$ und $\mathfrak{N}$ heißt $\mathcal{L}$-Homomorphismus, falls

- $\alpha(f_i^{\mathfrak{M}}(m_1, \ldots, m_n)) = f_i^{\mathfrak{N}}(\alpha(m_1), \ldots, \alpha(m_n))$ und
- $(m_1, \ldots, m_n) \in R_j^{\mathfrak{M}} \implies (\alpha(m_1), \ldots, \alpha(m_n)) \in R_j^{\mathfrak{N}}$

für alle $n \in \mathbb{N}$, alle n-stelligen Funktionszeichen $f_i \in \mathcal{L}$ und n-stelligen Relationszeichen $R_j \in \mathcal{L}$ und alle $m_1, \ldots, m_n \in M$. Wenn zusätzlich in der zweiten Bedingung die Äquivalenz $(m_1, \ldots, m_n) \in R_j^{\mathfrak{M}} \iff (\alpha(m_1), \ldots, \alpha(m_n)) \in R_j^{\mathfrak{N}}$ gilt, heißt α starker $\mathcal{L}$-Homomorphismus.

Ein $\mathcal{L}$-Isomorphismus ist ein bijektiver $\mathcal{L}$-Homomorphismus α, dessen Umkehrabbildung α^{-1} ebenfalls ein $\mathcal{L}$-Homomorphismus ist. Äquivalent ist α starker bijektiver $\mathcal{L}$-Homomorphismus.

Beispiel: (a) Wenn man im Beispiel oben die $\mathcal{L}_0$-Struktur $(\mathbb{Z}; +)$ betrachtet, sind $\mathbb{N}$ und $\mathbb{N} \setminus \{0\}$ Universen von $\mathcal{L}_0$-Unterstrukturen, da unter $+$ abgeschlossen. Im Falle der $\mathcal{L}_1$-Struktur $(\mathbb{Z}; +, 0)$ ist $\mathbb{N}$ Universum einer $\mathcal{L}_1$-Unterstruktur, nicht aber $\mathbb{N} \setminus \{0\}$. Schließlich ist $\mathbb{N}$ nicht mehr Universum einer $\mathcal{L}_2$-Unterstruktur von $(\mathbb{Z}; +, 0, -)$.

In Mathematik II wurde gezeigt, dass in allen drei Sprachen die $\mathcal{L}_i$-Homomorphismen zwischen Gruppen (in der jeweils natürlichen Interpretation als $\mathcal{L}_i$-Struktur) gerade die Gruppenhomomorphismen sind, während bei Monoiden die $\mathcal{L}_2$-Homomorphismen die Monoidhomomorphismen sind, es aber $\mathcal{L}_0$-Homomorphismen geben kann, die keine Monoidhomomorphismen sind.

(b) Boole'sche Algebren $\mathfrak{B} = (B; \sqcap, \sqcup, ^{\mathrm{c}}, 1, 0)$ haben eine natürliche Sprache mit zwei zweistelligen, einem einstelligen Funktionszeichen und zwei Konstanten. Ebenso kann man sie aber als partielle Ordnungen $(B; \leqslant)$ in einer Sprache mit einem zweistelligen Relationszeichen beschreiben. Die Unterstrukturen und Homomorphismen sind je nach Wahl der Sprache verschieden. Die Isomorphismen sind allerdings dieselben.

2.3 Semantik der Prädikatenlogik

Definition 2.3.1 *Eine Belegung der Individuenvariablen mit Werten in einer $\mathcal{L}$-Struktur $\mathfrak{M}$ (kurz: Belegung) ist eine Abbildung $\beta : \{v_i \mid i \in \mathbb{N}\} \to M$.*

Aus dem Kontext sollte immer klar sein, ob es sich bei einer Belegung um eine Belegung von Aussagenvariablen mit Wahrheitswerten oder um eine Belegung der Individuenvariablen mit Elementen einer Struktur handelt. „Variable“ wird ab sofort kurz für „Individuenvariable“ stehen.

Definition 2.3.2 *Sei $\mathcal{L}$ eine prädikatenlogische Sprache, $\mathfrak{M}$ eine $\mathcal{L}$-Struktur und β eine Belegung der Individuenvariablen. Dann definiert jeder $\mathcal{L}$-Term τ ein Element $\tau^{\mathfrak{M}}[\beta] \in M$, die Interpretation des Terms τ in $\mathfrak{M}$ unter β:*

- $v_i^{\mathfrak{M}}[\beta] \;:=\; \beta(v_i)$
- $(f_i\tau_1 \dots \tau_n)^{\mathfrak{M}}[\beta] \;:=\; f_i^{\mathfrak{M}}\big(\tau_1^{\mathfrak{M}}[\beta], \dots, \tau_n^{\mathfrak{M}}[\beta]\big)$ *für n-stelliges Funktionszeichen f_i.*

Lemma 2.3.3 *Falls zwei Belegungen β und β' auf den in einem Term τ vorkommenden Variablen übereinstimmen, so ist $\tau^{\mathfrak{M}}[\beta] = \tau^{\mathfrak{M}}[\beta']$.*

Beweis: Klar (oder formaler Beweis über den Aufbau der Terme). □

Definition 2.3.4 *Sei $\mathcal{L}$ eine prädikatenlogische Sprache, $\mathfrak{M}$ eine $\mathcal{L}$-Struktur und β eine Belegung in $\mathfrak{M}$. Dann bedeute $\mathfrak{M} \vDash \varphi\,[\beta]$, dass die $\mathcal{L}$-Formel φ in $\mathfrak{M}$ gilt, wenn die freien Variablen wie von β angegeben interpretiert werden. Dies wird wie folgt per Induktion über den Aufbau der Formeln definiert:*[20]

- $\mathfrak{M} \vDash R_j\tau_1 \dots \tau_n\,[\beta] \;:\Longleftrightarrow\; \big(\tau_1^{\mathfrak{M}}[\beta], \dots, \tau_n^{\mathfrak{M}}[\beta]\big) \in R_j^{\mathfrak{M}}$ *für n-stellige Relationszeichen R_j*
- $\mathfrak{M} \vDash \tau_1 \doteq \tau_2 \quad [\beta] \;:\Longleftrightarrow\; \tau_1^{\mathfrak{M}}[\beta] = \tau_2^{\mathfrak{M}}[\beta]$
- *Für Junktoren wird die Gültigkeit der Relation über deren Wahrheitswertfunktionalität definiert, d. h.*
  - *stets gilt* $\mathfrak{M} \vDash \top\,[\beta]$ *und* $\mathfrak{M} \nvDash \bot\,[\beta]$
  - $\mathfrak{M} \vDash \neg\varphi \quad [\beta] \;:\Longleftrightarrow\; \mathfrak{M} \nvDash \varphi\,[\beta]$
  - $\mathfrak{M} \vDash (\varphi_1 \wedge \varphi_2)\,[\beta] \;:\Longleftrightarrow\; \big(\mathfrak{M} \vDash \varphi_1\,[\beta]$ *und* $\mathfrak{M} \vDash \varphi_2\,[\beta]\big)$
  - *etc.*
- $\mathfrak{M} \vDash \exists v_i\,\varphi\;[\beta] \;:\Longleftrightarrow$ *es gibt ein $m \in M$ mit* $\mathfrak{M} \vDash \varphi\,[\beta\frac{m}{v_i}]$
- $\mathfrak{M} \vDash \forall v_i\,\varphi\;[\beta] \;:\Longleftrightarrow$ *für alle $m \in M$ gilt* $\mathfrak{M} \vDash \varphi\,[\beta\frac{m}{v_i}]$

  wobei $\beta\frac{m}{v_i}$ die Belegung β' ist mit $\beta'(v_j) := \begin{cases} m & \text{falls } i = j \\ \beta(v_j) & \text{falls } i \neq j \end{cases}$

Satz 2.3.5 *Falls zwei Belegungen β und β' auf den in einer $\mathcal{L}$-Formel φ vorkommenden freien Variablen übereinstimmen, so gilt $\mathfrak{M} \vDash \varphi\,[\beta] \iff \mathfrak{M} \vDash \varphi\,[\beta']$.*

Insbesondere gilt, dass für $\mathcal{L}$-Aussagen φ die Gültigkeit von $\mathfrak{M} \vDash \varphi\,[\beta]$ unabhängig von der Wahl von β ist. Man schreibt dann $\mathfrak{M} \vDash \varphi$.

Sprechweise: Für $\mathfrak{M} \vDash \varphi$ sagt man: „φ *trifft in* $\mathfrak{M}$ *zu*“, „φ *gilt in* $\mathfrak{M}$“, „φ *ist wahr in* $\mathfrak{M}$“ oder „$\mathfrak{M}$ *erfüllt* φ“, „$\mathfrak{M}$ *ist Modell von* φ“. Im Fall von $\mathfrak{M} \vDash \varphi\,[\beta]$ fügt man jeweils ein „*unter* β“ an.

Beweis: Beweis per Induktion über den Aufbau der Formeln, und zwar gleichzeitig für alle möglichen Belegungen (und nicht für feste β, β'):

Für $\top$ und $\bot$ ist die Aussage trivial; für andere atomare $\mathcal{L}$-Formeln gilt es nach Lemma 2.3.3.

Die Junktorenschritte sind wiederum klar nach Definition 2.3.4, da die freien Variablen von $\varphi = \neg\psi$ die gleichen wie die von ψ sind, und die freien Variablen von ψ_1 und ψ_2 jeweils enthalten sind in den freien Variablen von $\varphi = (\psi_1 * \psi_2)$ für einen beliebigen zweistelligen Junktor $*$.

[20] Und damit wird automatisch die verneinte Relation „$\mathfrak{M} \nvDash \varphi\,[\beta]$“ mitdefiniert.

Sei nun $\varphi = \exists v_i\, \psi$ und β stimme mit β' auf den freien Variablen von φ überein. Falls $\mathfrak{M} \vDash \varphi\,[\beta]$, dann gibt es ein $m_0 \in M$ mit $\mathfrak{M} \vDash \psi\,[\beta\frac{m_0}{v_i}]$. Die freien Variablen von ψ sind die freien Variablen von φ zusammen mit v_i: also stimmen $\beta\frac{m_0}{v_i}$ und $\beta'\frac{m_0}{v_i}$ auf ihnen überein. Nach Induktion gilt daher $\mathfrak{M} \vDash \psi\,[\beta'\frac{m_0}{v_i}]$ und somit $\mathfrak{M} \vDash \varphi\,[\beta']$. Analog für den Allquantor. □

Substitutionsregeln

Es folgen nun einige Substututionsregeln, die alle eine Verträglichkeit der Syntax mit der Semantik ausdrücken und die man als Ausdehnung des Kompositionalitätsprinzips verstehen kann. Sie haben die Form: Wenn eine Formel aus einer anderen dadurch entsteht, dass man Variablen durch Teilformeln ersetzt, ändert sich die Auswertung der Formel in der gleichen Weise durch Einsetzen der Auswertung der Teilformel.

Da die Prädikatenlogik eine Erweiterung der Aussagenlogik ist, also die Bildungsregeln aussagenlogischer Formeln in der Prädikatenlogik weiter Gültigkeit haben, kann man in einer aussagenlogischen Formel F Aussagenvariablen A_i durch $\mathcal{L}$-Formeln φ_i ersetzen und erhält eine $\mathcal{L}$-Formel $F\frac{\varphi_1}{A_1}\ldots\frac{\varphi_n}{A_n}$.

Lemma 2.3.6 (Uniforme Substitution) *Falls $\varphi_1,\ldots,\varphi_n$ $\mathcal{L}$-Formeln sind, $F(A_1,\ldots,A_n)$ eine aussagenlogische Formel ist und $\mathfrak{M}$ eine $\mathcal{L}$-Struktur mit Belegung β, dann gilt*

$$\mathfrak{M} \vDash F\,\frac{\varphi_1}{A_1}\ldots\frac{\varphi_n}{A_n}\,[\beta] \iff \beta_\varphi(F) = 1$$

wobei β_φ die (partielle) Belegung der Aussagenvariablen mit $\beta_\varphi(A_i) = 1 \iff \mathfrak{M} \vDash \varphi_i\,[\beta]$ ist.

Beweis: Klar nach Definition 2.3.4. □

Insbesondere folgt daraus, dass man sich in der Prädikatenlogik auf ein vollständiges Junktorensystem zurückziehen kann und dass in vielen Definitionen und Beweisen über den Aufbau von Formeln die Junktorenschritte auf z. B. Negation und Konjunktion beschränkt werden können (wie in Definition 2.3.4 bereits angedeutet).

Notation: Für $\mathcal{L}$-Terme $\tau(v_1,\ldots,v_n)$, $\mathcal{L}$-Strukturen $\mathfrak{M}$ und $m_1,\ldots,m_n \in M$ sei

$$\tau^{\mathfrak{M}}[m_1,\ldots,m_n] := \tau^{\mathfrak{M}}[\beta],$$

wobei β eine beliebige Belegung mit $\beta(v_i) = m_i$ ist (wohldefiniert nach Lemma 2.3.3).

Jeder Term $\tau = \tau(v_1,\ldots,v_n)$ definiert in einer $\mathcal{L}$-Struktur $\mathfrak{M}$ also eine n-stellige Abbildung $\tau^{\mathfrak{M}} : M^n \to M$ mit $(m_1,\ldots,m_n) \mapsto \tau^{\mathfrak{M}}[m_1,\ldots,m_n]$.

Für $\mathcal{L}$-Terme τ und σ sei $\tau\frac{\sigma}{v_i}$ der $\mathcal{L}$-Term, den man dadurch erhält, dass man jedes Vorkommen von v_i in τ durch σ ersetzt.

Lemma 2.3.7 (Substitutionslemma für Terme) *Sei σ ein $\mathcal{L}$-Term und β eine Belegung in einer $\mathcal{L}$-Struktur $\mathfrak{M}$. Dann gilt für alle $\mathcal{L}$-Terme τ:*

$$\left(\tau\frac{\sigma}{v_i}\right)^{\mathfrak{M}}[\beta] \;=\; \tau^{\mathfrak{M}}\left[\beta\,\frac{\sigma^{\mathfrak{M}}[\beta]}{v_i}\right]$$

Beweis: Klar durch Nachdenken darüber, was die Formel des Lemmas aussagt. □

Für Terme $\tau(v_1)$ bedeutet das Lemma so viel wie $(\tau\frac{\sigma}{v_1})^{\mathfrak{M}} = \tau^{\mathfrak{M}} \circ \sigma^{\mathfrak{M}}$. Die allgemeine Version kann man folgendermaßen schreiben, wenn n so groß gewählt ist, dass alle Variablen von τ und σ unter $v_1, \ldots, v_n$ sind:

$$\big(\tau\frac{\sigma}{v_i}\big)^{\mathfrak{M}}(m_1, \ldots, m_n) = \tau^{\mathfrak{M}}\big(m_1, \ldots, m_{i-1}, \sigma^{\mathfrak{M}}(m_1, \ldots, m_n), m_{i+1}, \ldots, m_n\big)$$

Notation: Für $\mathcal{L}$-Formeln $\varphi(v_1, \ldots, v_n)$, $\mathcal{L}$-Strukturen $\mathfrak{M}$ und $m_1, \ldots, m_n \in M$ setzt man

$$\mathfrak{M} \vDash \varphi\,[m_1, \ldots, m_n] : \Longleftrightarrow\ \mathfrak{M} \vDash \varphi\,[\beta]$$

wobei β eine beliebige Belegung mit $\beta(v_i) = m_i$ ist (wohldefiniert nach Satz 2.3.5).

Jede Formel $\varphi = \varphi(v_1, \ldots, v_n)$ definiert in einer $\mathcal{L}$-Struktur $\mathfrak{M}$ also eine n-stellige Relation $\varphi^{\mathfrak{M}} := \big\{(m_1, \ldots, m_n) \in M^n \mid \mathfrak{M} \vDash \varphi\,[m_1, \ldots, m_n]\big\}$.

Für $\mathcal{L}$-Formeln φ und $\mathcal{L}$-Terme σ sei $\varphi\frac{\sigma}{v_i}$ die $\mathcal{L}$-Formel, die man dadurch erhält, dass man jedes *freie* Vorkommen von v_i in φ durch σ ersetzt.

Definition 2.3.8 *Eine Individuenvariable v_i heißt frei für σ in φ, falls bei der Ersetzung $\varphi\frac{\sigma}{v_i}$ keine Variable von σ durch einen Quantor von φ gebunden wird.*

Formale Definition über den Aufbau der Formeln:

Die Variable v_i ist frei für σ in φ, falls (a) entweder v_i nicht frei in φ ist oder falls (b) v_i frei in φ ist und einer der folgenden Fälle gilt:

- *φ ist atomar;*
- *$\varphi = \neg\psi$ und v_i ist frei für σ in ψ;*
- *$\varphi = (\psi * \psi')$ und v_i ist frei für σ in ψ und in ψ' ($*$ ist beliebiger zweistelliger Junktor);*
- *$\varphi = \exists v_j\,\psi$ oder $= \forall v_j\,\psi$ und v_i ist frei für σ in ψ und v_j kommt in σ nicht vor.*

Lemma 2.3.9 (Substitutionslemma für Formeln) *Sei σ ein $\mathcal{L}$-Term und β eine Belegung in einer $\mathcal{L}$-Struktur $\mathfrak{M}$. Dann gilt für alle $\mathcal{L}$-Formeln φ und alle Variablen v_i, die frei für σ in φ sind:*

$$\mathfrak{M} \vDash \varphi\frac{\sigma}{v_i}\,[\beta] \iff \mathfrak{M} \vDash \varphi\,\big[\beta\,\frac{\sigma^{\mathfrak{M}}[\beta]}{v_i}\big]$$

Beweis: Beweis per Induktion über den Aufbau der Formeln. Für $\varphi = \top$ oder $\bot$ gilt die Aussage trivialerweise. Sei $\varphi = R_j\tau_1 \ldots \tau_n$ (und analog $\varphi = \tau_1 \doteq \tau_2$). Dann ist nach dem Substitutionslemma für Terme:

$$\begin{aligned}\mathfrak{M} \vDash \varphi\frac{\sigma}{v_i}\,[\beta] &\iff \Big((\tau_1\frac{\sigma}{v_i})^{\mathfrak{M}}[\beta], \ldots, (\tau_n\frac{\sigma}{v_i})^{\mathfrak{M}}[\beta]\Big) \in R_j^{\mathfrak{M}} \\ &\iff \Big(\tau_1^{\mathfrak{M}}\,\big[\beta\,\frac{\sigma^{\mathfrak{M}}[\beta]}{v_i}\big], \ldots, \tau_n^{\mathfrak{M}}\,\big[\beta\,\frac{\sigma^{\mathfrak{M}}[\beta]}{v_i}\big]\Big) \in R_j^{\mathfrak{M}} \iff \mathfrak{M} \vDash \varphi\,\big[\beta\,\frac{\sigma^{\mathfrak{M}}[\beta]}{v_i}\big]\end{aligned}$$

Die Junktorenschritte sind unproblematisch.

Sei also $\varphi = \exists v_j\,\psi$. Wegen der Annahme, dass v_i frei für σ in φ ist, kommt v_j in σ nicht vor. Wenn v_i nicht frei in φ ist, ist das Substitutionslemma trivial. Sei also v_i frei in φ. Dann muss $i \neq j$ sein. Nun gilt:

$$\begin{aligned}
\mathfrak{M} \vDash \exists v_j\, \psi \frac{\sigma}{v_i}\,[\beta] &\iff \text{es gibt ein } m_0 \in M \text{ mit } \mathfrak{M} \vDash \psi \frac{\sigma}{v_i}\,\Big[\beta \frac{m_0}{v_j}\Big] \\
\textit{(Induktionsvoraussetzung)} &\iff \text{es gibt ein } m_0 \in M \text{ mit } \mathfrak{M} \vDash \psi \left[\beta \frac{m_0}{v_j} \frac{\sigma^{\mathfrak{M}}[\beta \frac{m_0}{v_j}]}{v_i}\right] \\
\textit{(v_j kommt in σ nicht vor)} &\iff \text{es gibt ein } m_0 \in M \text{ mit } \mathfrak{M} \vDash \psi \left[\beta \frac{m_0}{v_j} \frac{\sigma^{\mathfrak{M}}[\beta]}{v_i}\right] \\
\textit{($i \neq j$)} &\iff \text{es gibt ein } m_0 \in M \text{ mit } \mathfrak{M} \vDash \psi \left[\beta \frac{\sigma^{\mathfrak{M}}[\beta]}{v_i} \frac{m_0}{v_j}\right] \\
&\iff \mathfrak{M} \vDash \exists v_j\, \psi \left[\beta \frac{\sigma^{\mathfrak{M}}[\beta]}{v_i}\right]
\end{aligned}$$

Analog für den Allquantor. □

Für $\mathcal{L}$-Formeln φ, ψ und eine Teilformel φ' von φ sei $\varphi\frac{\psi}{\varphi'}$ die $\mathcal{L}$-Formel, die man dadurch erhält, dass man die Teilformel φ' in φ durch ψ ersetzt. [21]

Lemma 2.3.10 (Äquivalente Substitution)
Falls φ' eine zu ψ äquivalente Teilformel von φ ist, dann ist $\varphi\frac{\psi}{\varphi'}$ äquivalent zu φ.

Beweis: Klar nach der Definition der logischen Äquivalenz in 2.4.1 und Definition 2.3.4. □

Logische Äquivalenz prädikatenlogischer Formeln wird zwar erst im nächsten Abschnitt definiert, entspricht aber der Intuition. Damit alle Substitutionsregeln an einem Ort versammelt sind, habe ich das Lemma der äquivalenten Substitution vorgezogen.

2.4 Allgemeingültige Formeln und Gesetze der Prädikatenlogik

Definition 2.4.1 *In den folgenden drei Definitionen (a) bis (c) ist stets gemeint, dass die angegebene Bedingung* für alle *$\mathcal{L}$-Strukturen $\mathfrak{M}$ und* alle *Belegungen β gelten soll.*

(a) *Eine $\mathcal{L}$-Formel φ heißt allgemeingültig, in Zeichen: $\vDash \varphi$, falls $\mathfrak{M} \vDash \varphi\,[\beta]$.*

(b) *Zwei $\mathcal{L}$-Formeln φ und ψ heißen (logisch) äquivalent zueinander, in Zeichen: $\varphi \sim \psi$, falls*

$$\mathfrak{M} \vDash \varphi\,[\beta] \iff \mathfrak{M} \vDash \psi\,[\beta].$$

(c) *Eine $\mathcal{L}$-Formel ψ folgt (logisch) aus einer Menge $\{\varphi_i \mid i \in I\}$ von $\mathcal{L}$-Formeln, in Zeichen: $\{\varphi_i \mid i \in I\} \vDash \psi$, falls gilt:*

$$\mathfrak{M} \vDash \varphi_i\,[\beta] \text{ für alle } i \in I \implies \mathfrak{M} \vDash \psi\,[\beta].$$

(a) *Eine $\mathcal{L}$-Formel φ heißt erfüllbar, falls es eine $\mathcal{L}$-Struktur $\mathfrak{M}$ und eine Belegungen β mit $\mathfrak{M} \vDash \varphi\,[\beta]$ gibt.*

Ich benutze für die Schreibweisen entsprechende Konventionen wie in der Aussagenlogik. Lemma 1.2.3 gilt entsprechend für die Prädikatenlogik. Logische Äquivalenzen etc. nenne ich in der Prädikatenlogik meist *Gesetze* und reserviere *Regel* für „Meta-Gesetze“ wie etwa in Lemma 2.4.6. Eine Besonderheit gilt für Formeln mit freien Variablen:

[21] Genauer ist wieder, wie in der Aussagenlogik, ein konkretes Vorkommen von φ' in φ gemeint.

Eine Formel $\varphi(v_1, \ldots, v_n)$ mit freien Variablen ist nach Definition genau dann allgemeingültig – weil sie für alle Belegungen gelten muss –, wenn die universell abquantifizierte Formel $\forall v_1 \ldots \forall v_n \, \varphi(v_1, \ldots, v_n)$ allgemeingültig ist. Insbesondere gilt also

$$\varphi(v_1, \ldots, v_n) \sim \psi(v_1, \ldots, v_n) \iff \models \forall v_1 \ldots \forall v_n \, (\varphi \leftrightarrow \psi).$$
$$\text{aber !!} \qquad \not\Longleftrightarrow \quad \forall v_1 \ldots \forall v_n \, \varphi \sim \forall v_1 \ldots \forall v_n \, \psi$$

Zum Beispiel ist $\forall v_1 \, P v_1 \sim \forall v_0 \, P v_0$, aber $P v_0$ und $P v_1$ sind nicht äquivalent zueinander.

Satz 2.4.2 (Logische Gesetze für Quantoren)

Unnötige Quantoren:

Wenn v_i nicht frei in φ ist, dann: $\quad \exists v_i \, \varphi \sim \forall v_i \, \varphi \sim \varphi$

Umbenennung gebundener Variablen:

Wenn v_i frei für v_j in φ ist und v_j nicht frei in φ ist, dann:

$\exists v_i \, \varphi \sim \exists v_j \, \varphi\frac{v_j}{v_i}$ $\qquad$ $\forall v_i \, \varphi \sim \forall v_j \, \varphi\frac{v_j}{v_i}$

Verhältnis von Quantoren untereinander:

$\forall v_0 \, \varphi \models \exists v_0 \, \varphi \quad (*)$ $\qquad$ $\exists v_0 \forall v_1 \, \varphi \models \forall v_1 \exists v_0 \, \varphi \quad (*)$

$\exists v_0 \exists v_1 \, \varphi \sim \exists v_1 \exists v_0 \, \varphi$ $\qquad$ $\forall v_0 \forall v_1 \, \varphi \sim \forall v_1 \forall v_0 \, \varphi$

Dualität (Verallgemeinerte Regeln von de Morgan*):*

$\neg \exists v_o \, \varphi \sim \forall v_0 \, \neg\varphi$ $\qquad$ $\neg \forall v_o \, \varphi \sim \exists v_0 \, \neg\varphi$

Weitere Vertauschungen von Quantoren mit Junktoren:

$\forall v_o \, (\varphi \wedge \psi) \sim (\forall v_o \, \varphi \wedge \forall v_o \, \psi)$ $\qquad$ $\exists v_o \, (\varphi \wedge \psi) \models (\exists v_o \, \varphi \wedge \exists v_o \, \psi) \quad (*)$

$\exists v_o \, (\varphi \vee \psi) \sim (\exists v_o \, \varphi \vee \exists v_o \, \psi)$ $\qquad$ $(\forall v_o \, \varphi \vee \forall v_o \, \psi) \models \forall v_o \, (\varphi \vee \psi) \quad (*)$

Falls v_0 nicht frei in φ ist:

$\forall v_o \, (\varphi \vee \psi) \sim (\varphi \vee \forall v_o \, \psi)$ $\qquad$ $\exists v_o \, (\varphi \wedge \psi) \sim (\varphi \wedge \exists v_o \, \psi)$

$\forall v_o \, (\varphi \wedge \psi) \sim (\varphi \wedge \forall v_o \, \psi)$ $\qquad$ $\exists v_o \, (\varphi \vee \psi) \sim (\varphi \vee \exists v_o \, \psi)$

„Quantorendefinition“:

$\varphi\frac{\tau}{v_i} \models \exists v_i \, \varphi$, falls v_i frei für τ in φ $\qquad$ $\forall v_i \, \varphi \models \varphi\frac{\tau}{v_i}$, falls v_i frei für τ in φ

Bei den mit $(*)$ gekennzeichneten logischen Folgerungen gilt die Umkehrung nicht!

Bei der Umbenennung gebundener Variablen muss man vorsichtig sein. In den beiden folgenden Fällen darf v_0 nicht durch v_1 ersetzt werden: $\varphi = \forall v_0 \, v_0 \doteq v_1$ ist nicht äquivalent zu $\forall v_1 \, v_1 \doteq v_1$ (v_1 ist frei in φ), und $\psi = \forall v_0 \forall v_1 \, v_0 \doteq v_1$ ist nicht äquivalent zu $\forall v_1 \forall v_1 \, v_1 \doteq v_1$ (v_0 ist nicht frei für v_1 in ψ). Also sind die beiden Bedingungen notwendig!

Ebenso sieht man bei den „Quantorendefinitionen“, dass die Bedingung notwendig ist: Beispielweise kann man nicht aus $\forall v_1 \, v_1 \doteq v_1$ auf $\exists v_0 \forall v_1 \, v_0 \doteq v_1$ schließen.

Für jede $\mathcal{L}$-Struktur $\mathfrak{M}$ kann man die Sprache um neue Konstanten für Elemente aus M erweitern: Die Spracherweiterung $\mathcal{L}_M$ besteht aus $\mathcal{L}$ zusammen mit neuen Konstanten c_m für jedes Element $m \in M$, und $\mathfrak{M}$ wird zu einer $\mathcal{L}_M$-Struktur $\mathfrak{M}_M$ durch $c_m^{\mathfrak{M}_M} = m$. Ist das Universum

$M = \{m_1, \dots, m_n\}$ endlich, dann gilt für eine $\mathcal{L}$-Aussage $\forall v_0\, \varphi$ bzw. $\exists v_0\, \varphi$:

$$\mathfrak{M} \vDash \forall v_0\, \varphi \iff \mathfrak{M}_M \vDash \left(\varphi\tfrac{c_{m_1}}{v_0} \wedge \dots \wedge \varphi\tfrac{c_{m_n}}{v_0}\right)$$
$$\text{und} \quad \mathfrak{M} \vDash \exists v_0\, \varphi \iff \mathfrak{M}_M \vDash \left(\varphi\tfrac{c_{m_1}}{v_0} \vee \dots \vee \varphi\tfrac{c_{m_n}}{v_0}\right)$$

In diesem Sinne ist also der Allquantor eine Art verallgemeinerte Konjunktion und der Existenzquantor eine Art verallgemeinerte Disjunktion.[22] Dies erklärt auch die verallgemeinerten *de Morgan*'schen Regeln und die Vertauschbarkeit des Allquantors mit der Konjunktion und des Existenzquantors mit der Disjunktion.

Beweis einiger der Quantorengesetze aus Satz 2.4.2:

- Erstes Dualitätsgesetz:

$$\begin{aligned}
\mathfrak{M} \vDash \neg\exists v_0\, \varphi\,[\beta] \iff \mathfrak{M} \nvDash \exists v_0\, \varphi\,[\beta] &\iff \text{es gibt kein } m \in M \text{ mit } \mathfrak{M} \vDash \varphi\tfrac{m}{v_0}\,[\beta] \\
&\iff \text{für alle } m \in M \text{ gilt } \mathfrak{M} \nvDash \varphi\tfrac{m}{v_0}\,[\beta] \\
&\iff \text{für alle } m \in M \text{ gilt } \mathfrak{M} \vDash \neg\varphi\tfrac{m}{v_0}\,[\beta] \\
&\iff \mathfrak{M} \vDash \forall v_0 \neg\varphi\,[\beta]
\end{aligned}$$

- Das zweite Dualitätsgesetz folgt daraus mit den Substitutionsprinzipien:

$$\neg\forall v_0\, \varphi \;\sim\; \neg\forall v_0\, \neg\neg\varphi \;\sim\; \neg\neg\exists v_0\, \neg\varphi \;\sim\; \exists v_0\, \neg\varphi$$

- Das Gesetz für „unnötige Quantoren“ folgt unmittelbar aus Satz 2.3.5.

- Umbenennung gebundener Variablen:

$$\begin{aligned}
\mathfrak{M} \vDash \exists v_j\, \varphi\tfrac{v_j}{v_i}\,[\beta] &\iff \text{es gibt ein } m \in M \text{ mit } \mathfrak{M} \vDash \varphi\tfrac{v_j}{v_i}\,[\beta\tfrac{m}{v_j}] \\
&\quad \overbrace{\phantom{\beta\tfrac{m}{v_j}(v_j)}}^{=m} \\
\textit{(v_i frei für v_j in φ + Substitutionslemma 2.3.9)} &\iff \text{es gibt ein } m \in M \text{ mit } \mathfrak{M} \vDash \varphi\,[\beta\tfrac{m}{v_j}\tfrac{\beta\frac{m}{v_j}(v_j)}{v_i}] \\
\textit{(v_j nicht frei in φ)} &\iff \text{es gibt ein } m \in M \text{ mit } \mathfrak{M} \vDash \varphi\,[\beta\tfrac{m}{v_i}] \\
&\iff \mathfrak{M} \vDash \exists v_i\, \varphi\,[\beta]
\end{aligned}$$

- $\exists$-Definition: Angenommen $\mathfrak{M} \vDash \varphi\tfrac{\tau}{v_i}\,[\beta]$. Wegen der Freiheitsannahme folgt mit dem Substitutionslemma 2.3.9 $\mathfrak{M} \vDash \varphi\,[\beta\tfrac{\tau^{\mathfrak{M}}[\beta]}{v_i}]$, d. h. $\mathfrak{M} \vDash \varphi\,[\beta\tfrac{m}{v_i}]$ mit $m = \tau^{\mathfrak{M}}[\beta]$. Insbesondere gibt es solch ein m, mithin $\mathfrak{M} \vDash \exists v_i\, \varphi\,[\beta]$.

- Durch die Dualität der Quantoren lässt sich die „$\forall$-Version“ eines Gesetzes auf die „$\exists$-Version“ zurückführen, wie am Beispiel des zweiten Dualitätsgesetzes gezeigt.

- Gegenbeispiele zu den fehlenden Umkehrungen: In $\mathfrak{M} = (\mathbb{Z}; <, 0)$ sieht man z. B.
  - $\mathfrak{M} \vDash \exists v_0\, v_0 \doteq 0$, aber $\mathfrak{M} \nvDash \forall v_0\, v_0 \doteq 0$.
  - $\mathfrak{M} \vDash \forall v_1 \exists v_0\, v_0 < v_1$, aber $\mathfrak{M} \nvDash \exists v_0 \forall v_1\, v_0 < v_1$.
  - $\mathfrak{M} \vDash \forall v_0\, (v_0 \doteq 0 \vee \neg v_0 \doteq 0)$, aber $\mathfrak{M} \nvDash (\forall v_0\, v_0 \doteq 0 \vee \forall v_0\, \neg v_0 \doteq 0)$.
  - $\mathfrak{M} \vDash (\exists v_0\, v_0 < 0 \wedge \exists v_0\, 0 < v_0)$, aber $\mathfrak{M} \nvDash \exists v_0\, (v_0 < 0 \wedge 0 < v_0)$.

- Alle anderen Gesetze ergeben sich unmittelbar aus der Anwendung der Definitionen. Bei $\forall v_0\, \varphi \vDash \exists v_0\, \varphi$ ist die Konvention, dass Strukturen nicht leer sein dürfen, entscheidend. □

[22] Manche Autoren schreiben daher $\bigwedge$ für $\forall$ und $\bigvee$ für $\exists$.

Folgerung 2.4.3 $\{\exists\}$ *und* $\{\forall\}$ *sind jeweils* vollständige Quantorensysteme*, d. h. jede $\mathcal{L}$-Formel ist logisch äquivalent zu einer Formel, in der jeweils andere Quantor nicht vorkommt.*

$\{\neg, \wedge, \exists\}$ *ist ein* vollständiges Junktoren-Quantoren-System*, d. h. jede $\mathcal{L}$-Formel ist logisch äquivalent zu einer Formel, in der keine anderen Junktoren und Quantoren vorkommen.*

Definition 2.4.4 **(a)** *Eine $\mathcal{L}$-Formel heißt $\mathcal{L}$-Tautologie, falls sie von der Form $F\,\frac{\varphi_1}{A_1}\ldots\frac{\varphi_n}{A_n}$ für $\mathcal{L}$-Formeln φ_i und eine aussagenlogische Tautologie $F(A_1,\ldots,A_n)$ ist.*

(b) *Folgende $\mathcal{L}$-Formeln heißen $\mathcal{L}$-Gleichheitsaxiome:*

[Reflexivität]	$\forall v_0\, v_0 \doteq v_0$
[Symmetrie]	$\forall v_0 \forall v_1\, (v_0 \doteq v_1 \leftrightarrow v_1 \doteq v_0)$
[Transitivität]	$\forall v_0\, \forall v_1 \forall v_2 ((\, v_0 \doteq v_1 \wedge v_1 \doteq v_2) \rightarrow v_0 \doteq v_2)$
[Kongruenz]	$\forall v_1 \ldots \forall v_{2n}\, ((v_1 \doteq v_{n+1} \wedge \cdots \wedge v_n \doteq v_{2n}) \rightarrow f v_1 \ldots v_n \doteq f v_{n+1} \ldots v_{2n})$
	$\forall v_1 \ldots \forall v_{2n}\, ((v_1 \doteq v_{n+1} \wedge \cdots \wedge v_n \doteq v_{2n}) \rightarrow (R v_1 \ldots v_n \leftrightarrow R v_{n+1} \ldots v_{2n}))$

für alle n-stelligen Funktionszeichen f und n-stelligen Relationszeichen R in $\mathcal{L}$ $(n \in \mathbb{N})$

(c) *Ein $\mathcal{L}$-$\exists$-Axiom ist eine Formel der Form $(\varphi\,\frac{\tau}{v_i} \rightarrow \exists v_i\, \varphi)$ für eine $\mathcal{L}$-Formel φ, einen $\mathcal{L}$-Term τ und eine Variable v_i, die frei für τ in φ ist.*

Das Präfix „$\mathcal{L}$-“ entfällt, wenn aus dem Kontext heraus klar ist, um welche Sprache $\mathcal{L}$ es sich handelt.

Lemma 2.4.5 *$\mathcal{L}$-Tautologien, $\mathcal{L}$-Gleichheitsaxiome, $\mathcal{L}$-$\exists$-Axiome sind allgemeingültig.*

Beweis: Für Tautologien folgt dies aus der uniformen Substitution in Lemma 2.3.6. Die Gleichheitsaxiome gelten trivialerweise in jedem Modell (da die Identität eine Kongruenzrelation ist). Die Allgemeingültigkeit der $\exists$-Axiome folgt aus dem Gesetz der $\exists$-Definition in Satz 2.4.2. □

Lemma 2.4.6 *Angenommen $(\varphi \rightarrow \psi)$ ist eine allgemeingültige $\mathcal{L}$-Formel.*

Modus Ponens[23]*: Wenn φ ebenfalls allgemeingültig ist, dann ist auch ψ allgemeingültig.*

$\exists$-Einführungsregel*: Wenn v_i nicht frei in ψ ist, dann ist auch $(\exists v_i\, \varphi \rightarrow \psi)$ allgemeingültig.*

Beweis: Die Gültigkeit des *modus ponens* ist klar. Angenommen also $(\varphi \rightarrow \psi)$ ist allgemeingültig und für $\mathfrak{M}$ und β gilt $\mathfrak{M} \vDash \exists v_i\, \varphi\,[\beta]$. Zu zeigen ist $\mathfrak{M} \vDash \psi\,[\beta]$. Es existiert also ein $m \in M$ mit $\mathfrak{M} \vDash \varphi\,[\beta\frac{m}{v_i}]$, und es gilt insbesondere $\mathfrak{M} \vDash (\varphi \rightarrow \psi)\,[\beta\frac{m}{v_i}]$, also auch $\mathfrak{M} \vDash \psi\,[\beta\frac{m}{v_i}]$. Da v_i nicht frei in ψ ist, folgt $\mathfrak{M} \vDash \psi\,[\beta]$. □

Folgende Umkehrung des *modus ponens* gilt nicht: Wenn aus der Allgemeingültigkeit von φ die Allgemeingültigkeit von ψ folgt, impliziert dies nicht die Allgemeingültigkeit von $(\varphi \rightarrow \psi)$. Insbesondere besagt die $\exists$-Einführungsregel nicht, dass $((\varphi \rightarrow \psi) \rightarrow (\exists v_i\, \varphi \rightarrow \psi))$ für beliebige φ, ψ allgemeingültig wäre (was im Allgemeinen nicht stimmt).

Am einfachsten sieht man es am folgenden Beispiel: Falls φ allgemeingültig ist, dann auch $\forall v_0\, \varphi$. Dagegen ist $(\varphi \rightarrow \forall v_0\, \varphi)$ im Allgemeinen nicht allgemeingültig, denn dies wäre gleichbedeutend mit der Allgemeingültigkeit von $\forall v_0\, (\varphi \rightarrow \forall v_0\, \varphi) \sim (\forall v_0 \neg\varphi \vee \forall v_0\, \varphi)$.

Dagegen ist $\exists v_0\, (\varphi \rightarrow \forall v_0\, \varphi) \sim (\neg\forall v_0\, \varphi \vee \forall v_0\, \varphi)$ allgemeingültig, was etwa im Beispiel von „Krivines Hut“ kontra-intuitiv wirken kann: *In jeder nicht-leeren Menschenmenge gibt es einen*

[23] Die aussagenlogische Schlussweise $A_0, (A_0 \rightarrow A_1) \vdash A_1$ hat den traditionellen Namen *modus ponens* und wird gleich im Beweis noch verwendet. Der *modus ponens* hier ist eine Folgerung daraus.

Menschen mit der Eigenschaft: Wenn dieser Mensch einen Hut trägt, dann tragen alle in der Menge einen Hut. (Wenn alle einen Hut tragen, kann jeder diese Rolle einnehmen; wenn nicht alle einen Hut tragen, wählt man jemanden, der keinen Hut trägt. Man darf die logische „wenn ... dann ...“-Implikation nicht mit einer kausalen Beziehung verwechseln.)

2.5 Der Gödel'sche Vollständigkeitssatz

Definition 2.5.1 (Hilbert-Kalkül) *Eine $\mathcal{L}$-Formel φ ist beweisbar (oder auch:* ableitbar*) im Hilbert-Kalkül $\mathbb{H}$, in Zeichen: $\vdash_{\mathbb{H}} \varphi$ (oder präziser: $\vdash^{\mathcal{L}}_{\mathbb{H}} \varphi$), falls es in $\mathbb{H}$ einen Beweis (eine* Ableitung*) von φ gibt, d. h. eine Folge von Formeln $\varphi_0, \ldots, \varphi_n$, wobei jedes φ_i ein Axiom des Kalküls ist oder sich durch eine Kalkülregel aus $\varphi_0, \ldots, \varphi_{i-1}$ ergibt und $\varphi_n = \varphi$ ist. Dabei sind*

Axiome von $\mathbb{H}$:	• *$\mathcal{L}$-Tautologien* • *$\mathcal{L}$-Gleichheitsaxiome* • *$\mathcal{L}$-$\exists$-Axiome* • *Formeln der Form* $(\forall v_i\, \varphi \leftrightarrow \neg\exists v_i \neg\varphi)$	Regeln von $\mathbb{H}$:	• *Modus Ponens* • *$\exists$-Einführungsregel*

Dabei bedeutet die $\exists$-Einführungsregel: Wenn $(\chi \to \psi)$ in der Folge $\varphi_0, \ldots, \varphi_{i-1}$ vorkommt und v_j nicht frei in ψ ist, dann darf $\varphi_i = (\exists v_j\, \chi \to \psi)$ sein. Entsprechend für den modus ponens*: Wenn $(\chi \to \psi)$ und χ in der Folge $\varphi_0, \ldots, \varphi_{i-1}$ vorkommen, dann darf $\varphi_i = \psi$ sein.*

Ziel dieses Abschnittes ist nun der Vollständigkeitssatz:

Satz 2.5.2 (Gödel'scher Vollständigkeitssatz für $\mathbb{H}$, 1929 [24])
Eine $\mathcal{L}$-Formel φ ist genau dann allgemeingültig, wenn sie in $\mathbb{H}$ beweisbar ist:

$$\vDash \varphi \iff \vdash_{\mathbb{H}} \varphi$$

Die Implikation „$\Leftarrow$“ im Vollständigkeitssatz heißt genauer *soundness* oder *Korrektheit* des Kalküls, die Implikation „$\Rightarrow$“ ist die eigentliche Vollständigkeit des Kalküls. Im Gegensatz zur Vollständigkeit ist die Korrektheit leicht zu sehen:

Beweis „$\Leftarrow$“: Folgt unmittelbar aus Lemma 2.4.5, den Dualitätsregeln für Quantoren und Lemma 2.4.6. □

Der Beweis der Vollständigkeitsrichtung „$\Rightarrow$“ wird den Rest des Abschnitts in Anspruch nehmen.

Bemerkung 2.5.3 *Genau genommen hängt die Definition der Beweisbarkeit in $\mathbb{H}$ von der gewählten Sprache $\mathcal{L}$ ab. Zunächst ist es denkbar, dass es für eine größere Sprache $\mathcal{L}' \supset \mathcal{L}$ einen Beweis einer $\mathcal{L}$-Formel φ geben könnte, in dem $\mathcal{L}'$-Formeln vorkommen, ohne dass es einen Beweis in $\mathcal{L}$, also nur mit $\mathcal{L}$-Formeln, gibt. Falls es in der Folge wichtig ist, werde ich daher die Sprache $\mathcal{L}$ beim Folgerungszeichen dazuschreiben: $\vdash^{\mathcal{L}}_{\mathbb{H}}$.*

Klar ist jedenfalls: Alles, was in $\mathcal{L}$ beweisbar ist, ist auch in $\mathcal{L}' \supset \mathcal{L}$ beweisbar. Der Vollständigkeitssatz wird auch zeigen, dass für $\mathcal{L}$-Formeln die Umkehrung gilt. Dann die Allgemeingültigkeit einer $\mathcal{L}$-Formel φ hängt nicht davon ab, ob man φ als $\mathcal{L}$-Formel oder als $\mathcal{L}'$-Formel ansieht:

Jede $\mathcal{L}'$-Struktur $\mathfrak{M}'$ wird auf eindeutige Weise zu einer $\mathcal{L}$-Struktur $\mathfrak{M} := \mathfrak{M}' \restriction_{\mathcal{L}}$, dem $\mathcal{L}$-Redukt *von $\mathfrak{M}'$, indem man die Interpretationen der Zeichen in $\mathcal{L}' \setminus \mathcal{L}$ „vergisst“. Umgekehrt kann man*

[24] Gödel hat 1929 die Vollständigkeit eines Hilbert-Kalküls gezeigt, der aber nicht genau wie $\mathbb{H}$ aussieht.

jede $\mathcal{L}$-Struktur $\mathfrak{M}$ zu (in der Regel vielen verschiedenen) $\mathcal{L}'$-Strukturen $\mathfrak{M}'$ expandieren, *indem man die zusätzlichen Zeichen beliebig interpretiert. Die Auswertung einer $\mathcal{L}$-Formel φ hängt von diesen zusätzlichen Zeichen nicht ab, d. h. $\mathfrak{M}' \vDash \varphi\,[\beta] \iff \mathfrak{M} \vDash \varphi\,[\beta]$.*

Zunächst ist es hilfreich, über einige zusätzliche Regeln zu verfügen. Klar ist folgender Spezialfall des *modus ponens*: Wenn φ beweisbar[25] ist und $(\varphi \to \varphi')$ eine Tautologie, dann ist auch φ' beweisbar (da alle Tautologien beweisbar sind). Jede „aussagenlogische Umformung“ einer beweisbaren Formel ist somit ebenfalls beweisbar. Falls etwa $\vdash_{\mathbb{H}} (\varphi \to \psi)$, dann ist auch die Kontraposition $(\neg\psi \to \neg\varphi)$ beweisbar, da $((\varphi \to \psi) \to (\neg\psi \to \neg\varphi))$ eine Tautologie ist. Meistens werde ich die Tautologie nicht explizit angeben, sondern zu einer Umformung nur sagen, sie ergebe sich *mit Aussagenlogik*.

Lemma 2.5.4
[verallgemeinerter Modus Ponens:]
Wenn alle $\varphi_1, \ldots, \varphi_n$ und $((\varphi_1 \wedge \cdots \wedge \varphi_n) \to \psi)$ beweisbar sind, dann ist ψ beweisbar.

[$\forall$**-Axiom**] *Wenn v_i frei für τ in φ ist, dann ist $(\forall v_i\,\varphi \to \varphi\frac{\tau}{v_i})$ beweisbar.*

[$\forall$**-Einführungsregel**]
Wenn $(\varphi \to \psi)$ beweisbar ist und v_i nicht frei in φ ist, dann ist $(\varphi \to \forall v_i\,\psi)$ beweisbar.

Beweis: (a) Dies gilt mit $(n+1)$-maligem *modus ponens*, da Folgendes eine Tautologie ist:

$$(((\varphi_1 \wedge \cdots \wedge \varphi_n) \to \psi) \to (\varphi_1 \to (\varphi_2 \to \cdots \to (\varphi_n \to \psi)\cdot)))$$

(b) Das $\exists$-Axiom für $\neg\varphi$, also $(\neg\varphi\frac{\tau}{v_i} \to \exists v_i \neg\varphi)$, ist beweisbar, und somit auch die aussagenlogische Umformung $(\neg\exists v_i \neg\varphi \to \varphi\frac{\tau}{v_i})$. Da $(\forall v_i\,\varphi \leftrightarrow \neg\exists v_i \neg\varphi)$ als Axiom beweisbar ist und

$$(((\neg\exists v_i \neg\varphi \to \varphi\frac{\tau}{v_i}) \wedge (\forall v_i\,\varphi \leftrightarrow \neg\exists v_i \neg\varphi)) \to (\forall v_i\,\varphi \to \varphi\frac{\tau}{v_i}))$$

eine Tautologie ist, folgt die Beweisbarkeit des $\forall$-Axioms.

(c) Wenn $(\varphi \to \psi)$ beweisbar ist, dann auch die Kontraposition $(\neg\psi \to \neg\varphi)$, mit $\exists$-Einführungsregel also auch $(\exists v_i \neg\psi \to \neg\varphi)$ und deren Kontraposition $(\varphi \to \neg\exists v_i \neg\psi)$. Wie unter (b) ersetzt man nun $\neg\exists v_i\neg$ durch $\forall v_i$. □

Alternativ könnte man den Hilbert-Kalkül um die $\forall$-Axiome und die $\forall$-Einführungsregel ergänzen und dafür auf die Axiome für die Dualität der Quantoren verzichten. Eine andere Möglichkeit besteht darin, mit einem vollständigen Junktoren-Quantoren-System wie $\{\neg, \wedge, \exists\}$ zu arbeiten. Dann ist es allerdings bequem, $\forall v_i$ als Abkürzung für $\neg\exists v_i\neg$ einzuführen.

Beispiel: Sei R ein zweistelliges Relationssymbol. Dann gilt $\vDash (\exists v_0 \forall v_1\, Rv_0v_1 \to \forall v_1 \exists v_0\, Rv_0v_1)$. Die folgende Formelfolge ist ein Beweis dafür in $\mathbb{H}$:

(1)	$(Rv_0v_1 \to \exists v_0\, Rv_0v_1)$	$\exists$-Axiom (da v_0 frei für v_0 in Rv_0v_1)
(2)	$(\forall v_1\, Rv_0v_1 \to Rv_0v_1)$	$\forall$-Axiom (da v_1 frei für v_1 in Rv_0v_1)
(3)	$(\forall v_1\, Rv_0v_1 \to \exists v_0\, Rv_0v_1)$	mit Aussagenlogik aus (1) und (2)
(4)	$(\exists v_0 \forall v_1\, Rv_0v_1 \to \exists v_0\, Rv_0v_1)$	mit $\exists$-Einführung aus (3) (da v_0 nicht frei in ...)
(5)	$(\exists v_0 \forall v_1\, Rv_0v_1 \to \forall v_1 \exists v_0\, Rv_0v_1)$	mit $\forall$-Einführung aus (4) (da v_1 nicht frei in ...)

[25] „Beweisbar“ steht ab sofort kurz für „beweisbar in $\mathbb{H}$“.

Wollte man versuchen, die Umkehrung $(\forall v_1 \exists v_0\, Rv_0v_1 \to \exists v_0 \forall v_1\, Rv_0v_1)$ zu beweisen, würde man gar keinen Ansatzpunkt finden (was allerdings keinen Beweis der Nicht-Beweisbarkeit liefert).

Lemma 2.5.5 *Sei $\varphi(v_1, \ldots, v_n)$ eine $\mathcal{L}$-Formel und $c_1, \ldots, c_n$ „neue“ Konstanten $\notin \mathcal{L}$. Dann:*

$$\vdash_{\mathbb{H}}^{\mathcal{L}} \varphi \quad \Longleftrightarrow \quad \vdash_{\mathbb{H}}^{\mathcal{L}} \forall v_1 \ldots \forall v_n\, \varphi \quad \Longleftrightarrow \quad \vdash_{\mathbb{H}}^{\mathcal{L} \cup \{c_1, \ldots, c_n\}} \varphi \frac{c_1}{v_1} \ldots \frac{c_n}{v_n}$$

Diese Äquivalenz gilt natürlich auch für Allgemeingültigkeit!

BEWEIS: Aus (*) $\vdash_{\mathbb{H}}^{\mathcal{L}} \varphi$ folgt mit Aussagenlogik: $\vdash_{\mathbb{H}}^{\mathcal{L}} (\top \to \varphi)$

mit n-maliger $\forall$-Einführung: $\vdash_{\mathbb{H}}^{\mathcal{L}} (\top \to \forall v_1 \ldots \forall v_n\, \varphi)$

mit *modus ponens*, da $\top$ Tautologie ist: $\vdash_{\mathbb{H}}^{\mathcal{L}} \forall v_1 \ldots \forall v_n\, \varphi$ (**)

also auch: $\vdash_{\mathbb{H}}^{\mathcal{L} \cup \{c_1, \ldots, c_n\}} \forall v_1 \ldots \forall v_n\, \varphi$

schließlich folgt mit dem $\forall$-Axiom $(\forall v_1 \ldots \forall v_n\, \varphi \to \varphi \frac{c_1}{v_1} \ldots \frac{c_n}{v_n})$

und *modus ponens*: $\vdash_{\mathbb{H}}^{\mathcal{L} \cup \{c_1, \ldots, c_n\}} \varphi \frac{c_1}{v_1} \ldots \frac{c_n}{v_n}$ (***)

Ersetzt man umgekehrt in einem $\mathcal{L} \cup \{c_1, \ldots, c_n\}$-Beweis von $\varphi \frac{c_1}{v_1} \ldots \frac{c_n}{v_n}$ jedes c_i durch eine Variable w_i, die in dem Beweis nicht vorkommt (z. B. $w_i = v_{i+m}$ für ein sehr großes m), dann erhält man daraus einen $\mathcal{L}$-Beweis von $\varphi \frac{w_1}{v_1} \ldots \frac{w_n}{v_n}$. Genau so, wie man oben auf dem Weg von (*) nach (***) die freien Vorkommen von v_i durch c_i ersetzt hat, kann man nun die Variablen w_i in v_i „umbenennen“, diese Mal allerdings ohne die Sprache $\mathcal{L}$ erweitern zu müssen. Also bekommt man aus (***) wieder (*) und hat die Äquivalenz in einem Ringschluss bewiesen. □

Folgerung 2.5.6

(a) *Man kann sich beim Beweis des Vollständigkeitssatzes auf Aussagen beschränken.*

(b) *Für $\mathcal{L}$-Aussagen φ gilt:* $\vdash_{\mathbb{H}}^{\mathcal{L}} \varphi \Longleftrightarrow \vdash_{\mathbb{H}}^{\mathcal{L} \cup \{c_1, \ldots, c_n\}} \varphi$.

Definition 2.5.7 *Sei $\mathcal{L}$ eine Sprache.*

(a) *Ein $\mathcal{L}$-Theorie ist eine Menge von $\mathcal{L}$-Aussagen (die häufig* Axiome *der Theorie heißen). Ein Modell einer Theorie T, $\mathfrak{M} \vDash T$, ist eine $\mathcal{L}$-Struktur $\mathfrak{M}$, in der alle Axiome von T gelten.*

(b) $T \vDash \varphi \;:\Longleftrightarrow$ *in jedem Modell von T gilt φ*

$T \vdash_{\mathbb{H}} \varphi :\Longleftrightarrow$ *es gibt $\psi_1, \ldots, \psi_n \in T$ mit* $\vdash_{\mathbb{H}} ((\psi_1 \wedge \cdots \wedge \psi_n) \to \varphi)$ [26]

(c) *T heißt $\mathbb{H}$-widerspruchsfrei, falls $T \nvdash_{\mathbb{H}} \bot$, und andernfalls $\mathbb{H}$-widersprüchlich.*

Lemma 2.5.8 *$T \vdash_{\mathbb{H}} \varphi \Longleftrightarrow T \cup \{\neg\varphi\}$ ist $\mathbb{H}$-widersprüchlich.*

BEWEIS: „⇒“: Nach Annahme ist $\vdash_{\mathbb{H}} ((\psi_1 \wedge \cdots \wedge \psi_n) \to \varphi)$ für $\psi_i \in T$. Man bekommt dann $\vdash_{\mathbb{H}} ((\psi_1 \wedge \cdots \wedge \psi_n \wedge \neg\varphi) \to \bot)$, da $((A_0 \to A_1) \to ((A_0 \wedge \neg A_1) \to \bot))$ eine Tautologie ist.

„⇐“: Nach Annahme ist $\vdash_{\mathbb{H}} ((\psi_1 \wedge \cdots \wedge \psi_n) \to \bot)$ für $\psi_i \in T \cup \{\neg\varphi\}$. Da $(\bot \to \varphi)$ eine Tautologie ist, gilt also auch $\vdash_{\mathbb{H}} ((\psi_1 \wedge \cdots \wedge \psi_n) \to \varphi)$. Falls $\neg\varphi$ nicht vorkommt unter den ψ_i, ist man fertig. Andernfalls kann man annehmen, dass nur $\psi_n = \neg\varphi$ ist. Dann sieht man mit Aussagenlogik, dass $((\psi_1 \wedge \cdots \wedge \psi_{n-1} \wedge \neg\varphi) \to \varphi) \sim ((\psi_1 \wedge \cdots \wedge \psi_{n-1}) \to \varphi)$. □

[26] Eine alternative und eigentlich naheliegendere Definition für $T \vdash_{\mathbb{H}} \varphi$ wäre zu sagen, dass φ in $\mathbb{H}$ beweisbar ist, wobei die Axiome von T als zusätzliche Axiome zum Kalkül zugelassen sind. Um mit dieser Version arbeiten zu können, muss man aber erst die hier gegebene Variante herleiten, d. h. $T \cup \{\psi\} \vdash_{\mathbb{H}} \varphi \Longleftrightarrow T \vdash_{\mathbb{H}} (\psi \to \varphi)$ zeigen. Dies formal exakt zu tun, ist etwas aufwendig.

Satz 2.5.9 (Vollständigkeitssatz für Theorien)
Eine $\mathcal{L}$-Theorie T ist genau dann $\mathbb{H}$-widerspruchsfrei, wenn sie ein Modell hat.

Beweis des Vollständigkeitssatzes 2.5.2 aus Satz 2.5.9:
Sei φ eine allgemeingültige $\mathcal{L}$-Aussage und betrachte die $\mathcal{L}$-Theorie $T := \{\neg\varphi\}$. Nach Voraussetzung hat T kein Modell und ist somit nach Satz 2.5.9 nicht $\mathbb{H}$-widerspruchsfrei, d. h. $T \vdash_{\mathbb{H}} \bot$. Dies bedeutet aber nach Definition $\vdash_{\mathbb{H}} (\neg\varphi \to \bot)$ bzw. mit Aussagenlogik $\vdash_{\mathbb{H}} \varphi$. □

Eine Richtung des Beweises von Satz 2.5.9 ist einfach:

Beweis „⇐“: Wenn $\mathfrak{M} \vDash T$ und $T \vdash_{\mathbb{H}} \varphi$, dann gilt auch $\mathfrak{M} \vDash \varphi$, denn nach Annahme ist $((\psi_1 \wedge \dots \wedge \psi_n) \to \varphi)$ für geeignete $\psi_i \in T$ beweisbar, also allgemeingültig, und gilt somit in $\mathfrak{M}$. Wenn T $\mathbb{H}$-widersprüchlich wäre, also $T \vdash_{\mathbb{H}} \bot$, müsste daher auch $\mathfrak{M} \vDash \bot$ gelten, was aber per Definition nie der Fall ist. □

Zu zeigen ist also noch:

Wenn T $\mathbb{H}$-widerspruchsfrei ist, dann besitzt T ein Modell.

Der hier präsentierte Beweis mittels der Methode von Henkin besteht in der Grundidee darin, die Sprache $\mathcal{L}$ durch Konstanten so anzureichern, dass die Konstanten bereits das Universum des Modells bilden: Damit dies funktioniert, müsste es, falls $T \vdash_{\mathbb{H}} \exists v_i\, \varphi$, eine Konstante c_φ geben mit $T \vdash_{\mathbb{H}} \varphi\frac{c_\varphi}{v_i}$. Solche Konstanten werden nun nach und nach hinzugenommen, dabei muss die Sprache und die Theorie sukzessive erweitert werden.

Definition 2.5.10 *Für jede $\mathcal{L}$-Formel $\varphi(v_i)$ sei $c_\varphi \notin \mathcal{L}$ eine „neue“ Konstante.*[27] *Dann sei*

$$\mathcal{L}^* := \mathcal{L} \cup \big\{c_\varphi \;\big|\; \varphi(v_i)\ \mathcal{L}\text{-Formel}\big\} \quad \text{und} \quad T^* := T \cup \Big\{(\exists v_i\, \varphi \to \varphi\frac{c_\varphi}{v_i}) \;\Big|\; \varphi(v_i)\ \mathcal{L}\text{-Formel}\Big\}.$$

Die Konstanten c_φ heißen Henkin-Konstanten, *die Formeln $(\exists v_i\, \varphi \to \varphi\frac{c_\varphi}{v_i})$* Henkin-Axiome. *Die* Henkin-Theorie *von T ist schließlich die $\mathcal{L}^{\mathsf{h}}$-Theorie T^{h}, wobei*

$$\mathcal{L}^{\mathsf{h}} := \mathcal{L} \cup \mathcal{L}^* \cup \mathcal{L}^{**} \cup \dots \quad \text{und} \quad T^{\mathsf{h}} := T \cup T^* \cup T^{**} \cup \dots$$

Lemma 2.5.11
Die Henkin-Theorie einer $\mathbb{H}$-widerspruchsfreien Theorie ist wieder $\mathbb{H}$-widerspruchsfrei.

Beweis: Angenommen T_i ist eine $\mathbb{H}$-widerspruchsfreie $\mathcal{L}_i$-Theorie mit $T_i \subseteq T_{i+1}$, dann ist auch $T_\omega := \bigcup_{i\in\mathbb{N}} T_i$ $\mathbb{H}$-widerspruchsfrei: Denn wäre $\vdash_{\mathbb{H}} ((\psi_1 \wedge \dots \wedge \psi_n) \to \bot)$ für $\psi_i \in T_\omega$, dann würden die endlich vielen ψ_i bereits aus einen T_{n_0} stammen und die endlich vielen in dem Beweis vorkommenden Formeln wären $\mathcal{L}_{n_1}$-Formeln für ein $n_1 \geqslant n_0$ und somit wäre schon T_{n_1} $\mathbb{H}$-widersprüchlich.

Es reicht daher zu zeigen, dass die Hinzunahme eines einzelnen Henkin-Axioms eine Theorie nicht $\mathbb{H}$-widersprüchlich machen kann: Angenommen also, $T \cup \{(\exists v_i\, \varphi \to \varphi\frac{c_\varphi}{v_i})\}$ ist eine $\mathbb{H}$-widersprüchliche $\mathcal{L} \cup \{c_\varphi\}$-Theorie für ein $\varphi(v_i)$ mit $T \vdash^{\mathcal{L}}_{\mathbb{H}} \exists v_i\, \varphi$ (*).

$$\text{Mit Lemma 2.5.8 gilt dann: } T \vdash^{\mathcal{L}\cup\{c_\varphi\}}_{\mathbb{H}} \neg(\exists v_i\, \varphi \to \varphi\frac{c_\varphi}{v_i})$$

$$\text{und Aussagenlogik zeigt daraus: } T \vdash^{\mathcal{L}\cup\{c_\varphi\}}_{\mathbb{H}} \neg\varphi\frac{c_\varphi}{v_i} \quad \text{d. h.} \quad T \vdash^{\mathcal{L}\cup\{c_\varphi\}}_{\mathbb{H}} (\varphi\frac{c_\varphi}{v_i} \to \bot).$$

[27] d. h. insbesondere $c_\varphi \neq c_\psi$ für $\varphi \neq \psi$

Mit Hilfe von Lemma 2.5.5 kann man die Konstante eliminieren und erhält $T \vdash^{\mathcal{L}}_{\mathbb{H}} (\varphi \to \bot)$, d. h. $\vdash^{\mathcal{L}}_{\mathbb{H}} ((\psi_1 \wedge \dots \psi_n) \to (\varphi \to \bot))$ für geeignete $\psi_i \in T$. Dies kann man aussagenlogisch umschreiben in $\vdash^{\mathcal{L}}_{\mathbb{H}} (\varphi \to ((\psi_1 \wedge \dots \psi_n) \to \bot))$, worauf man die $\exists$-Einführungsregel anwenden kann. Führt man die Umformungen wieder rückwärts aus, erhält man schließlich $T \vdash^{\mathcal{L}}_{\mathbb{H}} (\exists v_i\, \varphi \to \bot)$ und mit (*) und *modus ponens* dann $T \vdash^{\mathcal{L}}_{\mathbb{H}} \bot$. Somit ist T bereits $\mathbb{H}$-widersprüchlich. □

Definition 2.5.12 *Eine $\mathcal{L}$-Theorie heißt vollständig, falls sie widerspruchsfrei ist und für alle $\mathcal{L}$-Aussagen φ gilt $T \vdash_{\mathbb{H}} \varphi$ oder $T \vdash_{\mathbb{H}} \neg\varphi$.*

Lemma 2.5.13 *Jede $\mathbb{H}$-widerspruchsfreie $\mathcal{L}$-Theorie kann zu einer vollständigen $\mathcal{L}$-Theorie erweitert werden.*

Beweis: Man zählt alle $\mathcal{L}$-Aussagen durch $\{\varphi_i \mid i \in \mathbb{N}\}$ auf[28] und definiert induktiv $T_0 := T$ und

$$T_{i+1} := \begin{cases} T_i \cup \{\varphi_i\} & \text{falls } T_i \vdash_{\mathbb{H}} \varphi_i \\ T_i \cup \{\neg\varphi_i\} & \text{andernfalls} \end{cases}$$

Induktiv zeigt man nun, dass alle T_i $\mathbb{H}$-widerspruchsfrei sind:

1. Fall: Ist T_i $\mathbb{H}$-widerspruchsfrei und $T_i \vdash_{\mathbb{H}} \varphi_i$, dann ist auch $T_i \cup \{\varphi_i\}$ $\mathbb{H}$-widerspruchsfrei, denn andernfalls würde mit Lemma 2.5.8 $T_i \vdash_{\mathbb{H}} \neg\varphi_i$ gelten, also auch $T_i \vdash_{\mathbb{H}} (\varphi \wedge \neg\varphi_i) \sim \bot$ und T_i wäre bereits $\mathbb{H}$-widersprüchlich.

2. Fall mit $T_i \vdash_{\mathbb{H}} \neg\varphi_i$ geht analog.

3. Fall: T_i $\mathbb{H}$-widerspruchsfrei und weder $T_i \vdash_{\mathbb{H}} \varphi_i$, noch $T_i \vdash_{\mathbb{H}} \neg\varphi_i$. Dann ist z. B. $T_i \cup \{\neg\varphi_i\}$ $\mathbb{H}$-widerspruchsfrei, denn andernfalls würde wieder nach Lemma 2.5.8 $T_i \vdash_{\mathbb{H}} \neg\neg\varphi_i$, d. h. $T_i \vdash_{\mathbb{H}} \varphi_i$.

Schließlich ist $T_\infty := \bigcup_{i\in\mathbb{N}} T_i$ als Vereinigung $\mathbb{H}$-widerspruchsfreier Theorie selbst $\mathbb{H}$-widerspruchsfrei, und T_∞ ist vollständig, da per Konstruktion sogar für jede Formel φ entweder $\varphi \in T$ oder $\neg\varphi \in T$ gilt. □

Ausgehend von einer $\mathbb{H}$-widerspruchsfreien $\mathcal{L}$-Theorie T kann man also eine vollständige, $\mathbb{H}$-widerspruchsfreie $\mathcal{L}^{\mathsf{h}}$-Theorie $T^+ := T^{\mathsf{h}}_\infty$ konstruieren, die T erweitert und alle Henkin-Axiome erfüllt. Insbesondere ist das $\mathcal{L}$-Redukt eines Modells von T^+ ein Modell von T. Es reicht also zum Beweis des Vollständigkeitssatzes ein Modell $\mathfrak{M}^+$ von T^+ zu konstruieren.

Sei $\mathcal{T}$ die Menge der Terme in $\mathcal{L}^{\mathsf{h}}$.

Lemma 2.5.14 $\tau_0 \approx \tau_1 :\iff T^+ \vdash_{\mathbb{H}} \tau_0 \doteq \tau_1$ *definiert eine Äquivalenzrelation auf $\mathcal{T}$.*

Beweis: Dies folgt aus den Gleichheitsaxiomen. Ich beweise nur beispielhaft die Transitivität:

Sei $\tau_0 \approx \tau_1$ und $\tau_1 \approx \tau_2$, d. h $T^+ \vdash_{\mathbb{H}} \tau_0 \doteq \tau_1$ und $T^+ \vdash_{\mathbb{H}} \tau_1 \doteq \tau_2$. Dann folgt mit

Gleichheitsaxiom:	$T^+ \vdash_{\mathbb{H}} \forall v_0 \forall v_1 \forall v_2 ((v_0 \doteq v_1 \wedge v_1 \doteq v_2) \to v_0 \doteq v_2)$
$\forall$-*Axiom*:	$T^+ \vdash_{\mathbb{H}} \big(\forall v_0 \forall v_1 \forall v_2 ((v_0 \doteq v_1 \wedge v_1 \doteq v_2) \to v_0 \doteq v_2) \to ((\tau_0 \doteq \tau_1 \wedge \tau_1 \doteq \tau_2) \to \tau_0 \doteq \tau_2)\big)$
Modus Ponens:	$T^+ \vdash_{\mathbb{H}} ((\tau_0 \doteq \tau_1 \wedge \tau_1 \doteq \tau_2) \to \tau_0 \doteq \tau_2)$
verallgemeinertem Modus Ponens:	$T^+ \vdash_{\mathbb{H}} \tau_0 \doteq \tau_2$

□

Konstruktion von $\mathfrak{M}^+$:

[28] Das geht, wenn die Sprache $\mathcal{L}$ endlich oder abzählbar unendlich ist. Der Satz gilt auch für überabzählbare Sprachen, dann braucht man wieder etwas Mathematik überabzählbarer Mengen.

Das Universum von $\mathfrak{M}^+$ ist $\mathcal{T}/_\approx = \{\tau/_\approx \mid \tau \in \mathcal{T}\}$, also die Menge der $\approx$-Äquivalenzklassen von $\mathcal{T}$. Die Interpretation der Zeichen aus $\mathcal{L}^{\mathsf{h}}$ in $\mathfrak{M}^+$ geschieht wie folgt:

für Konstanten c:		$c^{\mathfrak{M}^+}$	$:=$	$c/_\approx$
für n-stellige Funktionszeichen f:		$f^{\mathfrak{M}^+}(\tau_1/_\approx, \dots, \tau_n/_\approx)$	$:=$	$(f\tau_1 \dots \tau_n)/_\approx$
für n-stellige Relationszeichen R:		$(\tau_1/_\approx, \dots, \tau_n/_\approx) \in R^{\mathfrak{M}^+}$	$:\iff$	$T^+ \vdash_{\mathbb{H}} R\tau_1 \dots \tau_n$
			$\iff$	$R\tau_1 \dots \tau_n \in T^+$

Lemma 2.5.15 *Die Interpretationen der Zeichen in $\mathfrak{M}^+$ sind wohldefiniert.*

Beweis: Sei $\tau_i/_\approx = \tau_i'/_\approx$, d.h. $\tau_i \doteq \tau_i' \in T^+$ für $i = 1, \dots, n$. Zu zeigen ist also

$$f^{\mathfrak{M}^+}(\tau_1/_\approx, \dots, \tau_n/_\approx) \quad = \quad f^{\mathfrak{M}^+}(\tau_1'/_\approx, \dots, \tau_n'/_\approx) \tag{1}$$

$$\text{und} \quad (\tau_1/_\approx, \dots, \tau_n/_\approx) \in R^{\mathfrak{M}^+} \iff (\tau_1'/_\approx, \dots, \tau_n'/_\approx) \in R^{\mathfrak{M}^+} \tag{2}$$

Wie im Beweis von Lemma 2.5.14 zeigt man, dass folgende Instanz des Gleichheitsaxiom aus T^+ folgt:

$$((\tau_1 \doteq \tau_1' \wedge \dots \wedge \tau_n \doteq \tau_n') \to f\tau_1 \dots \tau_n \doteq f\tau_1' \dots \tau_n')$$

Mit dem verallgemeinerten *modus ponens* erhält man daraus $f\tau_1 \dots \tau_n \doteq f\tau_1' \dots \tau_n' \in T^+$, also $f\tau_1 \dots \tau_n \approx f\tau_1' \dots \tau_n'$, was gerade (1) ist. (2) zeigt man ganz analog. □

Satz 2.5.16 $\mathfrak{M}^+ \vDash T^+$.

Beweis: Per Induktion über den Aufbau der $\mathcal{L}^{\mathsf{h}}$-Formeln zeigt man für Aussagen φ:

$$\mathfrak{M} \vDash \varphi \iff T^+ \vdash_{\mathbb{H}} \varphi \ (\iff \varphi \in T^+)$$

(Dass $\mathfrak{M}^+$ Modell von T^+ ist, bedeutet zunächst nur die Implikation „$\Leftarrow$“. Da T^+ vollständig ist, folgt aber auch die andere Richtung – wie unten beim Induktionsschritt für die Negation.)

Reduktion auf Aussagen: Im Falle einer Henkin-Theorie kann man Formeln mit freien Variablen wie folgt auf Aussagen zurückführen. Für jedes Element $m = \tau/_\approx \in M^+$ liegt $\exists v_0\, \tau \doteq v_0$ in T^+ (wegen der Vollständigkeit von T^+). Für die Henkin-Konstante $c_m := c_{\tau \doteq v_0}$ gilt dann $T^+ \vdash_{\mathbb{H}} \tau \doteq c_m$, also $m = c_m/_\approx$. Jede Äquivalenzklasse wird also von einer Konstanten repräsentiert. Für eine Belegung β, Terme $\tau(v_1, \dots, v_n)$ und Formeln $\varphi(v_1, \dots, v_n)$ sei nun

$$\tau\frac{c}{\beta} := \tau\frac{c_{\beta(v_1)}}{v_1}\dots\frac{c_{\beta(v_1)}}{v_n} \quad \text{und} \quad \varphi\frac{c}{\beta} := \varphi\frac{c_{\beta(v_1)}}{v_1}\dots\frac{c_{\beta(v_1)}}{v_n}.$$

$\tau\frac{c}{\beta}$ ist dann ein geschlossener Term, der in folgendem Sinn gleichwertig mit τ ist: Da $v_i^{\mathfrak{M}^+}[\beta] = \beta(v_i) = c_{\beta(v_i)}/_\approx$, folgt mit Lemma 2.5.15 induktiv $\tau^{\mathfrak{M}^+}[\beta] = (\tau\frac{c}{\beta})/_\approx$. Entsprechend ist $\varphi\frac{c}{\beta}$ eine Aussage, und per Induktion über den Aufbau der Formeln zeigt man nun:

$$\mathfrak{M} \vDash \varphi[\beta] \iff T^+ \vdash_{\mathbb{H}} \varphi\tfrac{c}{\beta} \ (\iff \varphi\tfrac{c}{\beta} \in T^+)$$

Es reicht, dabei das vollständige Junktoren-Quantoren-System $\{\neg, \wedge, \exists\}$ zu betrachten.

(1) Für *atomare Formeln* gilt (auch im Spezialfall $n = 2$ und „$R = \doteq$“):

$$\begin{aligned}
\mathfrak{M}^+ \vDash R\tau_1 \dots \tau_n\,[\beta] &\iff (\tau_1^{\mathfrak{M}^+}[\beta], \dots, \tau_n^{\mathfrak{M}^+}[\beta]) \in R^{\mathfrak{M}^+} \\
&\iff ((\tau_1\tfrac{c}{\beta})/_\approx, \dots, (\tau_n\tfrac{c}{\beta})/_\approx) \in R^{\mathfrak{M}^+} \\
&\iff T^+ \vdash_{\mathbb{H}} R\tau_1\tfrac{c}{\beta} \dots \tau_n\tfrac{c}{\beta} = (R\tau_1 \dots \tau_n)\tfrac{c}{\beta}
\end{aligned}$$

(2) *Negation:* Wegen der Vollständigkeit von T^+ gilt

$$T^+ \vdash_{\mathbb{H}} \neg\varphi\tfrac{c}{\beta} \iff T^+ \nvdash_{\mathbb{H}} \varphi\tfrac{c}{\beta} \iff \mathfrak{M}^+ \nvDash \varphi\,[\beta] \iff \mathfrak{M}^+ \vDash \neg\varphi\,[\beta]$$

(3) *Konjunktion:*

$$\begin{aligned}\mathfrak{M}^+ \vDash (\varphi \wedge \psi)\,[\beta] &\iff \mathfrak{M}^+ \vDash \varphi\,[\beta] \text{ und } \mathfrak{M}^+ \vDash \psi\,[\beta] \\ &\iff T^+ \vdash_{\mathbb{H}} \varphi\tfrac{c}{\beta} \text{ und } T^+ \vdash_{\mathbb{H}} \psi\tfrac{c}{\beta} \iff T^+ \vdash_{\mathbb{H}} (\varphi \wedge \psi)\tfrac{c}{\beta}\end{aligned}$$

(Bei der letzten Äquivalenz gilt „⇒“ mit dem verallgemeinerten *modus ponens* und „⇐“ mit normalem *modus ponens* auf die Tautologie $((\varphi \wedge \psi) \to \psi)$ etc.)

(4) *Existenzquantifikation:*

Sei $\exists v_i\,\psi\frac{c}{\beta} \in T^+$. Mit dem Henkin-Axiom $(\exists v_i\,\psi\frac{c}{\beta} \to \psi\frac{c_\psi}{v_i})$ und *modus ponens* erhält man $\psi\frac{c_\psi}{v_i}\frac{c}{\beta} \in T^+$, per Induktion also $\mathfrak{M}^+ \vDash \psi\frac{c_\psi}{v_i}\,[\beta]$ und damit $\mathfrak{M}^+ \vDash \exists v_i\,\psi\,[\beta]$.

Sei umgekehrt $\mathfrak{M}^+ \vDash \exists v_i\,\psi\,[\beta]$. Dann gibt es ein $m \in M$ mit $\mathfrak{M}^+ \vDash \psi\,[\beta\frac{m}{v_i}]$ bzw. $\mathfrak{M}^+ \vDash \psi\frac{c_m}{v_i}\,[\beta]$. Per Induktion gilt dann $\psi\frac{c}{\beta\frac{m}{v_i}} = \psi\frac{c_m}{v_i}\frac{c}{\beta} \in T^+$. Mit dem entsprechenden ∃-Axiom und *modus ponens* bekommt man dann $\exists v_i\,\psi\frac{c}{\beta} \in T^+$. □

Folgerung 2.5.17 (Kompaktheits- oder Endlichkeitssatz für die Prädikatenlogik)
Eine unendliche Menge von $\mathcal{L}$-Formeln ist genau dann erfüllbar (d. h. hat ein Modell), wenn sie endlich erfüllbar ist (d. h. wenn jede endliche Teilmenge ein Modell hat).

Beweis: Wenn eine unendliche Formelmenge T ein Modell hat, ist dies natürlich auch ein Modell jeder endlichen Teilmenge. Wenn T kein Modell hat, ist T nach dem Vollständigkeitssatz $\mathbb{H}$-widersprüchlich, beweist also $\bot$. In einem Beweis kommen aber nur endlich viele Formeln aus T vor, d. h. die endliche Teilmenge, die von diesen Formeln gebildet wird, ist bereits $\mathbb{H}$-widersprüchlich und hat demnach kein Modell. □

Bemerkungen: Es gibt sehr viele Varianten von Kalkülen, für die man den Vollständigkeitssatz beweisen kann. Wichtig in Hinblick auf die Folgerungen ist jeweils, dass Axiome maschinell überprüft und Regeln maschinell angewendet werden können.

Häufig unterscheidet man zwei Arten von Kalkülen: Axiomen- oder Hilbert-Kalküle mit vielen Axiomen und wenigen Regeln (meist nur *modus ponens*) und Regel- oder Gentzen-Kalküle mit wenigen Axiomen und vielen Regeln. *Sequenzenkalküle* und *Kalküle des natürlichen Schließens* sind zwei bekanntere Arten von Kalkülen.

Generell gibt es zwei Vorgehensweisen, wie man „eine Logik“ begründen kann. Eine Möglichkeit ist die *syntaktische* Vorgehensweise über einen Kalkül, also über ein Regelsystem, mit welchem man logische Gesetze ableiten kann. Dafür benutzt man gerne das „Folgerungszeichen“ $\vdash$. Die andere Möglichkeit ist die *semantische* Vorgehensweise über Auswertungen von Formeln in Modellen. Dafür benutzt man dann vorzugsweise das „Modellzeichen“ $\vDash$ mit dem Doppelstrich. Wenn beide Vorgehensweisen möglich sind und man über einen entsprechenden Vollständigkeitssatz die Gleichwertigkeit nachgewiesen hat, kann den Unterschied der beiden Zeichen freilich wieder vergessen.

Die Definition aussagenlogischer Tautologien und Folgerungen über Wahrheitstabellen ist eine semantische Vorgehensweise. Daher hätte man in der Aussagenlogik eigentlich eher $\vDash$ als

$\vdash$ schreiben sollen. Allerdings ist in der Aussagenlogik der Unterschied beider Vorgehensweisen nicht so relevant, da im Gegensatz zur Prädikatenlogik die Semantik bereits ein Entscheidungsverfahren liefert. Daher wird meist doch das traditionellere Zeichen $\vdash$ verwendet. Folgerung 1.3.5 (a) ist übrigens ein Vollständigkeitssatz für ein aussagenlogisches Regelsystem. Da die Prädikatenlogik hier als Erweiterung der Aussagenlogik konzipiert ist, kann man natürlich auch aus $\mathbb{H}$ einen vollständigen aussagenlogischen Kalkül gewinnen.

2.6 Der Satz von Herbrand und Unifikation

Definition 2.6.1 *Eine $\mathcal{L}$-Formel ist in* pränexer Normalform*, falls sie die Form*

$$\mathsf{Q}_1 v_{i_1} \ \ldots \ \mathsf{Q}_n v_{i_n} \ \psi$$

hat mit $\mathsf{Q}_i \in \{\exists, \forall\}$ und quantorenfreiem ψ (d. h. ψ enthält keine Quantoren).

Lemma 2.6.2 *Jede $\mathcal{L}$-Formel ist zu einer $\mathcal{L}$-Formel in pränexer Normalform äquivalent.*

Beweis: Atomare $\mathcal{L}$-Formeln sind quantorenfrei und damit in pränexer Normalform. Es reicht nun zu zeigen, dass Zusammensetzungen von Formeln in pränexer Normalform mit dem vollständigen Junktoren-Quantoren-System $\{\neg, \wedge, \vee, \exists, \forall\}$ wieder in pränexe Normalform gebracht werden können.

- $\exists v_i \varphi$ und $\forall v_i \varphi$ sind bereits in pränexer Normalform, wenn φ es ist.
- $\neg\, \mathsf{Q}_1 v_{i_1} \ \ldots \ \mathsf{Q}_n v_{i_n} \ \psi \ \sim \ \overline{\mathsf{Q}_1} v_{i_1} \ \ldots \ \overline{\mathsf{Q}_n} v_{i_n} \ \neg \psi$ mit $\overline{\exists} = \forall$ und $\overline{\forall} = \exists$.
- Seien $\mathsf{Q} v_j \, \varphi$ und χ in pränexer Normalform mit $\mathsf{Q} \in \{\exists, \forall\}$ und v_k eine Variable, die in χ nicht frei vorkommt. Dann gilt

$$(\mathsf{Q} v_j \, \varphi \wedge \chi) \ \sim \ (\mathsf{Q} v_k \, \varphi \frac{v_k}{v_j} \wedge \chi) \ \sim \ \mathsf{Q} v_k (\varphi \frac{v_k}{v_j} \wedge \chi)$$
$$(\mathsf{Q} v_j \, \varphi \vee \chi) \ \sim \ (\mathsf{Q} v_k \, \varphi \frac{v_k}{v_j} \vee \chi) \ \sim \ \mathsf{Q} v_k (\varphi \frac{v_k}{v_j} \vee \chi)$$

  Mit der Kommutativität der Kon- bzw. Disjunktion gelten die analogen Regeln, wenn der Quantor vor χ steht. Auf diese Weise kann man sukzessive alle Quantoren vor die Klammer ziehen. □

Bemerkung: Für den Beweis würde es ausreichen, z. B. mit dem vollständigen Junktoren-Quantoren-System $\{\neg, \wedge, \exists\}$ zu arbeiten. Die hier vorgestellte Variante ist in der Praxis gut verwendbar; für die Junktoren $\rightarrow$ und $\leftrightarrow$ gelten schwerer zu merkende Regeln.

Beispiel:
$$\begin{aligned}
(\exists v_0 \forall v_1 \, R v_0 v_1 \rightarrow \forall v_1 \exists v_0 \, R v_0 v_1) \ &\sim \ (\neg \exists v_0 \forall v_1 \, R v_0 v_1 \vee \forall v_1 \exists v_0 \, R v_0 v_1) \\
&\sim \ (\forall v_0 \exists v_1 \neg R v_0 v_1 \vee \forall v_1 \exists v_0 \, R v_0 v_1) \\
&\sim \ (\forall v_0 \exists v_1 \neg R v_0 v_1 \vee \forall v_2 \exists v_3 \, R v_3 v_2) \\
&\sim \ \forall v_0 \exists v_1 \forall v_2 \exists v_3 \, (\neg R v_0 v_1 \vee R v_3 v_2) \\
\text{aber auch:} \quad &\sim \ \forall v_2 \forall v_0 \exists v_1 \exists v_3 \, (\neg R v_0 v_1 \vee R v_3 v_2)
\end{aligned}$$

Die pränexe Normalform ist also nicht eindeutig. Abgesehen von Offensichtlichem (äquivalente Darstellung des quantorenfreien Teils; zusätzliche unnötige Quantoren; Umbenennung gebundener Variablen) kann die Reihenfolge mancher Quantoren vertauscht werden, anderer nicht.

Definition 2.6.3 *Eine $\mathcal{L}$-Formel in pränexer Normalform heißt universell, wenn in ihr keine Existenzquantoren vorkommen, und existenziell, wenn in ihr keine Allquantoren vorkommen.*

Quantorenfreie Formeln sind also sowohl universell als auch existenziell.

Satz 2.6.4 (a) *Eine $\mathcal{L}$-Aussage φ in pränexer Normalform kann auf algorithmische Weise zu einer erfüllbarkeitsäquivalenten universellen $\mathcal{L}^*$-Aussage φ^* gemacht werden, wobei $\mathcal{L}^* \supseteq \mathcal{L}$ zusätzliche Funktionszeichen (inklusive Konstanten) enthält.*

(b) *Eine $\mathcal{L}$-Aussage φ in pränexer Normalform kann auf algorithmische Weise zu einer existenziellen $\mathcal{L}^*$-Aussage φ_* gemacht werden, die genau dann allgemeingültig ist, wenn φ es ist.* [29]

Die zusätzlichen Funktionszeichen heißen Skolem-Funktionen, φ^* heißt Skolem-Normalform oder Skolemisierung von φ und φ_* heißt Herbrand-Normalform von φ.

Bemerkung: Trivialerweise gibt es zu jeder $\mathcal{L}$-Aussage φ sogar eine quantorenfreie erfüllbarkeitsäquivalente Ausssage, nämlich $\top$, falls φ erfüllbar ist, und $\bot$ andernfalls. Wichtig ist in diesem Satz daher, dass man die erfüllbarkeitsäquivalente Ausssage algorithmisch finden kann, also ohne vorher entscheiden zu müssen, ob φ erfüllbar ist oder nicht (was, wie in Kapitel 3 gezeigt wird, im Allgemeinen nicht geht).

Beweis: (a) Sei φ in pränexer Normalform. Nach Umbenennung der Variablen hat die Formel folgende Form:

$$\forall v_1 \dots \forall v_k \, \exists v_{k+1} \, \chi,$$

wobei χ eine $\mathcal{L}$-Formel ist, also weitere Quantoren enthalten kann, und $k \in \mathbb{N}$, d. h. $k = 0$ ist zugelassen. Nun führt man ein neues k-stelliges Funktionszeichen f ein (im Falle von $k = 0$ eine neue Konstante) und ersetzt die Formel durch

$$\forall v_1 \dots \forall v_k \, \chi \frac{f v_1 \dots v_k}{v_{k+1}}$$

Auf diese Weise ersetzt man von außen nach innen (d. h. von links nach rechts) alle vorkommenden Existenzquantoren und erhält zum Schluss φ^*.

Falls φ erfüllbar ist, definiert man in einem Modell $\mathfrak{M}$ von φ die neuen Funktionen $f^{\mathfrak{M}}$ folgendermaßen: Für $(m_1, \dots, m_k) \in M^k$ betrachtet man eine Belegung β mit $\beta(v_i) = m_i$ und wählt für $f^{\mathfrak{M}}(m_1, \dots, m_k)$ ein Element, welches den entsprechenden Existenzquantor (oben: $\exists v_{k+1}$) unter β erfüllt. Dadurch wird $\mathfrak{M}$ zu einem Modell $\mathfrak{M}^*$ von φ^* expandiert und folglich ist φ^* erfüllbar.

Ist umgekehrt $\mathfrak{M}^*$ ein Modell von φ^* und β eine Belegung, dann ist $f^{\mathfrak{M}^*}(\beta(v_1), \dots, \beta(v_k))$ ein Element, welches den entsprechenden Existenzquantor in φ unter β erfüllt.

(b) φ ist genau dann allgemeingültig, wenn $\neg\varphi$ nicht erfüllbar ist, also nach (a) genau dann, wenn $(\neg\varphi)^*$ nicht erfüllbar ist, d. h. wenn $\neg(\neg\varphi)^*$ allgemeingültig ist. Zieht man in dieser Fomel die äußere Negation nach innen, erhält man die existenzielle Formel φ_*. □

Beispiel: (a) Die Skolem-Normalform von $\forall v_0 \exists v_1 \forall v_2 \exists v_3 \, (\neg R v_0 v_1 \vee R v_3 v_2)$ ist also

$$\forall v_0 \forall v_2 \, (\neg R \, v_0 \, f v_0 \vee R \, g v_0 v_2 \, v_2)$$

mit neuem einstelligem Funktionszeichen f und neuem zweistelligem Funktionszeichen g. Für

[29] Die in Teil (b) benötigten Konstanten und Funktionen sind in der Regel nicht die gleichen wie in Teil (a); $\mathcal{L}^*$ soll also groß genug sein, um beide Formen zu ermöglichen.

die Herbrand-Normalform führt man folgende Schritte durch:

$$\begin{aligned} \textit{Formel negieren und pränex machen:} &\quad \exists v_0 \forall v_1 \exists v_2 \forall v_3 \, \neg(\neg R v_0 v_1 \vee R v_3 v_2) \\ \textit{Skolemisieren:} &\quad \forall v_1 \forall v_3 \, \neg(\neg R c v_1 \vee R v_3 h v_1) \\ \textit{erneut negieren und pränex machen:} &\quad \exists v_1 \exists v_3 \, (\neg R c v_1 \vee R v_3 h v_1) \end{aligned}$$

wobei c neue Konstante und h neues einstelliges Funktionszeichen. Im Endeffekt eliminiert man also die Existenzquantoren in gleicher Weise wie bei der Skolemisierung die Allquantoren.

(b) Bei mehreren Existenzquantoren hintereinander braucht man jeweils ein neues Funktionszeichen gleicher Stelligkeit. $\forall v_2 \forall v_0 \exists v_1 \exists v_3 \, (\neg R v_0 v_1 \vee R v_3 v_2)$ wird bei der Skolemisierung zu $\forall v_2 \forall v_0 \, (\neg R \, v_0 \, f v_0 v_2 \vee R \, g v_0 v_2 \, v_2)$ mit neuen zweistelligen Funktionszeichen f und g.

Erinnerung: Ein Term heißt *geschlossen*, wenn er keine Individuenvariablen enthält.

Satz 2.6.5 (Satz von Herbrand, 1930) *Sei $\mathcal{L}$ eine Sprache mit mindestens einer Konstante und $\varphi = \exists v_1 \ldots \exists v_n \, \psi$ eine existenzielle Aussage mit quantorenfreiem ψ. Dann ist φ genau dann allgemeingültig, wenn es geschlossene $\mathcal{L}$-Terme $\tau_{1i}, \ldots, \tau_{ni}$ für ein $N \in \mathbb{N}$ und $i = 1, \ldots, N$ gibt so dass*

$$\bigvee_{i=1}^{N} \psi(\tau_{i1}, \ldots, \tau_{in}) := \big(\psi \tfrac{\tau_{11}}{v_1} \ldots \tfrac{\tau_{1n}}{v_n} \; \vee \cdots \vee \; \psi \tfrac{\tau_{N1}}{v_1} \ldots \tfrac{\tau_{Nn}}{v_n} \big)$$

allgemeingültig ist.

Bemerkung: Man kann für N keine Schranke angeben und sollte sich typischerweise eine große Zahl darunter vorstellen.

Beweis: Klar ist, dass $\psi \frac{\tau_{i1}}{v_1} \ldots \frac{\tau_{in}}{v_n} \vDash \exists v_1 \ldots \exists v_n \, \psi$, also auch $\bigvee_{i=1}^{N} \psi(\tau_{i1}, \ldots, \tau_{in}) \vDash \exists v_1 \ldots \exists v_n \, \psi$, was die Richtung „$\Leftarrow$“ beweist.

Für „$\Rightarrow$“ nehmen wir an, dass φ allgemeingültig ist, aber keine der möglichen Disjunktionen von Term-Einsetzungen. Also ist jedes $\neg \bigvee_{i=1}^{N} \psi(\tau_{i1}, \ldots, \tau_{in}) \;\sim\; \bigwedge_{i=1}^{N} \neg\psi(\tau_{i1}, \ldots, \tau_{in})$ erfüllbar. Dies bedeutet, dass die Menge $\{\neg\psi \frac{\tau_1}{v_1} \ldots \frac{\tau_n}{v_n} \mid \tau_1, \ldots, \tau_n \text{ geschlossene } \mathcal{L}\text{-Terme}\}$ endlich erfüllbar ist und nach dem Kompaktheitssatz 2.5.17 also ein Modell $\mathfrak{M}$ hat.

Sei $M_0 := \{\tau^{\mathfrak{M}} \mid \tau \text{ geschlossener } \mathcal{L}\text{-Term}\}$ die Teilmenge von M, die aus den Interpretationen geschlossener Terme besteht. Per Definition ist M_0 unter den Funktionen $f^{\mathfrak{M}}$ für $f \in \mathcal{L}$ abgeschlossen, da

$$f^{\mathfrak{M}}(\tau_1^{\mathfrak{M}}, \ldots, \tau_n^{\mathfrak{M}}) = f\tau_1 \ldots \tau_n^{\mathfrak{M}}.$$

Also ist M_0 Universum einer $\mathcal{L}$-Unterstruktur $\mathfrak{M}_0$ von $\mathfrak{M}$. Unterstrukturen sind gerade so definiert, dass alle geschlossenen Terme die gleiche Interpretation in $\mathfrak{M}_0$ wie in $\mathfrak{M}$ haben und dass alle atomaren $\mathcal{L}$-Aussagen die gleiche Gültigkeit in $\mathfrak{M}_0$ wie in $\mathfrak{M}$ haben. Also gilt auch eine quantorenfreie $\mathcal{L}$-Aussagen in $\mathfrak{M}_0$ genau dann, wenn sie in $\mathfrak{M}$ gilt. Insbesondere folgt daraus $\mathfrak{M}_0 \vDash \neg\psi \frac{\tau_1}{v_1} \ldots \frac{\tau_n}{v_n}$ für alle Terme $\tau_1, \ldots, \tau_n$. Da M_0 aber nur aus Interpretationen von Termen besteht, gilt $\mathfrak{M}_0 \vDash \forall v_1 \ldots \forall v_n \neg\psi \;\sim\; \neg \exists v_1 \ldots \exists v_n \, \psi \;\sim\; \neg\varphi$: im Widerspruch zur Allgemeingültigkeit von φ! □

Bemerkung 2.6.6 (a) *Wenn man $\mathcal{L}$-Aussage ohne Gleichheitszeichen betrachtet, kommt auch in der Herbrand-Normalform kein Gleichheitszeichen vor. Durch die Term-Einsetzungen erhält man dann eine Disjunktion quantorenfreier $\mathcal{L}$-Aussagen. Die Frage, ob solche eine Aussage*

allgemeingültig ist, lässt sich in ein rein aussagenlogisches Problem überführen: Eine quantorenfreie $\mathcal{L}$-Aussage φ ohne Gleichheitszeichen ist genau dann allgemeingültig, wenn ihr „aussagenlogisches Gerüst“ F_φ eine Tautologie ist. Dabei entsteht F_φ aus φ, indem man die atomaren Teilformeln von φ durch Aussagenvariablen ersetzt (wobei zwei atomare Formeln genau dann durch die gleiche Aussagenvariable ersetzt werden, wenn sie gleich sind).

Um die Allgemeingültigkeit von F_φ zu überprüfen, versucht man am besten mit der Resolutionsmethode die Nicht-Erfüllbarkeit von $\neg F_\varphi$ zu zeigen. Indem man das Negationszeichen nach innen zieht, erhält man de facto folgende Variante der Resolutionsmethode:

Man bringt F_φ in DNF und betrachtet die einzelnen Disjunktionsglieder als „duale Klauseln“, d. h. jedes Disjunktionsglied wird zu einer Menge von Literalen, die als ihre Konjunktion verstanden wird. Auf diese dualen Klauseln wenden man nun Resolution an, und sieht, dass F_φ genau dann allgemeingültig ist, wenn sich die leere duale Klausel ergibt.

(b) *Man kann zeigen, dass man sich immer auf eine Aussage ohne Gleichheitszeichen zurückziehen kann: Sei E ein neues zweistelliges Relationszeichen und $\mathcal{L}_E := \mathcal{L} \cup \{E\}$. Man ersetzt nun jede Teilformel $\tau_1 \doteq \tau_2$ einer $\mathcal{L}$-Formel φ durch $E\tau_1\tau_2$, und erhält dadurch eine $\mathcal{L}_E$-Formel φ_E ohne Gleichheitszeichen. Sei $K_E := \{\gamma_E \mid \gamma\ \mathcal{L}\text{-Gleichheitsaxiom}\}$. In jedem Modell $\mathfrak{M}$ von K_E ist dann $E^{\mathfrak{M}}$ eine Äquivalenzrelation (sogar eine Kongruenzrelation bzgl. $\mathcal{L}$), und auf der Menge $M/E^{\mathfrak{M}}$ der Äquivalenzklassen kann man eine $\mathcal{L}_E$-Struktur $\mathfrak{M}/E$ so definieren, dass die natürliche Surjektion $M \to M/E, m \mapsto m/E^{\mathfrak{M}}$ ein $\mathcal{L}_E$-Homomorphismus ist. In $\mathfrak{M}/E$ wird E durch die Identität interpretiert, d. h. $\mathfrak{M}/E \vDash \varphi_E \iff \mathfrak{M}/E \vDash \varphi$. Man kann jede $\mathcal{L}$-Struktur $\mathfrak{N}$ als von der Form $\mathfrak{M}/E$ auffassen: Indem man E durch die Identität interpretiert, wird $\mathfrak{N}$ zu einer $\mathcal{L}_E$-Struktur $\mathfrak{N}^+ \vDash K_E$, für die $\mathfrak{N}^+/E = \mathfrak{N}^+$ gilt. Schließlich kann man zeigen: $\mathfrak{M}/E \vDash \varphi \iff \mathfrak{M} \vDash \varphi_E$. Aus all dem folgt, dass die $\mathcal{L}$-Formel φ genau dann allgemeingültig ist, wenn die $\mathcal{L}_E$-Formel ohne Gleichheitszeichen $\tilde{\varphi}_E := (\bigwedge K_E \to \varphi_E)$ es ist.*

Anstelle von E könnte man natürlich ein gleichheitsartiges Symbol wie etwa $\mathring{=}$ wählen. Da in $\tilde{\varphi}_E$ kein Gleichheitszeichen mehr vorkommt, könnte man es darin sogar wieder durch $\doteq$ ersetzen. Letztendlich läuft die Prozedur also darauf hinaus, die Formel φ durch $(\bigwedge K \to \varphi)$, wobei K die Menge der $\mathcal{L}$-Gleichheitsaxiome ist.

Wenn man die Allgemeingültigkeit von z. B. $(\exists v_0 \forall v_1\, Rv_0v_1 \to \forall v_1 \exists v_0\, Rv_0v_1)$ mit dem Satz von Herbrand zeigen möchte, muss man also geeignete Term-Einsetzungen in die Herbrand-Normalform $\exists v_1 \exists v_3\, (\neg Rcv_1 \vee Rv_3hv_1)$ betrachten. Eine Chance, die Allgemeingültigkeit einer Disjunktion von Term-Einsetzungen zu zeigen, hat man nur, wenn gleiche atomare Formeln auftreten. In diesem Fall möchte man also durch geeignete Einsetzungen erreichen, dass aus Rcv_1 und Rv_3hv_1 die gleiche Formel wird. Hier erreicht man dies offenbar, indem man zum einen c für v_1 und für v_3 einsetzt und dadurch $(\neg Rcc \vee Rchc)$ erhält, und zum andern hc für v_1 und z.B. c für v_3 einsetzt und dadurch $(\neg Rchc \vee Rchhc)$ erhält. Die Disjunktion beider Formeln ist offenbar allgemeingültig (das „aussagenlogische Gerüst“ ist $((\neg A_0 \vee A_1) \vee (\neg A_1 \vee A_2))$).

Unifikation

Die Frage, wie man durch Term-Einsetzungen gleiche atomare Formeln erreichen kann, wird durch das Verfahren der Unifikation geklärt.

Definition 2.6.7 *Sei $P = \{(\tau_1, \tau_1'), \ldots, (\tau_n, \tau_n')\}$ eine Menge von Paaren von $\mathcal{L}$-Termen. P*

wird durch eine Folge von Substitutionen $(\frac{\sigma_1}{v_{i_1}}, \ldots, \frac{\sigma_k}{v_{i_k}})$ *unifiziert, falls*

$$(\cdot((\tau_j \frac{\sigma_1}{v_{i_1}})\frac{\sigma_2}{v_{i_2}})\ldots)\frac{\sigma_k}{v_{i_k}} = (\cdot((\tau'_j \frac{\sigma_1}{v_{i_1}})\frac{\sigma_2}{v_{i_2}})\ldots)\frac{\sigma_k}{v_{i_k}}$$

für alle $j = 1, \ldots, n$ *(die Substitutionen werden hier also nacheinander ausgeführt!).*[30]
Die Menge P *heißt dann unifizierbar und die Folge der Substitutionen Unifikator von* P*.*

Satz 2.6.8 (J. A. Robinson 1965) *Falls* P *unifizierbar ist, so gibt es einen minimalen Unifikator (auch* Haupt-Unifikator *genannt), d. h. jeder Unifikator geht aus dem Hauptunifikator durch weitere Substitutionen hervor.*

Beispiel: (a) Wenn f einstellig, g und h zweistellig, k dreistellig und c eine Konstante ist, dann sind die beiden Terme $gv_1hv_2v_3$ und gfv_3v_4 durch den Hauptunifikator $(\frac{fv_3}{v_1}, \frac{hv_2v_3}{v_4})$ unifizierbar und ergeben beide den Term $gfv_3hv_2v_3$. Aber auch $gffv_1hfv_1fv_1$ ist mögliches Ergebnis einer Unifikation durch $(\frac{ffv_1}{v_1}, \frac{fv_1}{v_2}, \frac{fv_1}{v_3}, \frac{hfv_1fv_1}{v_4})$.

Dagegen sind $k\,v_0\,gv_0v_1\,v_3$ und $k\,fv_3\,v_3\,c$ nicht unifizierbar, da v_3 sowohl mit c als auch mit gv_0v_1 unifiziert werden müsste, was offenbar nicht geht.

(b) Der Hauptunifikator kann je nach Reihenfolge der Substitutionen verschiedene Gestalt annehmen und ist nur im Ergebnis eindeutig. Wenn z. B. $P = \{(v_0, fc), (fv_0, v_1)\}$, dann ist sowohl $(\frac{fc}{v_0}, \frac{ffc}{v_1})$ als auch $(\frac{fv_0}{v_1}, \frac{fc}{v_0})$ Hauptunifikator; beide liefern als Ergebnis die unifizierten Paare (fc, fc) und (ffc, ffc).

Anderes Beispiel: Für $P = \{(v_0, v_1)\}$ sind $\frac{v_0}{v_1}$ und $\frac{v_1}{v_0}$ Hauptunifikatoren, aber auch $(\frac{v_3}{v_0}, \frac{v_3}{v_1})$.

Folgendes Verfahren entscheidet, ob eine Menge von Paaren unifizierbar ist und bestimmt ggf. den Hauptunifikator:

1. Sei P gegeben. Starte mit leerer Folge Σ an Substitutionen.
2. Betrachte ein beliebiges $(\tau, \tau') \in P$:
   (a) Falls $\tau = f\varrho_1 \ldots \varrho_k$ und $\tau' = g\sigma_1 \ldots \sigma_l$ mit Funktionszeichen $f \neq g$ (auch Konstanten), dann stoppe mit Ausgabe „P *ist nicht unifizierbar*“.
   (b) Falls $\tau = f\varrho_1 \ldots \varrho_k$ und $\tau' = f\varrho'_1 \ldots \varrho'_k$, dann ersetze P durch $\big(P \setminus \{(\tau, \tau')\}\big) \cup \big\{(\varrho_1, \varrho'_1), \ldots, (\varrho_k, \varrho'_k)\big\}$.
   (c) Falls $\tau = v_i = \tau'$, ersetze P durch $P \setminus \{(\tau, \tau')\}$.
   (d) Falls $\tau = v_i \neq \tau'$ und v_i in τ' vorkommt, dann stoppe mit Ausgabe „P *ist nicht unifizierbar*“.
   (e) Falls $\tau = v_i$ und v_i in τ' nicht vorkommt, dann hänge $\frac{\tau'}{v_i}$ an Σ an und ersetze P durch $\big\{(\sigma\frac{\tau'}{v_i}, \sigma'\frac{\tau'}{v_i}) \;\big|\; (\sigma, \sigma') \in P,\ (\sigma, \sigma') \neq (\tau, \tau')\big\}$.
   (f) Die letzten beiden Fälle gelten ebenso mit vertauschten Rollen für τ und τ'.
3. Falls $P \neq \emptyset$: springe zu 2. Falls $P = \emptyset$: stoppe mit Ausgabe „*Hauptunifikator ist* Σ“.

Beweis von Satz 2.6.8 und der Korrektheit des Verfahrens:
Das *Unifkationsmaß* einer Menge P von Termpaaren sei $(\#V, \#F, \#P)$, wobei $\#V$ die Anzahl

[30] Man beachte den Unterschied: Bei gleichzeitiger Substitution ist z. B. $gv_0v_1\frac{v_1}{v_0}\frac{v_2}{v_1} = gv_1v_2$, bei nacheinander ausgeführter Substitution ist $(gv_0v_1\frac{v_1}{v_0})\frac{v_2}{v_1} = gv_1v_1\frac{v_2}{v_1} = gv_2v_2$.

der in P vorkommenden Individuenvariablen sei (jede Variable einmal gezählt, unabhängig wie oft sie vorkommt), $\#F$ die Anzahl von Vorkommen von Funktionszeichen und Konstanten in P und $\#$ die Anzahl von Paaren in P. Diese Tripel werden lexikografisch geordnet.

(2b) reduziert $\#F$, ohne $\#V$ zu ändern; (2c) reduziert $\#P$, ohne $\#F$ zu ändern und ohne $\#V$ zu erhöhen; (2e) reduziert $\#V$: In jedem Verfahrensschritt wird das Unifikationsmaß also kleiner, und daher stoppt das Verfahren nach endlich vielen Schritten mit $P = \emptyset$.

Im Fall (2a) ist klar, dass eine Substitutionsfolge genau dann $(\varrho_1, \varrho_1'), \ldots, (\varrho_k, \varrho_k')$ unifiziert, wenn sie (τ, τ') unifiziert. Im Fall (2c) wird lediglich ein bereits unifiziertes Termpaar entfernt. Diese beiden Schritte sind also neutral in dem Sinne, dass sie die Unifizierbarkeit bzw. Nicht-Unifizierbarkeit von P nicht ändern.

Im Fall (2e) muss eine Unifikation $\tau = v_i$ mit τ' identifizieren, also τ' für v_i einsetzen. Somit ist eine ausgegebene Substitutionsfolge zum einen tatsächlich ein Unifikator, und zum andern minimal. Schließlich ist es im Fall (2a) offensichtlich und im Fall (2d) leicht einzusehen, dass keine Unifikation möglich ist. Also ist das Verfahren auch insgesamt korrekt. □

Zusammenfassung der Herbrand’schen Methode:

Es gibt folgendes Semi-Entscheidungsverfahren für die Allgemeingültigkeit einer Formel φ:

1. Gegebenenfalls eliminiert man das Gleichheitszeichen, bringt dann φ in pränexe Normalform und den quantorenfreien Teil von φ in DNF.
2. Nun bestimmt man die Herbrand’sche Normalform φ_*.
3. Mithilfe von Unifkation sucht man N geeignete Term-Einsetzungen.
4. Mit der dualen Resolutionsmethode testet man die Disjunktion der Term-Einsetzungen auf Allgemeingültigkeit.
5. Man durchläuft Schritte 3. und 4. mit wachsendem N, bis sich eine allgemeingültige Disjunktion von Term-Einsetzungen ergibt. In diesem Fall ist φ allgemeingültig. Andernfalls erhält man keine Antwort und das Verfahren stoppt nicht.

Umgekehrt gilt: wenn φ allgemeingültig ist, gibt es ein N und geeignete Term-Einsetzungen, welche dies nachweisen. Wenn man die möglichen Term-Einsetzungen systematisch abarbeitet, gibt die Methode für allgemeingültige Formeln irgendwann ein Ergebnis. Daher ist es ein Semi-Entscheidungsverfahren.

Beispiel: Sei $\mathcal{L} = \{\circ, e\}$ mit einem zweistelliges Funktionssymbol $\circ$ und einer Konstanten e. Falls in einer $\mathcal{L}$-Struktur $\mathfrak{M}$ die Operation $\circ^{\mathfrak{M}}$ kommutativ ist, $e^{\mathfrak{M}}$ neutrales Element ist und jedes Element ein Linksinverses hat, dann hat natürlich auch jedes Element ein Rechtsinverses. Genaueres Hinschauen zeigt, dass bereits aus den beiden abgeschwächten Voraussetzungen $\forall v_0 \forall v_1\, (v_0 \circ v_1 \doteq e \leftrightarrow v_1 \circ v_0 \doteq e)$ und $\forall v_0 \exists v_1\, v_0 \circ v_1 \doteq e$ die Aussage $\forall v_0 \exists v_1\, v_1 \circ v_0 \doteq e$ folgt.

Dies soll mit der Herbrand’schen Methode nachvollzogen werden, d. h. es soll gezeigt werden, dass die Formel

$$\varphi = ((\forall v_0 \forall v_1\, (v_0 \circ v_1 \doteq e \leftrightarrow v_1 \circ v_0 \doteq e) \wedge \forall v_0 \exists v_1\, v_0 \circ v_1 \doteq e) \rightarrow \forall v_0 \exists v_1\, v_1 \circ v_0 \doteq e)$$

allgemeingültig ist.

(1) Der erste Schritt würde nun darin bestehen, die Gleichheitsaxiome hinzuzunehmen und φ zu ersetzen durch

$$\begin{aligned}\big((\forall v_0\, v_0 \doteq v_0 \wedge \forall v_0 \forall v_1\, (v_0 \doteq v_1 \leftrightarrow v_1 \doteq v_0) \\ \wedge \forall v_0\, \forall v_1 \forall v_2 ((\, v_0 \doteq v_1 \wedge v_1 \doteq v_2) \rightarrow v_0 \doteq v_2) \\ \wedge \forall v_0 \ldots \forall v_3\, ((v_0 \doteq v_2 \wedge v_1 \doteq v_3) \rightarrow v_0 \circ v_1 \doteq v_2 \circ v_3)) \;\rightarrow\; \varphi \big)\end{aligned}$$

Es stellt sich aber heraus, dass dies in diesem speziellen Fall nicht nötig ist. Damit es überschaubar bleibt, arbeite ich deshalb mit dem originalen φ.[31] Außerdem schreibe ich der besseren Lesbarkeit halber $\not\doteq$ für eine verneinte Gleichheit und $u, \ldots, z$ für die Individuenvariablen.

Als nächstes wird φ in pränexe Normalform mit quantoremfreiem Teil in DNF gebracht. Dabei bringt man die Allquantoren am besten möglichst weit nach außen, also etwa:

$$\forall y \exists w \forall x \exists u \exists v \exists z \big((u \circ v \doteq e \wedge v \circ u \not\doteq e) \vee (u \circ v \not\doteq e \wedge v \circ u \doteq e) \vee w \circ x \not\doteq e \vee z \circ y \doteq e \big)$$

(2) Die Herbrand'sche Normalform φ_* ist dann

$$\exists w \exists u \exists v \exists z \big((u \circ v \doteq e \wedge v \circ u \not\doteq e) \vee (u \circ v \not\doteq e \wedge v \circ u \doteq e) \vee w \circ fw \not\doteq e \vee z \circ c \doteq e \big)$$

mit neuer Konstante c und neuem einstelligen Funktionszeichen f.

Man hat nun unendlich viele Terme e, c, fe, fc, $e \circ e$, $e \circ c$, $c \circ e$, $c \circ c$, ffe, ffc, $e \circ fe$, …, die man in allen möglichen Kombinationen für u, v, w und z in φ_* einsetzen kann. Der Überschaubarkeit der Darstellung wegen wähle ich hier einige zielführende Einsetzungen aus und erläutere, wie man auf diese Einsetzungen kommt. Ein automatisiertes Vorgehen wird dagegen systematisch alle möglichen Einsetzungen vornehmen, die an einer Stelle eine atomare Teilformel ψ und an anderer Stelle (und ggf. durch eine andere Einsetzung) die negiert-atomare Teilformel ψ entstehen lassen.

(3a) Für eine erste Einsetzung sucht man eine atomare Teilformel und eine negierte atomare Teilformel „gleicher Gestalt“[32] die durch geeignete Term-Einsetzungen in der anschließenden Resolution einen Resolutionsschritt ermöglichen. Man wählt etwa $v \circ u \not\doteq e$ und $z \circ c \doteq e$, und unifiziert also $\{(v, z), (u, c)\}$, was durch den Hauptunifikator $(\frac{c}{u}, \frac{z}{v})$ geschieht. Setzt man nach dieser Substitution einen geschlossenen Term, zum Beispiel c, für w und z in φ_* ein, erhält man

$$\big((c \circ c \doteq e \wedge c \circ c \not\doteq e) \vee (c \circ c \not\doteq e \wedge c \circ c \doteq e) \vee c \circ fc \not\doteq e \vee c \circ c \doteq e \big)$$

(4a) Die atomaren Teilformeln werden nun durch Aussagenvariablen ersetzt und die Konjunktionsglieder als duale Klauseln aufgefasst. Durch A_0 für $c \circ c \doteq e$ und A_1 für $c \circ fc \doteq e$ erhält man die dualen Klauseln

$$\{A_0, \neg A_0\}, \{\neg A_0, A_0\}, \{\neg A_1\}, \{A_0\}.$$

Die Menge dieser dualen Klauseln ist bereits unter Resolution abgeschlossen und enthält nicht die leere Klausel. Also hat man noch keinen Beweis der Allgemeingültigkeit von φ.

(3b) Um durch die nächste Term-Einsetzung einen neuen Resolutionsschritt zu ermöglichen, möchte man eine Klausel mit positivem A_1 erhalten, im Idealfall sogar $\{A_1\}$. Positive Einerklauseln bekommt man nur aus der Teilformel $z \circ c \doteq e$. Der Versuch, daraus A_1, also $c \circ fc \doteq e$ zu erhalten, scheitert aber, da $\{(z, c), (c, fc)\}$ nicht unifizierbar ist.

[31] Würde man die volle Kommutativität $\forall v_0 \forall v_1\, (v_0 \circ v_1 \doteq v_1 \circ v_0)$ anstelle der ersten Voraussetzung nehmen, bräuchte man dagegen zumindest die Transitivität der Gleichheit.

[32] In diesem Beispiel haben alle atomaren Teilformen die gleiche Gestalt $\tau_1 \circ \tau_2 \doteq e$.

(3c) Im zweiten Versuch, eine Klausel mit positivem A_1 zu erhalten, betrachtet man $w \circ fw \neq e$ und $v \circ u \doteq e$ und unifiziert $\{(v, w), (u, fw)\}$ durch den Hauptunifikator $(\frac{w}{v}, \frac{fw}{u})$. Setzt man nach dieser Substitution c für w (und auch für z) in φ_* ein, erhält man

$$\big((fc \circ c \doteq e \wedge c \circ fc \neq e) \vee (fc \circ c \neq e \wedge c \circ fc \doteq e) \vee c \circ fc \neq e \vee c \circ c \doteq e \big)$$

zusammen mit der Einsetzung aus (3a) also die Disjunktion

$$\begin{aligned}\big((c \circ c \doteq e \wedge c \circ c \neq e) \ \vee \ (c \circ c \neq e \wedge c \circ c \doteq e) \vee \ c \circ fc \neq e \ \vee \ c \circ c \doteq e \vee \\ (fc \circ c \doteq e \wedge c \circ fc \neq e) \vee (fc \circ c \neq e \wedge c \circ fc \doteq e) \vee \ c \circ fc \neq e \ \vee \ c \circ c \doteq e \big)\end{aligned}$$

(was nun eine Disjunktion von Term-Einsetzungen ist, wie sie im Satz von Herbrand vorkommt).
(4c) Die neue atomare Formel $fc \circ c \doteq e$ wird durch A_2 ersetzt; zusammen mit der ersten Term-Einstzung erhält man also die dualen Klauseln

$$\{A_0, \neg A_0\}, \{\neg A_0, A_0\}, \{\neg A_1\}, \{A_0\}, \{A_2, \neg A_1\}, \{\neg A_2, A_1\}, \{\neg A_1\}, \{A_0\}.$$

Der Abschluss der *Menge* der dualen Klauseln unter Resolution ist

$$\big\{\{A_0\}, \{\neg A_1\}, \{\neg A_2\}, \{A_1, \neg A_2\}, \{\neg A_1, A_2\}, \{A_0, \neg A_0\}, \{A_1, \neg A_1\}, \{A_2, \neg A_2\}\big\}$$

(3d) Man sieht, dass man zum Ziel kommt, wenn man die Klausel $\{A_2\}$ erhält. Will man, analog zu (3b) A_2 aus $z \circ c \doteq e$ erhalten, also $fc \circ c \doteq e$, muss man $\{(z, fc), (c, c)\}$ unifizieren, was durch den Hauptunifikator $\frac{fc}{z}$ geschieht. Damit in (3c) die A_2 entsprechende Teilformel entsteht, hat man die Ersetzungen $(\frac{w}{v}, \frac{fw}{u}, \frac{c}{w})$ durchgeführt. Zusammen erhält man

$$\big((fc \circ c \doteq e \wedge c \circ fc \neq e) \vee (fc \circ c \neq e \wedge c \circ fc \doteq e) \vee c \circ fc \neq e \vee fc \circ c \doteq e \big)$$

(4d) Damit bekommt man eine neue duale Klausel $\{A_2\}$, im Abschluss unter Resolution also die leere Klausel und damit einen Beweis der Allgemeingültigkeit von φ.

(5) Wenn man genauer hinsieht, erkennt man, dass man bereits durch eine Einsetzung (also $N = 1$, nämlich fc für u und z und c für v und w) zum Ziel kommt. Im Beispiel ist dargestellt, wie man sich durch ein ansatzweise systematisches Vorgehen über die Schritte $N = 1, 2, 3$ dem Ziel annähert. Das funktioniert hier gut, weil man im Prinzip nur eine Möglichkeit hat, das Resolutionsverfahren weiterzutreiben. Bei umfangreicheren Formeln wird man nicht so zielgerichtet vorgehen können.

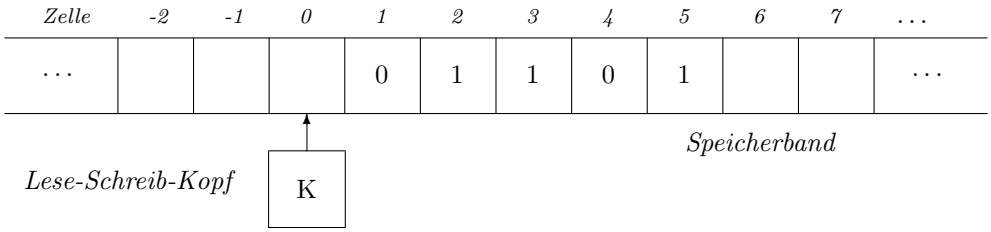
Die gerade angegebene Einsetzung, die bereits alleine ausreicht, kann man übrigens aus dem algebraischen Beweis bekommen: Dafür würde man sich zunächst ein beliebiges $c \in M$ wählen. Nach Voraussetzung gibt es zu jedem $m \in M$ ein Linksinverses. Dieses muss nicht eindeutig bestimmt sein, aber man kann sich eine Funktion $f : M \to M$ wählen, welches für jedes $m \in M$ ein Linksinverses angibt, also $f(m) \circ m = e$. (Diese Schritte entsprechen der Einführung der Skolem-Funktionen!) Nun gilt also $f(c) \circ c = e$ und nach Voraussetzung auch $f(c) \circ c = e \Leftrightarrow c \circ f(c) = e$, also folgt $c \circ f(c) = e$. Somit hat man ein Rechtinverses gefunden.

3 Berechenbarkeit

3.1 Turing-Maschinen

Turing-Maschinen stellen ein von Alan Turing 1936 – also deutlich vor der Konstruktion der ersten Computer – entwickeltes, besonders einfaches Rechnermodell dar. Es hat sich im Laufe der Zeit aber herausgestellt, dass der Begriff der Berechenbarkeit durch Turing-Maschinen mindestens ebenso umfassend wie der Begriff der Berechenbarkeit durch konkrete Rechner bzw. durch andere abstrakte Rechnermodelle ist (vgl. die *Church'sche These* S. 3.3).

Es gibt verschiedene Varianten von Turing-Maschinen, die sich aber problemlos gegenseitig simulieren lassen. Bei der hier vorgestellten Variante besteht die „*Hardware*“ aus einem beidseitig unendlichen *Speicherband* mit einzelnen, nebeneinander liegenden *Speicherzellen* (die man sich also als mit den ganzen Zahlen $\mathbb{Z}$ indiziert vorstellen kann), und einem *Lese-Schreib-Kopf*, der in jedem Berechnungsschritt auf genau eine Speicherzelle zugreift.



Die „*Programmiersprache*“ einer Turing-Maschine arbeitet mit einem endlichen *Alphabet* $A \neq \emptyset$. In jeder Zelle des Speicherbandes steht genau ein Symbol aus $A^+ := A \cup \{\emptyset\}$, wobei $\emptyset$ als Symbol bedeuten soll, dass die Zelle *leer* ist. (Bei einem Alphabet mit $0 \in A$ muss also zwischen einer leeren Zelle und einer Zelle mit dem Wert 0 unterschieden werden). Ein einelementiges Alphabet A ist gestattet. Sodann gibt es eine endliche Menge Z an *Zuständen*, darunter den *Startzustand* $z_0 \in Z$ und den *Endzustand* $z_{\text{end}} \in Z$. Zu jedem Zeitpunkt befindet sich die Maschine in genau einem Zustand. Der Zustand der Maschine ist also eine zusätzliche abstrakte Gegebenheit, die unabhängig von der Position des Lese-Schreib-Kopf und der aktuellen Informaiton auf dem Speicherband ist.

Das „*Programm*“ einer Turing-Maschine besteht aus einer *Übergangsfunktion*

$$U : \; Z \times A^+ \; \to \; Z \times A^+ \times \{-1, 0, 1\}$$

wobei $U(z_{\text{end}}, a) = (z_{\text{end}}, a, 0)$ für alle $a \in A^+$. Die Übergangsfunktion kann man sich vorstellen als eine Liste von Zuweisungen $(z, a) \mapsto (z', a', p')$.

Der Programmlauf einer Turing-Maschine funktioniert dann folgendermaßen:

- In der Startkonfiguration steht der Lese-Schreib-Kopf in einer definierten Startzelle. Alle Zellen links bis einschließlich der Startzelle sind leer. Unmittelbar in den l Zellen rechts von der Startzelle stehen Symbole aus A, die ein Wort der Länge l aus A^* als Eingabe des Programms bilden. Alle weiteren Zellen rechts davon sind leer (wie im Bild oben angedeutet). Die Turing-Maschine befindet sich außerdem im Startzustand z_0.
- In jedem Programmschritt befindet sich die Turing-Maschine in genau einem Zustand z und liest das Symbol $a \in A^+$, welches in der Zelle n steht, an der sich der Lese-Schreib-Kopf

befindet. Das Programm wertet dann $U(z,a) = (z',a',p')$ aus, die Maschine ersetzt das Symbol a in Zelle n durch a', bewegt den Lese-Schreib-Kopf in die Zelle $n+p'$ und geht in den Zustand z' über. Damit ist der Programmschritt beendet und der nächste folgt.

- Sobald sich weder der Zustand, noch die Position des Kopfes, noch die Symbole auf dem Band mehr ändern, sagt man „Die Maschine hält an“ oder „Die Maschine stoppt“. Per Definition stoppt die Maschine also stets dann, wenn sie den Endzustand z_{end} erreicht hat. In diesem Fall sagt man auch, dass die Maschine die Eingabe *akzeptiert*.

Eine Turing-Maschine kann für Entscheidungs- und für Berechnungsprobleme eingesetzt werden.

(A) Im Falle eines *Entscheidungsproblems* lässt man die Turing-Maschine mit Eingabe eines Wortes w aus A^* laufen und wertet es als positive Antwort auf die intendierte Frage, ob w eine gewisse Eigenschaft hat, wenn die Maschine im Endzustand anhält, und als negative Antwort, wenn sie in einen anderen Zustand $z \neq z_{\text{end}}$ anhält.

(B) Bei einem *Berechnungsproblem* wird der Inhalt des Speicherbandes beim Erreichen des Endzustandes in geeigneter Weise als Ausgabe interpretiert (zum Beispiel, indem man durch Entfernen der leeren Zellen ein Wort in A^* bekommt).

Für die folgende Betrachtung ist die Variante der nicht-deterministischen Turing-Maschine wichtig. Sie unterscheidet sich von der gewöhnlichen, *deterministischen* Turing-Maschine nur darin, dass die Übergangsfunktion durch eine *Übergangsrelation*

$$U \subseteq (Z \times A^+) \times (Z \times A^+ \times \{-1,0,1\})$$

ersetzt wird, die folgende zwei Eigenschaft haben muss:

- Für alle $(z,a) \in Z \times A^+$ gibt es mindestens ein Tripel (z',a',p') mit $(z,a,z',a',p') \in U$.
- Falls $(z_{\text{end}},a,z',a',p') \in U$, so ist $z' = z_{\text{end}}, a' = a$ und $p' = 0$.

Nicht-deterministische Turing-Maschinen werden hier nur für Entscheidungsprobleme betrachtet. Sie liefern bei Eingabe eines Wortes aus A^* genau dann die Antwort *ja*, wenn es die Möglichkeit gibt, in jedem Programmschritt im Zustand z und bei aktuellem Bandsymbol a ein Tripel (z',a',p') mit $(z,a,z',a',p') \in U$ so auszuwählen, dass der Endzustand erreicht wird.

3.2 NP-Vollständigkeit und der Satz von Cook

Sei $A \neq \emptyset$ ein endliches Alphabet, z. B. $A = \{0,1\}$. Jede Teilmenge $M \subseteq A^*$ definiert ein *Entscheidungsproblem*, nämlich:

Ist ein gegebenes $w \in A^*$ in M oder nicht?

Verkürzt wird daher die Teilmenge M ein Problem genannt.

Definition 3.2.1 **(a)** *Ein Problem ist (nicht-deterministisch) polynomiell, falls es ein Polynom $T \in \mathbb{R}[X]$ und eine (nicht-deterministische) Turing-Maschine gibt, die für alle $w \in A^*$ in höchstens $T(\lg(w))$ Berechnungsschritten entscheidet*[33]*, ob $w \in M$.*

Die Menge der polynomiellen Probleme wird mit P *(oder* PTIME*) bezeichnet, die Menge der nicht-deterministisch polynomiellen Probleme mit* NP*.*

(b) *Ein Problem M ist NP-schwer, falls es für jedes NP-Problem $N \subseteq B^*$ eine* polynomielle Reduktion *auf M gibt, d. h. eine berechenbare Funktion $r : B^* \to A^*$ mit $w \in N \iff r(w) \in M$ für alle $w \in B^*$, so dass $\lg(r(w)) \leqslant R(\lg(w))$ für ein Polynom $R \in \mathbb{R}[X]$.*

[33] Das heißt: die Maschine gibt die richtige Antwort auf das Entscheidungsproblem.

(c) *Ein Problem M ist* NP*-vollständig, falls es in* NP *liegt und* NP*-schwer ist.*

Der Berechenbarkeitsbegriff wird in Definition 3.3.1 präzisiert, kann hier aber zunächst im intuitiven Sinn verstanden werden.

Die Einordnung von Entscheidungsproblemen in die Klassen P und NP geschieht also durch Einschränkungen an die Berechnungszeit. Die Größe des benötigten Speicherplatzes spielt dabei keine Rolle.[34] Die *Komplexitätstheorie* untersucht diese und viele andere Problemklassen. Dabei gelten die polynomiellen Probleme als eine Annäherung an die „praktisch berechenbaren“ Probleme, da die Berechnungszeit in einem vernünftigen Verhältnis zur Eingabegröße steht (in der Praxis kommt es natürlich auf den Grad und die Größe der Koeffizienten des Polynoms T an). Für manche Probleme in NP kennt man dagegen nur Algorithmen mit exponentieller Berechnungszeit in Abhängigkeit von der Eingabegröße, so dass sich schon bei recht kleinen Eingaben eine exorbitante Berechnungszeit ergibt.

Ein großes offenes Problem der Logik und theoretischen Informatik ist die Frage, ob $\mathsf{P} = \mathsf{NP}$, also ob nicht jedes nicht-deterministische polynomielle Problem schon polynomiell ist. Es ist eines der sieben *Millennium Problems* des Clay Instituts.[35] Die meistens Fachleute gehen davon aus, dass $\mathsf{P} \neq \mathsf{NP}$, aber zum Beispiel wurde erst 2002 ein polynomieller Algorithmus für das Primzahlproblem entdeckt (also für die Frage, ob eine gegebene natürliche Zahl eine Primzahl ist), während das elementare Verfahren, alle kleineren Zahlen als Teiler zu testen, exponentiell ist in der Eingabegröße.

Man sieht zunächst, dass $\mathsf{P} \subseteq \mathsf{NP}$, da man jede deterministische Turing-Maschine auch als nicht-deterministische Turing-Maschine auffassen kann, indem man die Übergangsfunktion durch ihren Graphen als Übergangsrelation ersetzt.[36] Wenn man für ein NP-schweres Problem M zeigen könnte, dass es in P liegt (also auch in NP, d. h. solch ein M wäre notwendigerweise NP-vollständig), hätte man $\mathsf{P} = \mathsf{NP}$ gezeigt, da man dann alle NP-Probleme über die polynomielle Reduktion auf M ebenfalls in polynomieller Zeit lösen könnte. Daher gelten NP-schwere bzw. NP-vollständige Probleme als praktisch nicht berechenbar (wobei es für einige wichtige NP-vollständige Fragestellungen gute Näherungsalgorithmen gibt).

Beispiel: Sei A ein endliches Alphabet, in dem aussagenlogische Formeln codiert werden können. (Im Wesentlichen enthält A also die Junktoren, die Klammern und eine Möglichkeit, unendlich viele Aussagenvariablen zu beschreiben, z. B. dadurch, dass man die Aussagenvariable A_{398652} als Zeichenfolge $A398652$ schreibt.) Dann ist die Menge der Codes aussagenlogischer Formeln ein polynomielles Problem:

Eine Turing-Maschine kann den „äußersten“ Junktor (d. h. den in polnischer Notation am weitestens links stehenden) identifizieren und löschen, ggf. mit zugehörigen Klammern, und so die Formel rekursiv abbauen, bis nur noch Aussagenvariablen übrig bleiben. Sobald andere Zeichen übrig bleiben oder ein Rekursionsschritt nicht klappt, weil die Syntax nicht stimmt, gibt die Maschine eine negative Antwort. Es gibt so viele Rekursionsschritte wie Junktoren, und in jeden Schritt muss die Formel im Wesentlichen einmal durchlaufen werden, d. h. man hat einen quadratischen Algorithmus (polynomiell vom Grad 2) in der Länge des Codes.

[34] Es gibt aber auch die über analoge Einschränkung an den Speicherplatz definierten Problemklassen PSPACE und NPSPACE.

[35] `http://www.claymath.org/millennium-problems`

[36] Die Terminologie ist also nicht besonders exakt; anstelle von „nicht-deterministisch“ wäre *nicht notwendigerweise deterministische Turing-Maschine* präziser.

Lemma 3.2.2 *Ein Problem $M \subseteq A^*$ liegt genau dann in* NP*, wenn es ein Alphabet B, ein Polynom $Q \in \mathbb{R}[X]$ und ein polynomielles Problem $M^+ \subseteq A^* \times B^* \subseteq (A \cup B)^*$ gibt, so dass*

$$w \in M \iff \text{es gibt ein } z \in B^* \text{ mit } w^\frown z \in M^+ \text{ und } \lg(z) \leqslant Q(\lg(w))$$

Die Elemente z heißen dann Zertifikate für w.

Beweisskizze: Wenn es polynomielle Zertifikate gibt, konstruiert man eine nicht-deterministischen Turing-Maschine, die in ihren möglichen Programmläufen alle $z \in B^*$ der Länge höchstens $Q(\lg(w))$ daraufhin testet, ob sie ein Zertifikat für w sind, was für jedes einzelne z in Polynomialzeit geht.

Hat man umgekehrt eine nicht-deterministische Turing-Maschine, die in höchstens $T(\lg(w))$ vielen Schritten entscheidet, ob $w \in M$, dann kann nach jedem Berechnungsschritt auch nur ein Wort der Länge höchstens $\lg(w) + T(\lg(w))$ auf dem Band stehen (da jeder Schritt höchstens ein Symbol hinzufügt). Alle diese Bandzustände hintereinander und zum Beispiel jeweils durch ein neues Symbol getrennt und mit zusätzlichen Symbolen für den jeweils aktuellen Zustand und die jeweils aktuelle Bandposition versehen, beschreiben also den gesamten Programmablauf in einem Wort der Länge höchstens $T(\lg(w)) \cdot (\lg(w) + T(\lg(w)) + 3)$. Dieses Wort dient als Zertifikat, da man sicher in polynomieller Zeit überprüfen kann, ob ein gegebenes Wort auf diese Weise einem Programmablauf der Turing-Maschine entspricht. Man findet also Zertifikate mit einer Länge, die durch das Polynom $Q(X) = T(X) \cdot (T(X) + X + 3)$ beschränkt ist. □

Ich werde in der Folge einfach von Formeln sprechen, wenn ich eigentlich Darstellungen von Formeln in einem endlichen Alphabet (Codes) meine. Die Schwierigkeit hierbei ist, dass durch große Indizes die Länge des Codes beliebig viel größer als die Länge der Formel werden kann. Für das prinzipielle Verständnis des Satzes von Cook kann man aber die Länge der Formel als Eingabegröße nehmen und entweder so tun, als hätte man nur endlich viele Aussagenvariablen, oder als würden die Komplexitätsbetrachtungen auch mit einem unendlichen Alphabet funktionieren.

Beispiel: Die Menge SAT der erfüllbaren aussagenlogischen Formeln ist ein typisches NP-Problem; die erfüllenden Belegungen dienen hier als Zertifikate. Denn mit $B = \{0, 1\}$ erhält man ein polynomielles Problem $\mathsf{SAT}^+ = \{F^\frown\beta \mid \beta(F) = 1\}$, da es (ebenfalls in quadratischer Zeit) möglich ist, den Wahrheitswert einer Formel für eine gegebene Belegung auszurechnen.

3-SAT ist das Problem der erfüllbaren aussagenlogischen Formeln in KNF, in denen maximal drei Literale pro Klausel vorkommen.

Satz 3.2.3 (Satz von Cook und Levin, 1971)
Das Erfüllbarkeitsproblem SAT *ist* NP*-vollständig. Sogar:* 3-SAT *ist* NP*-vollständig.*

Wenn man $\mathsf{SAT} \in \mathsf{P}$ bzw. $\mathsf{SAT} \notin \mathsf{P}$ zeigen könnte, hätte man also die P=NP-Frage beantwortet. Bei der elementaren Methode, Erfüllbarkeit über Wahrheitstafeln zu testen, muss man im Allgemeinen exponentiell viele Belegungen in der Anzahl der vorkommenden Aussagenvariablen betrachten. Andere Verfahren wie z. B. die Resolutionsmethode oder Baumkalküle liefern zwar häufig schnellere Ergebnisse, sind aber auch nicht polynomiell.

Beweis: Dass SAT in NP liegt, wurde im Beispiel oben erläutert. Sei $N \subseteq B^*$ ein NP-Problem, das durch eine nicht-deterministische Turing-Maschine $\mathcal{C}_N$ entschieden wird, wobei das Polynom

$T \in \mathbb{R}[X]$ die Anzahl der Berechnungsschritten beschränke. N soll nun polynomiell auf SAT reduziert werden, indem das Verhalten von $\mathcal{C}_N$ für eine Eingabe w durch eine aussagenlogische Formel $C_{N,w}$ beschrieben wird.

Dazu benutzt man für $t \in \mathbb{N}$, $n \in \mathbb{Z}$, $a \in A^+$ und $z \in Z$ die folgenden Aussagenvariablen:

$S_{t,a,n}$	für:	*Zum Zeitpunkt t steht das Symbol a in der n-ten Speicherzelle.*
$Z_{t,z}$	für:	*Zum Zeitpunkt t befindet sich die Maschine im Zustand z.*
$K_{t,n}$	für:	*Zum Zeitpunkt t befindet sich der Lese-Schreib-Kopf in der n-ten Speicherzelle.*

Dies sind insgesamt abzählbar viele Aussagenvariablen, die man in geeigneter Weise mit den $(A_i)_{i\in\mathbb{N}}$ identifiziert. Für jedes einzelne $w \in A^*$ kann in den höchstens $T(\lg(w))$ vielen Berechnungsschritten allerdings höchstens eine Speicherzelle im Abstand $T(\lg(w))$ von der Startzelle 0 erreicht werden. Man braucht also für jedes w nur die endlich vielen Aussagenvariablen mit $t \leqslant T(\lg(w))$ und $|n| \leqslant T(\lg(w))$.

Für fest gewähltes w und damit auch festes $T := T(\lg(w))$ ist die aussagenlogische Formel $C_{N,w}$ nun die Konjunktion folgender Formeln:

- Zu jedem Zeitpunkt $t \leqslant T$ befindet sich die Maschine genau in einem Zustand:
$$\bigwedge_{t\leqslant T} \Big(\bigvee_{z\in Z} Z_{t,z} \wedge \bigwedge_{z\neq z'} \neg(Z_{t,z} \wedge Z_{t,z'}) \Big)$$
- Zu jedem Zeitpunkt $t \leqslant T$ befindet sich der Kopf in genau einer der betrachteten Zellen:
$$\bigwedge_{t\leqslant T} \Big(\bigvee_{|n|\leqslant T} K_{t,n} \wedge \bigwedge_{n\neq n'} \neg(K_{t,n} \wedge K_{t,n'}) \Big)$$
- Zu jedem Zeitpunkt $t \leqslant T$ steht in jeder Speicherzelle n mit $|n| \leqslant T$ genau ein Symbol:
$$\bigwedge_{t\leqslant T} \bigwedge_{|n|\leqslant T} \Big(\bigvee_{a\in A^+} S_{t,a,n} \wedge \bigwedge_{a\neq a'} \neg(S_{t,a,n} \wedge S_{t,a',n}) \Big)$$
- Zum Zeitpunkt $t = 0$ befindet sich die Maschine im Startzustand und der Kopf in der 0-ten Speicherzelle:
$$(Z_{0,z_0} \wedge K_{0,0})$$
- Die Eingabe ist $w = (a_1, \ldots, a_l)$ mit $a_i \in A$:
$$(S_{0,a_1,1} \wedge S_{0,a_2,2} \wedge \cdots \wedge S_{0,a_l,l} \wedge \bigwedge_{n\leqslant T, n\notin\{1,\ldots,l\}} S_{0,\emptyset,n})$$
- Die Programmschritte laufen in Übereinstimmung mit der Übergangsrelation:
$$\bigwedge_{t\leqslant T-1} \big((S_{t,a,n} \wedge Z_{t,z} \wedge K_{t,n}) \rightarrow \bigvee_{(z,a,z',a',p')\in U} (S_{t+1,a',n} \wedge Z_{t+1,z'} \wedge K_{t+1,n+p'})\big)$$

Da A, Z und U fest sind, sieht man, dass $C_{N,w}$ eine polynomielle Länge in T hat, also auch eine polynomielle Länge in $l = \lg(w)$.

Die Turing-Maschine $\mathcal{C}_N$ akzeptiert nun w in höchstens T Berechnungsschritten genau dann, wenn die Formel $(C_{N,w} \wedge Z_{T,z_{\text{end}}})$ erfüllbar ist. Mit Lemma 1.5.1 kann diese Formel in eine erfüllbarkeitsäquivalente 3-SAT-Formel überführt werden, wobei sich die Länge nur um eine Konstante ändert, also polynomiell in $\lg(w)$ bleibt. □

3.3 Das Halteproblem und die Unentscheidbarkeit der Prädikatenlogik

Sei $A = \{a_1, \ldots, a_n\} \neq \emptyset$ ein endliches Alphabet. Wie üblich bezeichnet A^* die Menge der endlichen Wörter über A.

Wenn $D \subseteq M$ und $f : D \to N$ eine Funktion ist, kann man f als *partielle Funktion* von M nach N betrachten, Notation: $f : M \dashrightarrow N$. Für $x \in D$ ist $f(x)$ der Funktionswert von x und f heißt *definiert* in x; für $x \in M \setminus D$ heißt f an der Stelle x *unbestimmt*.

Jede Turing-Maschine $\mathcal{C}$ mit Alphabet A bestimmt eine partielle Funktion $f_{\mathcal{C}}$ mit

$$f_{\mathcal{C}}(w) = \begin{cases} v & \text{falls } \mathcal{C} \text{ bei Eingabe } w \text{ stoppt und Ausgabe } v \text{ liefert} \\ \text{unbestimmt} & \text{sonst} \end{cases}$$

Definition 3.3.1 *(a) Eine Funktion $f : A^* \to B^*$ heißt* berechenbar[37]*, falls es eine Turing-Maschine gibt, die bei Eingabe $w \in A^*$ stoppt und die Ausgabe $f(w)$ liefert.*

Eine partielle Funktion $f : A^* \dashrightarrow B^*$ heißt berechenbar, falls es eine Turing-Maschine gibt, die bei Eingabe $w \in A^*$ stoppt und die Ausgabe $f(w)$ liefert, sofern $f(w)$ definiert ist, und nicht stoppt, falls $f(w)$ unbestimmt ist.

(b) Eine Teilmenge $W \subseteq A^*$ heißt entscheidbar*, falls die charakteristische Funktion χ_W mit*

$$\chi_W(w) = \begin{cases} 1 & w \in W \\ 0 & w \notin W \end{cases}$$

berechenbar ist.

Äquivalent: Es gibt eine Turing-Maschine, die bei Eingabe $w \in A^*$ entscheidet, ob $w \in W$. (Denn steht 1 für die Ausgabe „ja“ und 0 für die Ausgabe „nein“, berechnet sie gerade die charakteristische Funktion von W.)

(c) Eine Teilmenge $W \subseteq A^*$ heißt semi-entscheidbar *oder* aufzählbar*, falls die „partielle charakteristische Funktion“ $\overline{\chi_W}$*[38] *mit*

$$\overline{\chi_W}(w) = \begin{cases} 1 & w \in W \\ \text{unbestimmt} & w \notin W \end{cases}$$

berechenbar ist.

Äquivalent: Es gibt eine Turing-Maschine, die bei Eingabe $w \in A^*$ genau dann stoppt, wenn $w \in W$.

Bemerkung 3.3.2
(a) Für jedes endliche, nicht-leere Alphabet $A = \{a_1, \ldots, a_n\}$ gibt es eine Bijektion $\mathbb{N} \to A^*$:

- Falls $|A| = 1$, betrachte die Abbildung, die der Zahl n die Folge $\underbrace{a_1 \cdots a_1}_{n \text{ mal}}$ zuordnet.
- Falls $|A| = n$, kann man zum Beispiel die Elemente von A irgendwie anordnen und dann die Elemente von A^* der Länge nach aufzählen und bei gleicher Länge nach lexikografischer Ordnung. Die Umkehrabbildung von $w = (w_0, \ldots, w_l) \mapsto \sum_{i=0}^{l} (\#w_i + 1) \cdot n^i$, wobei $\#a_k = k$, ist eine solche Abbildung.

[37] Genauer: *Turing-berechenbar* oder *Turing-Maschinen-berechenbar*.
[38] Weder Name noch Bezeichnungsweise sind Standard-Notationen!

(b) Diese Bijektionen sind durch geeignete Turing-Maschinen berechenbar. Man kann also stets $A^* = \mathbb{N}$ annehmen.

Außerdem sieht man, dass eine Turing-Maschine, die eine Bijektion $\mathbb{N} \to A^*$ berechnet, zu einer Maschine umgewandelt werden kann, welche die Elemente von A^* aufzählt, d. h. nach und nach alle Elemente von A^* produziert.

(c) Es gibt eine berechenbare Bijektion $\beta_2 : \mathbb{N} \times \mathbb{N} \to \mathbb{N}$, z. B. die Abbildung $(m, n) \mapsto \frac{1}{2}(m^2 + n^2 + 2mn + 3m + n)$. Induktiv erhält man dadurch Bijektionen $\beta_k : \mathbb{N}^k \to \mathbb{N}$. Also kann man Abbildungen von $\mathbb{N}^k \to \mathbb{N}$ auf berechenbare Weise als Abbildungen $\mathbb{N} \to \mathbb{N}$ darstellen. Definition 3.3.1 überträgt sich damit auf Funktionen $\mathbb{N}^k \to \mathbb{N}$ bzw. $A_1^* \times \cdots \times A_k^* \to B^*$.[39]

These von Church: *Jede intuitiv berechenbare Funktion ist Turing-berechenbar.* (D. h. jeder andere präzise Berechbarkeitsbegriff ist entweder gleichwertig oder schwächer als Turing-Berechenbarkeit.)

Diese These ist für viele Maschinenmodelle und Programmiersprachen verifiziert und allgemein akzeptiert. Sie wird im folgenden (und in der voranstehenden Bemerkung) insofern benutzt, dass bei vielen Berechenbarkeitsfragen nur die intuitive Berechenbarkeit klar gemacht wird, und nicht konkrete Turing-Maschinen konstruiert werden.

Lemma 3.3.3 *(a) $W \subseteq A^*$ ist genau dann semi-entscheidbar, wenn es eine berechenbare Funktion $f : A^* \to A^*$ gibt mit* $\mathrm{Bild}(f) = W$.

(b) $W \subseteq A^*$ ist genau dann entscheidbar, wenn W und $A^* \setminus W$ semi-entscheidbar sind.

Beweis: (a) Identifiziere o. E. A^* mit $\mathbb{N}$.

„⇐“: Gegeben $n \in \mathbb{N}$, berechne sukzessive alle $f(m)$ für $m \in \mathbb{N}$. Falls $f(m) = n$, stoppe.

„⇒“: Betrachte die Maschine $\mathcal{M}$, die in ihrem $(\frac{l(l+1)}{2} + k)$-ten Berechnungsschritt den k-ten Berechnungsschritt für $\overline{\chi_W}(l)$ durchführt (für $l \in \mathbb{N}$ und $k = 0, \ldots, l$). Setze $f(m) = n$, falls n die m-te Zahl ist, für die die Antwort $\overline{\chi_W}(n) = 1$ erreicht wird.

(b) „⇐“: Man berechnet $\chi_W(w)$ und gibt 1 aus, falls $\chi_W(w) = 1$, und geht in eine beliebige unendliche Schleife, falls $\chi_W(w) = 0$. Dadurch berechnet man $\overline{\chi_W}$. Für $\overline{\chi_{A^* \setminus W}}$ geht es analog.

„⇒“: Man berechnet in abwechselnden Schritten $\overline{\chi_W}$ und $\overline{\chi_{A^* \setminus W}}$. Eine der beiden Berechnungen muss stoppen; falls es $\overline{\chi_W}$ ist, Ausgabe 1, andernfalls Ausgabe 0. □

Die Technik aus der Beweisrichtung „⇒“ in Teil (a), unendlich viele Berechnungen sukzessive gleichzeitig durchzuführen, nenne ich auch „diagonale Berechnung“.

Wenn $u : A^* \times A^* \to A^*$ eine berechenbare Funktion ist, dann sind natürlich für alle $w \in A^*$ auch die Funktionen $u_w : A^* \to A^*, u_w(v) = u(w, v)$ berechenbar. Analog für partielle Funktionen.

Satz 3.3.4 *(a) Es gibt keine* universelle berechenbare Funktion*, d. h. kein berechenbares $u : A^* \times A^* \to A^*$, so dass $\{u_w : A^* \to A^* \mid w \in A^*\}$ gerade die Menge aller berechenbaren Funktionen $A^* \to A^*$ ist.*

(b) Es gibt eine universelle partielle berechenbare Funktion*, d. h. ein berechenbares $u : A^* \times A^* \dashrightarrow A^*$, so dass $\{u_w : A^* \dashrightarrow A^* \mid w \in A^*\}$ gerade die Menge aller berechenbaren partiellen Funktionen $A^* \dashrightarrow A^*$ ist.*

[39] Man kann sogar eine berechenbare Bijektion $\mathbb{N}^* \to \mathbb{N}$ angeben: das leere Wort wird auf 0 abgebildet, ein $w \neq \emptyset$ auf $\beta_2(\lg(w) - 1, \beta_{\lg w}(w)) + 1$.

BEWEIS: (a) Gäbe es solch ein u, wäre auch die Diagonalfunktion $\delta : A^* \to A^*, w \mapsto u(w,w)$ berechenbar, und damit auch $\delta' : A^* \to A^*, w \mapsto u(w,w)^\frown a_1$ (anstelle von a_1 kann ein beliebiges Symbol aus A stehen). Also müsste es ein $w_0 \in A^*$ geben mit $\delta' = u_{w_0}$. Dann wäre aber $\delta'(w_0) = u(w_0,w_0)^\frown a_1 \neq u(w_0,w_0) = u_{w_0}(w_0)$.

(b) (Beweisskizze, vgl. Vorlesung „Theoretische Informatik“) Programme einer Turing-Maschine $\mathcal{C}$ können auf berechenbare Weise in A^* codiert werden. Wähle eine feste Codierung $\mathcal{C} \mapsto \ulcorner \mathcal{C} \urcorner \in A^*$.

Nun gibt es eine *universelle Turing-Maschine* $\mathcal{U}$, die als Eingabe ein Paar $(c,w) \in A^* \times A^*$ bekommt: Falls $c = \ulcorner \mathcal{C} \urcorner$, simuliert $\mathcal{U}$ den Lauf von $\mathcal{C}$ mit Eingabe w und gibt ggf. deren Ausgabe aus. In allen anderen Fällen läuft sie unendlich lange weiter. Die universelle partielle berechenbare Funktion ist die durch $\mathcal{U}$ bestimmte Funktion $f_{\mathcal{U}}$. □

Beispiel: Man kann $\mathcal{L}$-Formeln über einem endlichen Alphabet auffassen, wenn man die unendlich vielen Individuenvariablen und möglicherweise unendlich vielen Symbole in $\mathcal{L}$ auf geeignete Weise codiert. Die Menge aller $\mathcal{L}$-Formeln ist dann entscheidbar. (Dies folgt zum Beispiel aus der eindeutigen Lesbarkeit.)

Die Vollständigkeit des Hilbert-Kalküls zeigt:

Satz 3.3.5 *Die Menge aller allgemeingültigen $\mathcal{L}$-Formeln ist semi-entscheidbar.*

Satz 3.3.6 (Halteproblem, Turing 1936) *Die Menge*

$$\{(c,w) \in A^* \times A^* \mid c = \ulcorner \mathcal{C} \urcorner \text{ und die Turing-Maschine } \mathcal{C} \text{ stoppt bei Eingabe } w\}$$

ist semi-entscheibar, aber nicht entscheidbar.

BEWEIS: Semi-Entscheidbarkeit: Falls $c = \ulcorner \mathcal{C} \urcorner$, simuliere den Lauf von $\mathcal{C}$ mit Eingabe w und stoppe, falls $\mathcal{C}$ stoppt.

Unentscheidbarkeit: Falls das Halteproblem entscheidbar wäre, könnte man eine universelle berechenbare Funtkion u konstruieren: Bei Eingabe (c,w), prüfe zunächst, ob $c = \ulcorner \mathcal{C} \urcorner$. Falls ja, prüfe ob $\mathcal{C}$ mit Eingabe w stoppt. Falls auch dies, berechnet $f_{\mathcal{C}}(w)$, gib dies als Ausgabe aus und stoppe. In allen andern Fällen stoppe und gib beliebigen festen Wert (z. B. a_1) aus. □

Folgerung 3.3.7 *Es gibt semi-entscheidbare, nicht entscheidbare Teilmengen von A^*.*

Eine unendliche Menge M heißt *abzählbar*, falls es eine Bijektion $\mathbb{N} \to M$ gibt, und anderfalls *überabzählbar*.[40]

Satz 3.3.8 (Cantor 1877) *Die Menge* $\mathrm{Abb}(\mathbb{N}, \{0,1\})$ *der* 0-1-*Folgen ist überabzählbar; ebenso die Potenzmengen* $\mathrm{Pot}(\mathbb{N})$ *und* $\mathrm{Pot}(\mathbb{A}^*)$ *für jedes endliche Alphabet A.*

BEWEIS: Nach Bemerkung 3.3.2 gibt es eine Bijektion $\beta : \mathbb{N} \to A^*$, die sich zu einer Bijektion $\tilde{\beta} : \mathrm{Pot}(\mathbb{N}) \to \mathrm{Pot}(\mathbb{A}^*)$ fortsetzt, nämlich $\tilde{\beta}(M) = \{\beta(n) \mid m \in M\}$ für $M \subseteq \mathbb{N}$. Außerdem steht $\mathrm{Pot}(\mathbb{N})$ in Bijektion mit der Menge der 0-1-Folgen, indem man jeder Teilmenge $M \subseteq \mathbb{N}$ die charakteristische Funktion χ_M zuordnet.

[40] Man darf *abzählbar* nicht mit *aufzählbar* verwechseln! Jede aufzählbare Menge ist abzählbar, aber nicht umgelehrt.

Es reicht also, die Überabzählbarkeit der Menge der 0-1-Folgen zu zeigen, was mit *Cantors Diagonalargument* funktioniert: Angenommen $\beta : \mathbb{N} \to \mathrm{Abb}(\mathbb{N}, \{0,1\})$ wäre eine Bijektion. Dann betrachtet man die 0-1-Folge α mit $\alpha(n) = 1 - \beta(n)(n)$. Nach Annahme gibt es ein $n_0 \in \mathbb{N}$ mit $\alpha = \beta(n_0)$. An der Stelle n_0 ist aber $\beta(n_0)(n_0) \neq \alpha(n_0)$: Widerspruch. □

Der Beweis der Nicht-Existenz einer universellen berechenbaren Funktion benutzt genau dieses Diagonalargument von Cantor!

Da jede Turing-Maschine durch ihren Code (also eine endliche Zeichenfolge über einem endlichen Alphabet A) beschreibbar und A^* abzählbar ist, gibt es also nur abzählbar viele (semi-) entscheidbare Mengen von $\mathbb{N}$. Überabzählbar viele Teilmengen von $\mathbb{N}$ sind also nicht (semi-) entscheidbar. Da es aber auch nur abzählbar viele Beschreibungen von Teilmengen von $\mathbb{N}$ gibt (durch einen endlichen Text, ggf. mit Formeln!), sind die meisten dieser nicht-(semi-)entscheidbaren Mengen auch gar nicht irgendwie beschreibbar. Die Menge

$$\big\{(c,w) \in A^* \times A^* \;\big|\; c = \ulcorner \mathcal{C} \urcorner \text{ und die Turing-Maschine } \mathcal{C} \text{ stoppt } \textit{nicht} \text{ bei Eingabe } w\big\}$$

ist ein Beispiel einer beschreibbaren, aber nicht semi-entscheidbaren Menge. Denn da ihr Komplement – die Menge aus dem Halteproblem – semi-entscheidbar ist, wäre dieses (d. h. das Halteproblem) nach Lemma 3.3.3 (b) auch entscheidbar.

Satz 3.3.9 (Unentscheidbarkeit der Prädikatenlogik, Gödel 1931) *Falls $\mathcal{L}$ mindestens ein zweistelliges Relationszeichen oder ein einstelliges Funktionszeichen enthält, ist*

$$\{\varphi \mid \varphi \textit{ allgemeingültige } \mathcal{L}\textit{-Aussage}\}$$

nicht entscheidbar.

Beweis: Da φ genau dann allgemeingültig ist, wenn $\neg\varphi$ nicht erfüllbar ist, kann man ebensogut zeigen, dass die Menge der erfüllbaren $\mathcal{L}$-Aussagen nicht entscheidbar ist.

Der Beweis führt die Frage auf das Halteproblem zurück: Für jede Turing-Maschine $\mathcal{C}$ und jede Eingabe w wird auf berechenbare Weise eine $\mathcal{L}$-Formel $\sigma_{\mathcal{C},w}$ konstruiert, die genau dann erfüllbar ist, wenn $\mathcal{C}$ bei Eingabe w stoppt. Wenn die Menge der allgemeingültigen $\mathcal{L}$-Aussagen entscheidbar wäre, könnte man also auch das Halteproblem entscheiden.

Ohne Einschränkung sei $A = \{0,1\}$ das Alphabet, über dem die Turing-Maschine arbeitet. Die Zellen des Bandes kann man sich durch $\mathbb{Z}$ indiziert vorstellen (mit Startposition 0), die Arbeitsschritte (oder Zeitpunkte) sind durch $\mathbb{N}$ indiziert (ebenfalls mit Startwert 0). Der Kürze der Beschreibung wegen werden die gleichen Variablen sowohl für Zellen als auch für Zeitpunkte verwende, und der Einfachheit halber wird eine von $\mathcal{C}$ abhängige Sprache $\mathcal{L} = \mathcal{L}_{\mathcal{C}}$ benutzt. Man kann sich dann überlegen, dass man diese Sprache z. B. durch ein einziges zweistelliges Relationszeichen codieren kann.

$\mathcal{L}$ besteht aus

- einem zweistelligen Relationszeichen $<$ für die Anordnung der Zellen und der Zeitpunkte;
- zwei einstellige Funktionszeichen N und N^{-1} für die nachfolgende Zelle/den nachfolgenden Zeitpunkt bzw. die vorangehende Zelle/den vorangehenden Zeitpunkt;
- einer Konstanten 0 für die Startposition und den Startzeitpunkt;
- einem zweistelligen Relationszeichen K, wobei Kzt die Position z des Lese-Schreib-Kopfes zum Zeitpunkt t angibt;

- zwei zweistellige Relationszeichen A_0 und A_1, wobei A_izt angibt, dass zum Zeitpunkt t das Zeichen i in Zelle z steht;
- für jeden Zustand Z_j der Turing-Maschine ein einstelliges Relationszeichen Z_j, wobei Z_jt angibt, dass sich die Maschine zum Zeitpunkt t im Zustand Z_j befindet.[41]

Nun drückt man in einer $\mathcal{L}$-Aussage τ die Grundgegebenheiten einer Turing-Maschine aus, d. h.

- $<$ beschreibt eine diskrete lineare Ordnung ohne Endpunkte, was bedeutet, dass jedes Element z einen unmittelbaren Nachfolger Nz und einen unmittelbaren Vorgänger $N^{-1}z$ hat;
- in jeder Zelle steht immer maximal ein Symbol: $\forall z\forall t\,\neg(A_0zt \wedge A_1zt)$;
- die Maschine ist zu jedem Zeitpunkt t in genau einem Zustand und zu Beginn im Startzustand: Z_00;
- der Kopf steht immer in genau einer Zelle und zu Beginn in der Startposition: $K00$.

In einer weiteren $\mathcal{L}$-Aussage $\pi_{\mathcal{C}}$ wird das Programm von $\mathcal{C}$ formuliert. Sie besteht aus einer Konjunktion von Aussagen der Form

$$\forall z\forall t(((t \doteq 0 \vee 0 < t) \wedge Kzt \wedge Z_jt \wedge A_izt) \to (Kz'Nt \wedge Z_{j'}Nt \wedge A_{i'}zNt)),$$

dabei sind z', j' und i' durch das Programm festgelegt; z' ist dabei z oder Nz oder $N^{-1}z$.

Schließlich drückt man in einer $\mathcal{L}$-Aussage ε_w die Eingabe $w = (w_0, \dots, w_k)$ aus:

$$((A_{w_0}00 \wedge A_{w_1}N00 \wedge \dots \wedge A_{w_k}N^k00) \wedge \forall z((z < 0 \vee N^k0 < z) \to (\neg A_0z0 \wedge \neg A_1z0)))$$

(dabei bedeutet N^k, dass das Funktionszeichen N k-mal hintereinander steht).

Wenn die Turing-Maschine $\mathcal{C}$ nun bei Eingabe w nicht stoppt, ergibt sich aus dem Lauf der Maschine ein Modell der $\mathcal{L}$-Aussage

$$\sigma_{\mathcal{C},w} := (\tau \wedge \pi_{\mathcal{C}} \wedge \varepsilon_w \wedge \forall t(0 < t \to \neg Z_{\text{end}}t)) :$$

Universum des Modells ist $\mathbb{Z}$ und man interpretiert die Zeichen in $\mathcal{L}$ so, wie intendiert. (Für negative Zeitpunkte kann man irgendwelche mit den allgemeinen Bedingungen konsistente Konfigurationen wählen).

Ist umgekehrt die $\mathcal{L}$-Aussage $\sigma_{\mathcal{C},w}$ erfüllbar, dann hat sie ein Modell $\mathfrak{M}$. Die Menge M_0, die aus $0^{\mathfrak{M}}$ und seinen unmittelbaren Nachfolgern $N\dots N0^{\mathfrak{M}}$ und Vorgängern $N^{-1}\dots N^{-1}0^{\mathfrak{M}}$ besteht, ist Universum einer (als Ordnung) zu $\mathbb{Z}$ isomorphen Unterstruktur. Auf M_0 beschreibt $\sigma_{\mathcal{C},w}$ nun den Lauf der Turing-Maschine $\mathcal{C}$ bei Eingabe w: In Zelle z steht zum Zeitpunkt t das Symbol i genau dann, wenn $\sigma_{\mathcal{C},w} \vdash A_izt$, etc. In diesem Lauf wird der Endzustand nicht erreicht. □

Bemerkung 3.3.10 Wenn $\mathcal{L}$ nur Prädikate und Konstanten enthält, ist es entscheidbar, ob eine $\mathcal{L}$-Aussage allgemeingültig ist oder nicht.

3.4 Rekursive Funktionen und Arithmetik

Definition 3.4.1 (a) *Eine Funktion $f : \mathbb{N}^k \to \mathbb{N}$ für $k \in \mathbb{N}$ heißt* primitiv rekursiv, *wenn sie entweder eine der folgenden Grundfunktionen ist:*

- *die konstante Funktion $0 : \mathbb{N}^0 \to \mathbb{N}$;*

[41] Um eine von $\mathcal{C}$ unabhängige Sprache zu erhalten, kann man stattdessen ein zweistelliges Relationszeichen Z nehmen und durch Ztj ausdrücken, dass sich die Maschine zum Zeitpunkt t im Zustand Z_j befindet, wobei j das j-te, in der Anordung $<$ auf 0 folgende Element ist.

- *die Nachfolgerfunktion* $N : \mathbb{N} \to \mathbb{N}$, $n \mapsto n+1$;
- *eine Projektionsfunktion* $\pi_i^k : \mathbb{N}^k \to \mathbb{N}$, $(n_1, \ldots, n_k) \mapsto n_i$;

oder sich daraus durch endliche Anwendung folgender Regeln daraus ergibt:

- (Komposition:) *Wenn* $f_1, \ldots, f_l : \mathbb{N}^k \to \mathbb{N}$ *und* $g : \mathbb{N}^l \to N$ *primitiv rekursiv sind, dann auch die Funktion* $h : \mathbb{N}^k \to \mathbb{N}$ *mit*

$$h(n_1, \ldots, n_k) := g(f_1(n_1, \ldots, n_k), \ldots, f_l(n_1, \ldots, n_k)).$$

- (Primitive Rekursion:) *Wenn* $f : \mathbb{N}^k \to \mathbb{N}$ *und* $g : \mathbb{N}^{k+1} \to \mathbb{N}$ *primitiv rekursiv sind, dann auch die Funktion* $h : \mathbb{N}^{k+1} \to \mathbb{N}$ *mit*

$$\begin{aligned} h(n_1, \ldots, n_k, 0) &:= f(n_1, \ldots, n_k) \\ h(n_1, \ldots, n_k, n+1) &:= g(h(n_1, \ldots, n_k, n), n_1, \ldots, n_k). \end{aligned}$$

(b) *Eine partielle Funktion* $f : \mathbb{N}^k \dashrightarrow \mathbb{N}$ *heißt* rekursiv*, wenn zusätzlich folgende Regel zugelassen ist:*

- (μ-Rekursion) *Wenn* $f : \mathbb{N}^{k+1} \dashrightarrow \mathbb{N}$ *rekursiv ist, dann auch die Funktion* $h : \mathbb{N}^k \dashrightarrow \mathbb{N}$ *mit*

$$h(n_1, \ldots, n_k) := \begin{cases} \mu n\, f(n_1, \ldots, n_k, n) = 0 & \text{falls es existiert} \\ \text{unbestimmt} & \text{sonst} \end{cases}$$

Dabei steht $\mu n\, f(n_1, \ldots, n_k, n) = 0$ *für das kleinste* n *mit* $f(n_1, \ldots, n_k, n) = 0$.

(c) *Eine Teilmenge von* $\mathbb{N}^k$ *heißt* (primitiv) rekursiv*, wenn ihre charakteristische Funktion (primitiv) rekursiv ist. Die Teilmenge heißt* rekursiv aufzählbar*, wenn sie Definitionsbereich einer partiellen rekursiven Funktion ist.* [42]

Satz 3.4.2 *Die rekursiven (partiellen) Funktionen sind genau die berechenbaren (partiellen) Funktionen.*

Beweisidee: Die Grundfunktionen lassen sich durch Turing-Maschinen berechnen, und die verschiedenen Kompositions- und Rekursionsschritte lassen sich mit Turing-Maschinen nachbilden. Umgekehrt kann man zeigen, dass sich die Arbeitsweise von Turing-Maschinen durch rekursive Funktionen beschreiben lassen. □

Folgerung 3.4.3 *Die rekursiv aufzählbaren Teilmengen von* $\mathbb{N}^k$ *sind genau die semi-entscheidbaren Teilmengen von* $\mathbb{N}^k$.

Im Folgenden soll $\mathbb{N}$ nicht nur die *Menge* der natürlichen Zahlen bezeichnen, sondern die $\mathcal{L}_{\mathbb{N}}$-*Struktur* darauf, wobei $\mathcal{L}_{\mathbb{N}} = \{<, +, \cdot, 0, 1\}$ und die Zeichen auf die übliche Art und Weise interpretiert werden. Diese Struktur heißt auch die „Arithmetik“.

Außerdem sei $\tau_0 = 0$ und τ_n der Term $\underbrace{(\cdot(1+1)+\ldots)+1}_{n \text{ mal}}$ für $n \geqslant 1$. Es ist also $\tau_n^{\mathbb{N}} = n$.

Definition 3.4.4 *Für eine* $\mathcal{L}$*-Struktur* $\mathfrak{M}$ *ist die* $\mathcal{L}$-Theorie *von* $\mathfrak{M}$ *die Formelmenge*

$$\mathrm{Th}(\mathfrak{M}) := \{\varphi \mid \varphi \text{ } \mathcal{L}\text{-Aussage mit } \mathfrak{M} \vDash \varphi\}.$$

[42] Äquivalent: Wenn sie die leere Menge ist oder das Bild einer rekursiven Funktion ist.

Bemerkung 3.4.5 Wenn die Theorie einer $\mathcal{L}$-Struktur $\mathfrak{M}$ semi-entscheidbar ist, dann ist sie auch entscheidbar: Denn das Komplement von $\mathrm{Th}(\mathfrak{M})$ ist dann ebenfalls semi-entscheidbar, da die Menge der $\mathcal{L}$-Formeln entscheidbar ist und

$$\varphi \notin \mathrm{Th}(\mathfrak{M}) \iff \neg\varphi \in \mathrm{Th}(\mathfrak{M}).$$

Lemma 3.4.6 *Alle rekursiven partiellen Funktionen sind durch $\mathcal{L}_\mathbb{N}$-Formeln definierbar. Es gibt einen berechenbaren Übergang von einer Turing-Maschine $\mathcal{C}$ zu einer $\mathcal{L}_\mathbb{N}$-Formel $\varphi_\mathcal{C}(v_0, v_1)$, die $f_\mathcal{C}$ definiert, d. h. $f_\mathcal{C}(m) = n \iff \mathbb{N} \vDash \varphi_\mathcal{C}[m, n]$ $\left(\iff \mathbb{N} \vDash \varphi_\mathcal{C} \frac{\tau_m}{v_0} \frac{\tau_n}{v_1} \right)$.*

Beweisskizze: Die Formeln kann man unmittelbar aus der Definition rekursiver Funktionen gewinnen, da sowohl die Grundfunktionen definierbar sind als auch Komposition und Rekursionen in $\mathcal{L}_\mathbb{N}$ ausdrückbar sind. Der berechenbare Übergang ergibt sich aus dem Beweis von Satz 3.4.2, der konstruktiv ist. □

Satz 3.4.7 (Erster Unvollständigkeitssatz von Gödel für die Arithmetik)
(a) $\mathrm{Th}(\mathbb{N})$ *ist nicht entscheidbar.*

(b) *Wenn $T \subseteq \mathrm{Th}(\mathbb{N})$ semi-entscheidbar ist, dann ist $T^\vdash := \{\varphi \mid T \vdash \varphi\}$ nicht vollständig, d. h. es gibt $\mathcal{L}_\mathbb{N}$-Aussagen ψ so, dass weder $T \vdash \psi$ noch $T \vdash \neg\psi$.*

Da eine der beiden Formeln ψ und $\neg\psi$ in $\mathbb{N}$ gelten muss, wird die Version (b) oft durch den Slogan „*Es gibt in $\mathbb{N}$ wahre Sätze, die nicht beweisbar sind*“ wiedergegeben.

Beweis: (a) Sei $\mathcal{C}$ eine Turing-Maschine, die die partielle Funktion $f_\mathcal{C} : \mathbb{N} \dashrightarrow \mathbb{N}$ berechnet. Ohne Einschränkung kann man annehmen, dass $\mathcal{C}$ genau dann bei Eingabe n stoppt, wenn $f_\mathcal{C}(n)$ definiert ist (also dass der Fall, dass $\mathcal{C}$ stoppt, aber keine vernünftige Ausgabe liefert, nicht eintritt). Dann gilt für $\varphi_\mathcal{C}[m, n]$ wie oben:

$$\mathbb{N} \vDash \exists v_1\, \varphi_\mathcal{C} \frac{\tau_n}{v_0} \iff f_\mathcal{C}(n) \text{ ist definiert} \iff \mathcal{C} \text{ stoppt bei Eingabe } n$$

Also würde aus der Entscheidbarkeit von $\mathrm{Th}(\mathbb{N})$ die Entscheidbarkeit des Halteproblems folgen.

(b) Aus der Semi-Entscheidbarkeit von T folgt analog zu Satz 3.3.5 die Semi-Entscheidbarkeit von $T^\vdash$. Also ist $T^\vdash \neq \mathrm{Th}(\mathbb{N})$, da $\mathrm{Th}(\mathbb{N})$ nach Bemerkung 3.4.5 und Teil (a) nicht semi-entscheidbar ist. □

Zum Abschluss sei ein schwerer Satz erwähnt, der zeigt, dass die rekursiv aufzählbaren Mengen in $\mathbb{N}$ durch Formeln einfacher Art definierbar sind:

Satz 3.4.8 (Satz von Matiyasevich[43], 1970) *$R \subseteq \mathbb{N}^k$ ist genau dann rekursiv aufzählbar, wenn es ein $l \in \mathbb{N}$ und Polynome $P, Q \in \mathbb{N}[X_1, \ldots, X_k]$ gibt mit*

$$R = \big\{(n_1, \ldots, n_k) \in \mathbb{N} \;\big|\; \text{es gibt } m_1, \ldots, m_l \in \mathbb{N} \text{ mit } P(n_1, \ldots, n_k, m_1, \ldots, m_l) = Q(n_1, \ldots, n_k, m_1, \ldots, m_l)\big\}$$

d. h. wenn R durch eine Formel

$$\exists v_{k+1} \ldots \exists v_{k+l}\; \tau_1(v_1, \ldots, v_k, v_{k+1}, \ldots, v_{l+k}) \doteq \tau_2(v_1, \ldots, v_k, v_{k+1}, \ldots, v_{l+k})$$

mit $\mathcal{L}_\mathbb{N}$-Termen τ_1, τ_2 definiert ist.

44

Beweishälfte: Einfach zu sehen ist, dass durch solche Formeln definierte Relationen rekursiv aufzählbar, d. h. semi-entscheidbar, sind: Man betrachtet eine Aufzählung von $\mathbb{N}^l$ und berechnet für gegebenes $(n_1, \ldots, n_k)$ und alle $(m_1, \ldots, m_l)$ „diagonal“ aus, ob $P(n_1, \ldots, n_k, m_1, \ldots, m_l) = Q(n_1, \ldots, n_k, m_1, \ldots, m_l)$. □

Die andere Richtung ist ein schweres mathematisches Theorem, und es ist auch überhaupt nicht einfach, für offensichtlich entscheidbare Mengen wie etwa die Menge aller Primzahlen die Polynome zu finden!

[44] Aufgrund entscheidender Vorarbeiten anderer auch *Satz von Davis, Putnam, Robinson und Matiyasevich* genannt, kurz *DPRM-Theorem* oder *MRDP-Theorem*.

4 Anhang

Literatur

Es gibt eine Vielzahl von Büchern zur (mathematischen) Logik, von denen ich hier nur diejenigen aufführe, mit denen ich gearbeitet habe. Es gibt in der Logik leider weder eine Standardterminologie noch Standardnotationen. Besonders ältere Bücher sind daher manchmal sehr gewöhnungsbedürftig, aber auch neuere sind manchmal schwer zu verstehen, wenn man sie mittendrin aufschlägt. Warnen möchte ich vor der Vielzahl von Einführungen in die formalisierte Logik für Nicht-Mathematiker und Nicht-Informatiekr (typischerweise für Philosophen). Diese Bücher haben oft einen anderen Fokus, sind manchmal ideologisch und teilweise schlichtweg schlecht.

- René Cori & Daniel Lascar: *Logique mathématique*, 2. Auflage, Masson, Paris 1994.
  (Gibt es auch in englischer Übersetzung.)
- Heinz-Dieter Ebbinghaus, Jörg Flum & Wolfgang Thomas: *Einführung in die mathematische Logik*, 5. Auflage, Spektrum Verlag, Berlin 2007.
- Uwe Schöning: *Logik für Informatiker*, 5. Auflage, Spektrum Verlag, Berlin 2005.
- Martin Ziegler: *Mathematische Logik*, Birkhäuser, Basel 2010.

Zu der Vorlesung „Logik für Studierende der Informatik“ gibt es von Martin Ziegler ein Kurzskript vom WS 2006/07 (letzte Version WS 2012/13) und von Heike Mildenberger ein ausführliches Skript vom WS 2011/12.

Nicht abgedeckt in den obigen Büchern und Skripten ist der Abschnitt über intuitionistische Logik. Die hier vorgestellte Semantik versteht man am besten über die Interpretation der intuitionistische Aussagenlogik in der Modallogik S4. Vorschläge für weitere Lektüre in diese Richtung sind:

- Melvin Fitting: *Intuitionistic Logic, Model Theory, and Forcing*, North-Holland, Amsterdam 1969.
- Patrick Blackburn, Maarten de Rijke & Yde Venema: *Modal Logic*, Cambridge University Press 2001.

Namenspaten und wichtige Personen

George Boole (1815–1864): formale Aussagenlogik; Boole'sche Algebra
Luitzen Egbertus Jan Brouwer (1881–1966): Begründer des Intuitionismus
Georg Cantor (1845–1918): Begründer der Mengenlehre; Diagonalargument
Alonzo Church (1903–1995): Church'sche These
Stephen Cook (∗1939): Satz von Cook und Levin
Haskell Brooks Curry (1900–1982): Currying, Curry-Howard-Korrspondenz (und Programmiersprache Haskell)
Martin Davis (∗1928): Satz von Davis, Putnam, Robinson und Matiyasevich
Gottlob Frege (1848–1925): formale Prädiaktenlogik; aus seinem „Behauptungsstrich“ ist das Zeichen $\vdash$ entstanden
Gerhard Gentzen (1909–1945): Gentzen-Kalküle
Kurt Gödel (1906–1978): Vollständigkeitssatz, Unvollständigkeitssatz
Helmut Hasse (1898–1979): Hasse-Diagramme
Leon Henkin (1921–2006): Henkin-Modelle, -Konstanten und -Axiome
Jacques Herbrand (1908–1931): Herbrand-Normalform, Satz von Herbrand
Arend Heyting (1898–1980): formaler Intuitionismus; Heyting-Algebren
David Hilbert (1862–1943): Hilbert-Kalküle
Alfred Horn (1918–2001): Horn-Formeln, Horn-Klauseln
William Howard (∗1926): Curry-Howard-Korrespondenz
Jean-Louis Krivine (∗1939): Krivines Hut
Leonid Levin (∗1948): Satz von Cook und Levin
Adolf Lindenbaum (1904–1941): Tarski-Lindenbaum-Algebra
Yuri Matiyasevich (∗1947): Satz von Davis, Putnam, Robinson und Matiyasevich
Augustus de Morgan (1806–1871): Regeln von de Morgan
Charles Sanders Peirce (1839–1914): formale Prädiaktenlogik; Peirce-Funktion: anderer Name für den NOR-Junktor
Hilary Putnam (1926–2016): Satz von Davis, Putnam, Robinson und Matiyasevich
Willard Van Orman Quine (1908–2000): Methode von Quine
John Alan Robinson (1930–2016): Resolution und Unifikation
Julia Robinson (1919–1985): Satz von Davis, Putnam, Robinson und Matiyasevich
Henry Maurice Sheffer (1882–1964): Sheffer-Strich: anderer Name für den NAND-Junktor
Albert Thoralf Skolem (1887–1963): Skolem-Funktionen und -Normalform, Skolemisierung
Marshall Stone (1903–1989): Darstellungssatz von Stone
Alfred Tarski (1901–1983): Tarski-Lindenbaum-Algebra
Alan Turing (1912–1954): Turing-Maschine
John Venn (1834–1923): Venn-Diagramme
Martin Ziegler (∗1948): Symbol $\doteq$ für das Gleichheitszeichen in der Prädikatenlogik

Farbenlehre

Formalisierte Logik innerhalb der Mathematik oder Informatik untersucht formale Sprachen mit formalen Methoden. Man muss daher sorgsam das Untersuchungsobjekt (die sogenannte *Objektsprache*) von der Sprache, in der die Untersuchung stattfindet (die sogenannte *Metasprache*), unterscheiden, gerade weil auf beiden Ebenene die gleichen Konzepte auftreten. Teilweise geschieht dies durch unterschiedliche Zeichen (z. B. $\leftrightarrow$ und $\doteq$ in der Objektsprache vs. $\Leftrightarrow$ und $=$ in der Metasprache). Ich habe in diesem Skript versucht, die verschiedenen Ebenen und Objekte durch Benutzung von Farben transparenter zu machen:

- Blau steht für Symbole der Objektsprache (als einzelne Symbole).
- Violett steht für objektsprachliche Terme.
- Rot steht für aussagenlogische oder prädikatenlogische Formeln.
- Grün steht für Auswertungen von Termen oder Formeln in Strukturen.
- Schwarz steht für Belegungen.

Dabei ist die Trennung nicht immer ganz klar, da z. B. Terme in Formeln vorkommen und z. B. Aussagenvariablen sowohl Symbole als auch Formeln sind und nicht immer klar ist, in welcher Funktion sie eher angesprochen sind. Ich habe mir dafür folgende Regeln gegeben:

- Die Farbe des umfassenderen Objektes hat Vorrang. (Symbole und Terme in Formeln sind also rot, Auswertungen in Belegungen schwarz, aber Belegungen in Auswertungen grün.)

Blau steht insbesondere für

- Variablen für objektsprachlicher Symbole;
- Zeichenfolgen objektsprachlicher Symbole, die keine Terme oder Formeln sind;
- (einzelne) Vorkommen von Aussagen- oder Individuenvariablen in Termen oder Formeln;

aber nicht für

- nicht-objektsprachliche Symbole (z. B. Wahrheitswerte);
- Variablen für Zeichenfolgen objektsprachlicher Symbole, die keine Terme oder Formeln sind;
- Symbole in Namen von Regeln (z. B. „$\exists$-Einführung“).

Violett steht insbesondere für

- Abkürzungen und Variablen für Terme;
- Individuenvariablen als Elemente des Definitionsbereichs einer Belegung;
- Individuenvariablen und Terme in Substitutionen (außer in Belegungen);
- freie Variablen eineer Formel und Variablen, die „frei für“ sind, sonfern nicht einzelne Vorkommen gemeint sind.

Rot steht insbesondere für

- Abkürzungen und Variablen für Formeln;
- Aussagenvariablen als Elemente des Definitionsbereichs einer Belegung.

Grün steht insbesondere für

- Wahrheitswertfunktionen von Junktoren;
- Auswertung einer aussagenlogischen Formel unter einer Belegung (klassisch) bzw. in einem Zustand (intuitionistisch);
- Interpretation von Funktions- und Relationszeichen in einer $\mathcal{L}$-Struktur (durch Funktionen und Relationen);
- Interpretationen von Termen in einer $\mathcal{L}$-Struktur;

aber nicht für

- Elemente der Auswertungsstrukturen, wenn sie nicht als Auswertung einer Formel oder eines Terms gegeben sind.

Schlussbemerkung und „Plagiats-Disclaimer“

Dieses Skript ist zu der im Wintersemester 2016/17 an der Albert-Ludwigs-Universität in Freiburg gehaltenen Vorlesung „Logik für Studierende der Informatik“ entstanden und wurde im Wintersemester 2017/18 überarbeitet. Mit einigen Einschränkungen gibt das Skript den Vorlesungsstoff des Wintersemesters 2016/17 wieder:

- Terminologie und Notationen habe ich in einigen Punkten überarbeitet.
- Die Tableau-Methode wurde in der Vorlesungs nur angedeutet; die Methode von Quine kam gar nicht vor.
- Ich habe im Skript einige, zum Teil ausführliche Beispiele ergänzt (etwa zur Resolution oder zur Herbrand'schen Methode).
- In der Vorlesung kamen die Abschnitte 4.1 und 4.2 zwischen 2.5 und 2.6.

Das Skript ist nach in der Mathematik gängiger Vorgehensweise angefertigt. Dies bedeutet, dass es keinen Anspruch auf eine eigene wissenschaftliche Leistung erhebt und keine eigenen Ergebnisse wiedergibt, sondern die Ergebnisse anderer darstellt (die im Wesentlichen zwischen 1850 und 1975 entstanden sind). Die konkrete Darstellung und Ausformulierung stammt allerdings von mir (inklusive aller Fehler).

Da Mathematik weitgehend ahistorisch betrieben wird, ließe sich nur mühsam zurückverfolgen, von wem welche Fragestellungen, Begriffe, Sätze, Beweise oder Beweistechniken stammen. Zu einem großen Teil muss man die Entwicklung eines mathematischen Teilgebiets auch als ein Gemeinschaftswerk einer mathematischen *Community* verstehen. Ich habe mich nicht um eine historische Darstellung bemüht; Zuweisungen von Sätzen oder Beweisen zu Mathematikern folgen der Überlieferung und müssen nicht immer historisch exakt sein.

Der Aufbau der Vorlesung folgt der von Martin Ziegler im Wintersemester 2012/13 gehaltenen Vorlesung. Meine Darstellung ist insgesamt natürlich stark beeinflusst von den Logik-Vorlesungen, die ich bei Jacques Stern, Jean-Louis-Krivine und Ramez Labib-Sami in Paris gehört bzw. bei Martin Ziegler in Freiburg als Assistent betreut habe, sowie von den Skripten und Büchern von Kollegen, die im Literaturverzeichnis angegeben sind. Diese verschiedenen Einflüsse sind nicht zu trennen und können daher nicht einzeln dargelegt werden. Der Beweis des Vollständigkeitssatzes in Abschnitt 3.5 ist allerdings im Wesentlichen ohne Änderungen dem Buch von Martin Ziegler entnommen (bis auf die abschließende Konstruktion des Modells).

Korrekturen verdanke ich Andreas Claessens sowie unzähligen meist anonymen Informatikstudenten. Über die Mitteilung verbliebener Fehler bin ich dankbar.