Grundlagen der Logik in der Informatik

Vorlesungsskript Wintersemester 2015/16
Friedrich-Alexander-Universität Erlangen-Nürnberg
Department Informatik, Lehrstuhl 8 (Theoretische Informatik)

Aufbauend auf einer Mitschrift der Vorlesung von Lutz Schröder im Sommersemester 2012 von

Johannes Schilling (dario@zerties.org)
Dominik Paulus (dominik@d-paulus.de)
Ulrich Rabenstein (ulrich.rabenstein@studium.uni-erlangen.de)
Tobias Polzer

überarbeitet von Lutz Schröder, mit weiteren Beiträgen von

Andreas Schieb

Inhaltsverzeichnis

0.1 Literatur

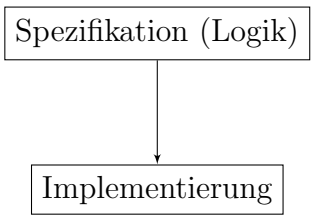
U. Schöning Logik für Informatiker, Spektrum akademischer Verlag, 2000

J. Barwise, J. Etchemendy Language, Proof & Logic, CSLI, 2000

M. Huth, M. Ryan Logic in Computer Science, CUP, 2000

0.2 Logik in der Informatik

- Logik als Problem für Informatiker
  - SAT
  - Automatisches/Halbautomatisches Theorembeweisen
- Logik als Programmierparadigma
  - Prolog
  - Mercury
- Logik als Abfrageformalismus
  - SQL
  - Datalog
- Logik als (Wissens-) Repräsentationsformalismus
  - Ontologien
  - Semantic Web
  - OWL
- Logik als Entwicklungsmethode



1 Aussagenlogik

Redet über atomare Aussagen A, B, C, ... ohne Rücksicht auf deren innere Struktur.

$$(\text{z. B. EsRegnet} \to \text{HabeSchirm} \vee \text{WerdeNass})$$

und deren Wahrheitswerte, hier klassisch: $\{\underbrace{\bot}_{\text{falsch}}, \underbrace{\top}_{\text{wahr}}\}$

1.0.1 Grundelemente einer Logik

- *Syntax*: „Was kann ich hinschreiben?“
- *Semantik*: „Was bedeutet das?“
- *Beweismethoden*: „Wie ziehe ich daraus Schlüsse?“

2 Induktion

Induktion ist das Prinzip der Reduktion einer Aussage über ein Objekt auf gleichartige Aussagen über „einfachere“ Objekte in einem jeweils geeigneten Sinn. Wenn Objekte nur endlich oft „einfacher werden“ können, erreicht man so nach endlich vielen Reduktionen ein Objekt, das nicht mehr einfacher werden kann; wenn man außerdem für solche Objekte die entsprechende Aussage beweisen kann („Induktionsanfang“), hat man die Aussage für alle Objekte bewiesen. Das Prinzip ist das gleiche wie bei der Programmierung rekursiver Funktionen – dort ruft man (jedenfalls dann, wenn man an Terminierung interessiert ist) eine Funktion rekursiv mit Argumenten auf, die „einfacher“ als das ursprüngliche Argument sind, und hat Basisfälle, in denen keine rekursiven Aufrufe stattfinden.

Wir fassen kurz die wesentlichen im weiteren benötigten Induktionsprinzipien zusammen. Wir verweisen auch auf den auf der Veranstaltungshomepage verfügbaren Text von Thomas Voß.

2.1 Induktion über natürliche Zahlen

Das vermutlich bekannteste Induktionsprinzip ist die Induktion über natürliche Zahlen. In der einfachsten Form lautet das Prinzip wie folgt. Wenn eine Aussage $P(n)$ über natürliche Zahlen n die Eigenschaften

- *Induktionsanfang*: P gilt für die 0 ($P(0)$) und
- *Induktionsschritt*: für jede natürliche Zahl n folgt $P(n+1)$ aus $P(n)$ ($\forall n \in \mathbb{N}.\, (P(n) \implies P(n+1)$; man bezeichnet hier $P(n)$ als die *Induktionsvoraussetzung*)

erfüllt, dann gilt P für jede natürliche Zahl ($\forall n \in \mathbb{N}.\, P(n)$).

Als Beispiel diene hier folgende einfache Identität:

$$\sum_{i=1}^{n} (2i - 1) = n^2.$$

Zum besseren Abgleich mit dem allgemeinen Induktionsprinzip bezeichnen wir diese Aussage mit $P(n)$. Man beweist $\forall n.\, P(n)$ durch Induktion über n:

- Induktionsanfang: Es gilt $\sum_{i=1}^{0}(2i-1) = 0 = 0^2$ (also $P(0)$).
- Induktionsschritt: Sei $n \in \mathbb{N}$, so dass $\sum_{i=1}^{n}(2i-1) = n^2$ (also $P(n)$); zu zeigen ist dann $P(n+1)$, also $\sum_{i=1}^{n+1}(2i-1) = (n+1)^2$. Man rechnet wie folgt:

$$\begin{aligned}\sum_{i=1}^{n+1}(2i-1) &= \sum_{i=1}^{n}(2i-1) + 2(n+1) - 1\\ &\overset{IV}{=} n^2 + 2(n+1) - 1\\ &= n^2 + 2n + 1 = (n+1)^2.\end{aligned}$$

  Hierbei haben wir mit IV den Umformungsschritt markiert, in dem die Induktionsvoraussetzung $P(n)$ angewendet wird.

Course-Of-Values Induction Nicht immer führt Induktion nach obigem Schema zum Ziel. Wenn wir z.B. den Fundamentalsatz der Arithmetik

Jede positive natürliche Zahl ist ein endliches Produkt von Primzahlen

beweisen wollen, wird uns die Annahme, dass n ein Produkt von Primzahlen ist, erkennbar nicht weiterhelfen beim Beweis der Behauptung, dass $n+1$ ein Produkt von Primzahlen ist (im Gegenteil teilen ja die Primzahlen, aus denen n zusammengesetzt ist, $n+1$ gerade *nicht*). Stattdessen verwenden wir folgendes stärkere Induktionsprinzip:

Satz 1 (Course-of-Values Induction)**.** *Sei $P(n)$ eine Aussage über natürliche Zahlen.*[1] *Wenn für jedes n aus*

$$\forall k < n.\, P(k)$$

$P(n)$ folgt ($\forall n.\, (\forall k < n.\, P(k)) \implies P(n)$), so gilt P für jede natürliche Zahl ($\forall n.\, P(n)$).

Beweis. In der Tat lässt sich diese Prinzip mittels normaler Induktion über n beweisen. Dies ist gleichzeitig eine Illustration des Prinzip der *Verstärkung des Induktionsziels*: Wenn sich eine Aussage $P(n)$ nicht durch Induktion beweisen lässt, kommt man oft weiter, wenn man stattdessen eine *stärkere* Aussage $Q(n)$ (d.h. eine Aussage $Q(n)$ mit $\forall n.\, Q(n) \implies P(n)$) per Induktion beweist. Man hat dann zwar im Induktionsschritt mehr zu zeigen, hat aber dazu eine stärkere Induktionsannahme zur Verfügung. Auch im vorliegenden Fall kann man sich überzeugen, dass die Induktionsannahme $P(n)$ unter den Annahmen des Satzes nicht ausreicht, um $P(n+1)$ zu folgern (dazu braucht man $P(k)$ für alle $k < n+1$, die Induktionsannahme liefert dies aber nur für den Fall $k = n$). Stattdessen beweisen wir unter den Annahmen des Satzes die stärkere Aussage

$$\forall k \leq n.\, P(n)$$

durch Induktion über n:

[1] Wir sind hier, wie schon vorher, ungenau bezüglich der Ausdrucksmittel, die zur Formulierung von P zur Verfügung stehen, insofern bleibt der Satz hier zum Teil informell. Man könnte mit weiter unten eingeführtem Wissen z.B. verlangen, dass P eine Formel in Logik erster Stufe ist, die nur 0 und Nachfolger erwähnt.

- Induktionsanfang: nach Annahme können wir $P(0)$ folgern, wenn $P(k)$ für alle natürlichen Zahlen $k < 0$ gilt. Da es keine solchen Zahlen gibt, ist dies der Fall, also gilt $P(0)$. Damit gilt natürlich auch $\forall k \leq 0.\, P(k)$.
- Induktionsschritt: Es gelte $\forall k \leq n.\, P(n)$; zu zeigen ist $\forall k \leq n+1.\, P(n)$. Für die meisten k folgt dies sofort aus der Induktionsannahme; zu zeigen bleibt $P(n+1)$. Nach der Annahme des Satzes reicht es dazu aus, zu zeigen, dass $P(k)$ für alle $k < n+1$ gilt; das ist aber gerade unsere Induktionsannahme $\forall k \leq n.\, P(n)$.

□

Mit diesem Prinzip beweisen wir nun den eingangs erwähnten Fundamentalsatz der Arithmetik: Sei $n > 0$. Wir nehmen an, jede Zahl $k < n$ sei ein Produkt von endlich vielen Primzahlen (Sonderfälle hierbei per Konvention: 1 ist ein Produkt von 0 Primzahlen, und jede Primzahl ist Produkt aus einer einzigen Primzahl). Wir müssen zeigen, dass dann n selbst ein Produkt endlich vieler Primzahlen ist. Wir unterscheiden dazu zwei Fälle: Wenn n selbst prim ist oder $n = 1$, dann ist n per eben vereinbarter Konvention ein endliches Produkt von Primzahlen. Andernfalls ist n zusammengesetzt, also $n = km$ mit $k, m < n$. Nach Induktionsvoraussetzung sind dann k und m endliche Produkte von Primzahlen, also $k = p_1 \dots p_r$, $m = q_1 \dots q_s$ mit $p_1, \dots, p_r, q_1, \dots, q_s$ Primzahlen; damit ist auch $n = km = p_1 \dots p_r q_1 \dots q_s$ ein Produkt endlich vieler Primzahlen.

2.1.1 Backus-Naur-Form

Die Backus-Naur-Form (BNF) ist eine verbreitete Art, die Syntax von formalen Sprachen (genauer: sogenannten kontextfreien Sprachen, s. BFS) darzustellen. Man arbeitet mit zwei Alphabeten T und N von *terminalen* und *nichtterminalen* Symbolen; wir beschränken uns hier der Einfachheit halber auf den Fall mit nur einem nichtterminalen Symbol. Eine BNF hat dann die Form

$$n \ ::= \ B_1 \mid \dots \mid B_m$$

mit $n \in N$ und $B_1, \dots, B_m \in (N \cup T)^*$; die B_i heißen *Alternativen*. Eine Alternative

$$B_i = w_0 n w_1 n \dots n w_k \text{ mit } w_0, \dots, w_k \in T^*$$

lesen wir als eine Regel

> wenn $v_1, \dots, v_k$ *Instanzen* von n sind, dann auch $w_0 v_1 w_1 v_2 \dots v_k w_k$.

Die so entstehende Regelmenge lesen wir *induktiv*, d.h. ein Wort über T ist dann eine Instanz von n, wenn sich dies durch endlich viele Regelanwendungen herleiten lässt. Insbesondere sind alle Instanzen von n endliche Worte.

Unser erstes Beispiel ist die Grammatik

$$\varphi, \psi ::= \bot \mid A \mid \varphi \wedge \psi \mid \neg\, \varphi \qquad (A \in \mathscr{A}),$$

wobei wir zwecks leichterer Notation zwei verschiedene Namen ϕ, ψ für dasselbe Nichtterminal verwenden. Dabei ist $\mathscr{A}$ eine gegebene Menge von *Atomen*, d.h. nicht weiter unterteilbaren Aussagen. Wir haben hier also $T = \mathscr{A} \cup \{\bot, \wedge, \neg\}$ (bzw. $T = \mathscr{A} \cup \{\bot, \wedge, \neg,), (\}$, wobei wir aber Klammern in der Grammatik implizit lassen und Terme nur bei Bedarf klammern). Instanzen von ϕ nennen wir *(aussagenlogische) Formeln.* Dies entspricht in der Lesart als Regeln dem Regelsystem

1. $\bot$ und $A \in \mathscr{A}$ sind Formeln.
2. Wenn ϕ eine Formel ist, dann auch $\neg\phi$.
3. Wenn ϕ und ψ Formeln sind, dann auch $\phi \wedge \psi$.

Z.B. ist $A \wedge \neg\bot$ eine Formel; durch Rückwärtsanwenden der Regeln sieht man dies wie folgt:

- $A \wedge \neg\bot$ ist eine Formel, denn (3):
- $A \in \mathscr{A}$ ist eine Formel (1) und $\neg\bot$ ist eine Formel, denn (2):
  - $\bot$ ist eine Formel (1).

2.2 Strukturelle Induktion

Man kann nun Induktion nicht nur über den natürlichen Zahlen verwenden, sondern auch über im wesentlichen allen endlichen azyklischen Datenstrukturen (und sogar noch allgemeineren Objekten, was hier aber zu weit führt) – insbesondere z.B. über mittels einer Grammatik definierten Objekten, wie etwa aussagenlogischen Formeln.

Wir stellen uns ein solches Objekt dabei eher als eine baumförmige Struktur als einen flachen String vor (d.h. wir stellen uns z.B. Formeln fertig geparst vor). In ihrer einfachsten Form besagt strukturelle Induktion dann, dass jede Eigenschaft, die für alle Blätter gilt und sich von direkten Kindern auf Elternknoten vererbt, für alle Bäume gilt.

Formal stellt sich dies wie folgt dar: Aus einer BNF

$$n ::= B_1 \mid \cdots \mid B_m$$

erhalten wir ein Induktionsprinzip zum Beweis einer Eigenschaft P für alle Instanzen von n, in dem m verschiede Induktionsschritte durchzuführen sind, einer für jedes B_i. Der Induktionsschritt für $B_i = w_0 n w_1 n \ldots n w_k$ verlangt, dass man unter der Annahme, dass Instanzen $v_1, \ldots, v_k$ von n bereits P erfüllen (Induktionsvoraussetzung), zeigt, dass auch die neu erzeugte Instanz

$$w_0 v_1 w_1 \ldots v_k w_k$$

P erfüllt. Wenn n nicht in B_i vorkommt, B_i also nur aus terminalen Symbolen besteht, ist der Induktionschritt für B_i natürlich eher eine Art Induktionsanfang (von denen es dann mehrere geben kann), da man keine Induktionsvoraussetzung hat. Die Rechtfertigung dieses Induktionsprinzips, d.h. der Beweis der Tatsache, dass man nach Durchführung aller Induktionsschritte tatsächlich folgern kann, dass $P(w)$ für alle Instanzen w von n gilt, ist per Course-of-Values-Induktion über die Länge von w, bei Fallunterscheidung über die Regel, mit der man w erzeugt hat (wegen der impliziten Klammerung ist diese Regel eindeutig bestimmt).

Ein erstes Beispiel dieses Prinzips ist die eingangs diskutierte Induktion über natürliche Zahlen. Wir können nämlich die natürlichen Zahlen als die Instanzen der Grammatik

$$n ::= Z \mid S(n)$$

ansehen – diese sind Z, $S(Z)$, $S(S(Z))$, $S(S(S(Z)))$, ... Dann haben wir gemäß den obigen Vorschriften beim Beweis einer Eigenschaft P für alle Instanzen von n zwei Induktionsschritte, einen für jede Alternative der Grammatik:

- Z: Hier kommt n nicht vor, zu zeigen ist also einfach $P(Z)$. Dies entspricht dem üblichen Induktionsanfang (Z steht für 0).

- $S(n)$: Hier ist zu zeigen, dass, wenn n die Eigenschaft P hat, dann auch $S(n)$, wobei jetzt n als Platzhalter für eine beliebige Instanz steht. Da S für die Nachfolgerfunktion steht, entspricht dies genau dem üblichen Induktionsschritt.

Für aussagenlogische Formeln erhalten wir ein strukturelles Induktionsprinzip mit vier Induktionsschritten (von denen zwei in Wirklichkeit Induktionsanfänge sind): Um zu zeigen, dass eine Eigenschaft $P(\phi)$ für alle aussagenlogischen Formeln ϕ gilt, zeigt man

- $P(\bot)$;

- $P(A)$ für alle $A \in \mathscr{A}$;

- wenn $P(\phi)$, dann auch $P(\neg\phi)$; und

- wenn $P(\phi)$ und $P(\psi)$, dann auch $P(\phi \wedge \psi)$.

Wir verwenden dieses Prinzip ganz entsprechend auch zur *rekursiven Definition* von Funktionen, wie etwa in der folgenden Definition.

Definition 2 (Atome einer Formel)**.** Die Menge $\mathrm{At}(\varphi)$ der in φ vorkommenden Atome ist rekursiv definiert durch

$$\begin{aligned} \mathrm{At}(A) &= \{A\} \\ \mathrm{At}(\top) &= \emptyset \\ \mathrm{At}(\neg\varphi) &= \mathrm{At}(\varphi) \\ \mathrm{At}(\varphi \wedge \psi) &= \mathrm{At}(\varphi) \cup \mathrm{At}(\psi) \end{aligned}$$

Das Schema der rekursiven Aufrufe ist dasselbe wie das der Induktionsvoraussetzungen im Induktionsprinzip, d.h. rekursive Aufrufe erfolgen immer auf die Bestandteile des aktuellen Arguments – bei der Klausel für $\mathrm{At}(\varphi \wedge \psi)$ z.B. auf ϕ und ψ. Als Beispiel zeigen wir

> *Für jede Formel ϕ ist die Menge* $\mathrm{At}(\phi)$ *endlich*

durch Induktion über ϕ:

- $\mathrm{At}(\bot) = \emptyset$ ist endlich.

- Für $A \in \mathscr{A}$ ist $\mathrm{At}(A) = \{A\}$ endlich.

- Sei $\mathrm{At}(\phi)$ endlich; dann ist auch $\mathrm{At}(\neg\phi) = \mathrm{At}(\phi)$ endlich.

- Seien $\mathrm{At}(\phi)$ und $\mathrm{At}(\psi)$ endlich; dann ist auch $\mathrm{At}(\phi \wedge \psi) = \mathrm{At}(\phi) \cup \mathrm{At}(\psi)$ endlich, da die Vereinigung zweier endlicher Mengen wieder eine endliche Menge ist.

2.3 Syntax

2.3.1 Syntax der Aussagenlogik

Wir definieren (wie im Abschnitt über Induktion) die Menge der aussagenlogischen Formeln φ, ψ durch die Grammatik

$$\varphi, \psi ::= \bot \mid A \mid \varphi \wedge \psi \mid \neg\, \varphi \qquad (A \in \mathscr{A}).$$

Wir vereinbaren:

- $\neg$ bindet am stärksten
- $\top = \neg\bot$
- $(\varphi \vee \psi) = \neg(\neg\varphi \wedge \neg\psi)$
- $\varphi \to \psi = \neg\varphi \vee \psi$
- $\varphi \leftrightarrow \psi = (\varphi \to \psi) \wedge (\psi \to \varphi)$
- $\wedge$ bindet stärker als $\vee$, und $\vee$ bindet stärker als $\to$ und $\leftrightarrow$.

(Hierbei bezeichnet $=$ syntaktische Gleichheit; z.B. $A \wedge B \neq B \wedge A$!)

2.4 Semantik der Aussagenlogik

2.4.1 Informelle Semantik

Die Aussprache und intuitive Bedeutung der logischen Operatoren ist

$\varphi \wedge \psi$	„ϕ und ψ“
$\neg\varphi$	„nicht ϕ“
$\varphi \vee \psi$	„ϕ oder ψ“
$\varphi \to \psi$	„wenn ϕ, dann ψ“
$\varphi \leftrightarrow \psi$	„genau dann φ, wenn ψ“
$\top$	„Wahr“
$\bot$	„Falsch“

Hierbei ist „oder“ als *inklusives Oder* zu lesen, d.h. es dürfen auch beide Aussagen wahr sein; „wenn“-„dann“ ist eine *materielle Implikation*, d.h. wenn ϕ nicht gilt, ist $\phi \to \psi$ stets wahr – z.B. auch dann, wenn ψ falsch ist. Ferner ist $\phi \to \psi$ stets wahr, wenn ψ gilt, ohne Rücksicht darauf, ob diese Tatsache etwas mit ϕ zu tun hat.

Für die logischen Operatoren werden folgende sprachliche Bezeichnungen verwendet:

$\neg$	Negation
$\wedge$	Konjunktion
$\vee$	Disjunktion
$\rightarrow$	Implikation
$\leftrightarrow$	Biimplikation, Äquivalenz
$\top$	Wahrheit, konstant wahre Aussage
$\bot$	Falschheit, konstant falsche Aussage, Absurdität

2.4.2 Wahrheitsbelegungen und Erfülltheit

Definition 3 (Wahrheitsbelegung, Erfülltheit)**.** Wir schreiben $2 = \{\bot, \top\}$ für die Menge der *Wahrheitswerte*. Eine *Wahrheitsbelegung (WB)* ist eine Abbildung

$$\kappa : \mathscr{A} \to 2$$

legt also Wahrheitswerte für alle Atome fest.

Wir definieren die *Erfüllheitsrelation* $\kappa \vDash \varphi$ (lies: κ erfüllt φ) rekursiv durch

$$\begin{aligned} \kappa &\nvDash \bot \\ \kappa \vDash A &\iff \kappa(A) = \top \\ \kappa \vDash \varphi \wedge \psi &\iff \kappa \vDash \varphi \text{ und } \kappa \vDash \psi \\ \kappa \vDash \neg\varphi &\iff \kappa \nvDash \varphi. \end{aligned}$$

In Worten: κ erfüllt $\bot$ nicht; κ erfüllt ein Atom A, wenn es A den Wert „wahr“ zuweist; κ erfüllt die Konjunktion zweier Formeln, wenn κ beide Formeln erfüllt; κ erfüllt die Negation einer Formel ϕ, wenn κ die Formel ϕ nicht erfüllt.

Erfülltheit verhält sich für die abgeleiteten Operatoren wie folgt:

$$\begin{aligned} \kappa &\vDash \top \text{ stets} \\ \kappa &\vDash \bot \text{ nie} \\ \kappa \vDash \varphi \vee \psi &\iff (\kappa \vDash \varphi \text{ oder } \kappa \vDash \psi) \\ \kappa \vDash \varphi \rightarrow \psi &\iff (\text{Falls } \kappa \vDash \varphi, \text{ so auch } \kappa \vDash \psi) \\ \kappa \vDash \varphi \leftrightarrow \psi &\iff (\kappa \vDash \varphi \text{ genau dann wenn } \kappa \vDash \psi) \end{aligned}$$

Beispiel 4. Die Wahrheitsbelegung κ ordne den Atomen A und B die Werte $\kappa(A) = \top$ und $\kappa(B) = \bot$ zu; Kurzschreibweise: $\kappa = [A \mapsto \top, B \mapsto \bot]$ (das spezifiziert κ nicht eindeutig, was im folgenden aber nicht stört). Dann gilt

$$\kappa \vDash ((A \vee \neg B) \rightarrow B) \iff (\text{Falls } \kappa \vDash (A \vee \neg B), \text{ so auch } \kappa \vDash B)$$

Es gilt $\kappa \vDash A \vee \neg B \iff (\kappa \vDash A \text{ oder } \kappa \vDash \neg B)$. Nun gilt $\kappa(A) = \top$, also $\kappa \vDash A$, somit $\kappa \vDash A \vee \neg B$. Es gilt aber $\kappa \nvDash B$ (da $\kappa(B) = \bot$), also $\kappa \nvDash (A \vee \neg B) \rightarrow B$.

Eine andere Wahrheitsbelegung κ_2, die den Atomen andere Wahrheitswerte zuordnet, kann die Formel aber erfüllen, z. B. gilt $\kappa_2 \vDash (A \vee \neg B) \rightarrow B$ für $\kappa_2 = [A \mapsto \top, B \mapsto \top]$.

2.5 Logische Konsequenz

Eine logische Konsequenz oder logische Folgerung ist die korrekte Ableitung einer neuen Formel aus einer Menge als gültig vorausgesetzter Formeln.

Für die formale Definition definieren wir zunächst die Erfülltheit einer Menge $\Phi \subseteq F$ von Formeln: Eine Wahrheitsbelegung κ erfüllt die Menge Φ genau dann, wenn κ alle Formeln φ erfüllt, die in Φ enthalten sind, d.h.

$$\kappa \vDash \Phi \;:\Longleftrightarrow\; \forall \varphi \in \Phi.\, \kappa \vDash \varphi$$

Definition 5 (logische Konsequenz)**.** Sei $\Phi \subseteq F$ eine Menge von Formeln. Eine Formel $\psi \in F$ ist eine *logische Konsequenz* von Φ, wenn für alle Wahrheitsbelegungen $\kappa : \mathscr{A} \to 2$ gilt, dass, falls $\kappa \vDash \Phi$, so auch $\kappa \vDash \psi$. Man schreibt für „ψ ist logische Konsequenz von Φ“ auch $\Phi \vDash \psi$. In Symbolen:

$$\Phi \vDash \psi \;:\Longleftrightarrow\; \forall \kappa.\, (\kappa \vDash \Phi \implies \kappa \vDash \psi).$$

Daraus lassen sich nun einige neue Begriffe und Definitionen ableiten.

Definition 6 (Gültigkeit einer Formel ψ)**.** Eine Formel ψ ist *gültig*, wenn sie aus der leeren Menge von Annahmen folgt:

$$\models \psi :\Longleftrightarrow \emptyset \vDash \psi \iff \forall \kappa : \kappa \vDash \psi,$$

d.h. also wenn alle Wahrheitsbelegungen ψ erfüllen. Wir nennen ψ in diesem Fall auch *tautologisch* oder eine *Tautologie*.

Definition 7 (Erfüllbarkeit einer Menge von Formeln Φ)**.** Eine Formelmenge Φ ist *unerfüllbar*, wenn sich aus ihr ein Widerspruch herleiten lässt, d.h. wenn $\bot$ eine logische Konsequenz von Φ ist: $\Phi \vDash \bot$. Dies ist gleichbedeutend damit, dass keine Wahrheitsbelegung Φ erfüllt: $\forall \kappa : \kappa \nvDash \Phi$.

Andernfalls, d.h. wenn $\exists \kappa : \kappa \vDash \Phi$, heißt Φ *erfüllbar*.

Definition 8 (Logische Äquivalenz zweier Formeln φ und ψ)**.** Zwei Formeln φ, ψ sind logisch äquivalent ($\varphi \equiv \psi$), wenn $\varphi \leftrightarrow \psi$ gültig ist:

$$\varphi \equiv \psi :\Longleftrightarrow \models \phi \leftrightarrow \psi.$$

Lemma 9. *Eine Formel ψ ist genau dann logische Konsequenz einer Formelmenge Φ, wenn die Vereinigung von Φ und der Negation von ψ unerfüllbar ist:*

$$\Phi \vDash \psi \iff (\Phi \cup \{\neg\psi\}) \vDash \bot.$$

Beweis. Wir zeigen zwei Implikationen:

„$\Longleftarrow$“: Sei $\kappa \models \Phi$; wir müssen $\kappa \models \psi$ zeigen. Beweis durch Widerspruch: wenn $\kappa \not\models \psi$, dann per Definition $\kappa \models \neg\psi$, also $\kappa \models \Phi \cup \{\neg\psi\}$. Letzteres ist aber nach Annahme unerfüllbar, Widerspruch.

„$\Longrightarrow$“: Diese Richtung zeigen wir durch *Kontraposition*: allgemein ist $A \implies B$ äquivalent zu $\neg B \implies \neg A$. Wir nehmen also die Negation der rechten Seite an und zeigen dann die Negation der linken Seite. Sei also $\Phi \cup \{\neg\psi\} \not\models \bot$. Dann existiert κ mit $\kappa \models \Phi \cup \{\neg\psi\}$. Insbesondere gilt dann $\kappa \models \Phi$, aber $\kappa \not\models \psi$, also $\Phi \not\models \psi$.

□

Beispiel 10 (Erfüllbarkeit und logische Konsequenz)**.**

Erfüllbar: $A \to \neg A$ (Für die Belegung $\kappa(A) = \bot$)

Unerfüllbar: $A \wedge \neg A$

Gültig: $A \vee \neg A$, $(A \wedge B) \to A$

Logische Konsequenz: $\{A \to B, A\} \models B$ (diesen Schluss nennt man „modus ponens“)

2.6 Wahrheitstafeln

Eine Wahrheitstafel ist eine tabellarische Auflistung der Wahrheitswerte einer Formel in Abhängigkeit von den Wahrheitswerten der (endlich vielen!) in ihr vorkommenden Atome. Wahrheitstafeln liefern Entscheidungsalgorithmen für Erfüllbarkeit, Gültigkeit, logische Konsequenz und logische Äquivalenz in der Aussagenlogik; diese sind aber offenbar in der Praxis nicht skalierbar, da stets die gesamte (exponentiell große) Wahrheitstafel erzeugt werden muss. Konkret:

- φ ist gültig genau dann, wenn alle Werte für φ in der Wahrheitstafel $\top$ sind .
- φ ist erfüllbar genau dann, wenn in der Wahrheitstafel für φ $\top$ als Wert für φ vorkommt.
- $\varphi \equiv \psi$ genau dann, wenn ϕ und ψ in der gemeinsamen (!) Wahrheitstafel in jeder Zeile den gleichen Wert haben.
- $\Phi \models \phi$ (für Φ endlich) genau dann, wenn in der gemeinsamen Wahrheitstafel für Φ und ψ die Formel ψ in jeder Zeile, in der alle $\varphi \in \Phi$ den Wert $\top$ haben, ebenfalls den Wert $\top$ hat.

Beispiel 11. Wahrheitstafel von $A \to B = \neg A \vee B$:

A	B	$\neg A$	$\neg A \vee B$
$\bot$	$\bot$	$\top$	$\top$
$\bot$	$\top$	$\top$	$\top$
$\top$	$\bot$	$\bot$	$\bot$
$\top$	$\top$	$\bot$	$\top$

$\neg A \vee A$ ist gültig:

A	$\neg A$	$\neg A \vee A$
$\bot$	$\top$	$\top$
$\top$	$\bot$	$\top$

$A \to \neg A$ ist erfüllbar:

A	$\neg A$	$A \to \neg A$
$\bot$	$\top$	$\top$
$\top$	$\bot$	$\bot$

$\neg(A \to B) \models A$:

A	B	$A \to B$	$\neg(A \to B)$
$\bot$	$\bot$	$\top$	$\bot$
$\bot$	$\top$	$\top$	$\bot$
$\top$	$\bot$	$\bot$	$\top$
$\top$	$\top$	$\top$	$\bot$

$\{A \to B, A\} \models B$:

A	B	$A \to B$
$\bot$	$\bot$	$\top$
$\bot$	$\top$	$\top$
$\top$	$\bot$	$\bot$
$\top$	$\top$	$\top$

(Im letzten Beispiel muss man Erfülltheit von B nur in der letzten Zeile prüfen, da nur dort die beiden Annahmen A und $A \to B$ erfüllt sind.)

$B \vee \neg B \equiv A \vee \neg A$: bei beiden Formeln steht nur $\top$ in der Wahrheitstafel.

Im letzten Beispiel sieht man, dass zwei Formeln trotz unterschiedlicher verwendeter Atome äquivalent sein können.

Definition 12 (Atome einer Formel)**.** Die Menge $\mathrm{At}(\varphi)$ der in φ vorkommenden Atome ist wie im Abschnitt über Induktion rekursiv definiert durch

$$\begin{aligned}\mathrm{At}(A) &= \{A\}\\ \mathrm{At}(\top) &= \emptyset\\ \mathrm{At}(\neg\varphi) &= \mathrm{At}(\varphi)\\ \mathrm{At}(\varphi \wedge \psi) &= \mathrm{At}(\varphi) \cup \mathrm{At}(\psi)\end{aligned}$$

Beispiel 13. Wir berechnen $\mathrm{At}((A \wedge B) \wedge \neg A)$:

$$\begin{aligned}\mathrm{At}((A \wedge B) \wedge \neg A) &= \mathrm{At}(A \wedge B) \cup \mathrm{At}(\neg A)\\ &= \mathrm{At}(A) \cup \mathrm{At}(B) \cup \mathrm{At}(A)\\ &= \{A\} \cup \{B\} \cup \{A\} = \{A, B\}\end{aligned}$$

Das folgende Lemma ist die formale Legitimation dafür, dass Wahrheitstafeln sich auf die in φ vorkommenden Atome beschränken dürfen.

Lemma 14. *Die Erfülltheit $\kappa \vDash \varphi$ hängt nur von den Belegungen der Atome von φ, also von den Werten $\kappa(A)$ für $A \in \mathrm{At}(\varphi)$ ab; d.h. wenn κ und κ' auf $\mathrm{At}(\varphi)$ übereinstimmen, so erfüllt κ genau dann φ, wenn κ' dies tut. Formal: Wenn $\kappa, \kappa' : \mathscr{A} \to 2$ mit $\kappa(A) = \kappa'(A)$ für alle $A \in \mathrm{At}(\varphi)$, dann gilt*

$$\kappa \vDash \varphi \iff \kappa' \vDash \varphi.$$

Beweis. Strukturelle Induktion über ϕ. Wir gehen alle Alternativen der Grammatik durch:

$\bot$: Es gilt $\kappa \nvDash \bot$ und $\kappa' \nvDash \bot$.

A: Es $A \in \mathrm{At}(A)$, also $\kappa \vDash A \iff \kappa(A) = \top \iff \kappa'(A) = \top \iff \kappa' \vDash A$.

$\neg\phi$: Es gilt $\mathrm{At}(\neg\phi) = \mathrm{At}(\phi)$, d.h. κ und κ' stimmen auf $\mathrm{At}(\phi)$ überein. Daher $\kappa \models \neg\phi \iff \kappa \not\models \phi \overset{\text{IV}}{\iff} \kappa' \not\models \phi \iff \kappa' \models \neg\phi$.

$\phi\wedge\psi$: Es gilt $\mathrm{At}(\phi\wedge\psi) \supseteq \mathrm{At}(\phi), \mathrm{At}(\psi)$, d.h. κ und κ' stimmen auf $\mathrm{At}(\phi)$ und auf $\mathrm{At}(\psi)$ überein. Daher $\kappa \models \phi \wedge \psi \iff (\kappa \models \phi \text{ und } \kappa \models \psi) \overset{\text{IV}}{\iff} (\kappa' \models \phi \text{ und } \kappa' \models \psi) \iff \kappa' \models \phi \wedge \psi$.

□

Die Semantik von φ ist also bestimmt durch endliche Tabellierung von $\kappa \vDash \varphi$ für alle $\kappa : A_0 \to 2$ mit $A_0 \subseteq \mathscr{A}$ endlich, $\mathrm{At}(\varphi) \subseteq A_0$.

Wir präzisieren mittels der At-Notation das Wahrheitstafelverfahren für logische Äquivalenz:

Lemma 15. *$\varphi \equiv \psi$ genau dann wenn φ, ψ identische Wahrheitstafeln über $\mathrm{At}(\varphi) \cup \mathrm{At}(\psi)$ haben.*

2.7 Logische Äquivalenzen

Im folgenden geben wir eine Übersicht wichtiger logischer Äquivalenzen.

- $\neg\neg\varphi \equiv \varphi$ (Doppelnegationselimination)
- $\left.\begin{array}{l}\neg(\varphi \wedge \psi) \equiv (\neg\varphi \vee \neg\psi)\\ \neg(\varphi \vee \psi) \equiv (\neg\varphi \wedge \neg\psi)\end{array}\right\}$ (De Morgansche Gesetze)
- $\left.\begin{array}{l}\varphi \wedge (\psi \vee \chi) \equiv (\varphi \wedge \psi) \vee (\varphi \wedge \chi)\\ \varphi \vee (\psi \wedge \chi) \equiv (\varphi \vee \psi) \wedge (\varphi \vee \chi)\end{array}\right\}$ (Distributivgesetze)
- $\left.\begin{array}{l}(\varphi \wedge \psi) \wedge \chi \equiv \varphi \wedge (\psi \wedge \chi)\\ \varphi \vee (\psi \vee \chi) \equiv (\varphi \vee \psi) \vee \chi\end{array}\right\}$ (Assoziativgesetze)
- $\left.\begin{array}{l}\chi \wedge \top \equiv \chi\\ \chi \vee \bot \equiv \chi\end{array}\right\}$ (Neutrale Elemente)
- $\varphi \wedge \psi \equiv \psi \wedge \varphi$ (Kommutativität)
- $\varphi \wedge \varphi \equiv \varphi$ (Idempotenz)

3 Formale Deduktion in Aussagenlogik

Systeme formalen Schließens dienen der rein syntaxbasierten *Herleitung* von Formeln, die einfach als zu manipulierende Zeichenketten angesehen werden. Ein Deduktionssystem besteht typischerweise aus *Axiomen*, also Formeln, die ohne weitere Voraussetzungen als hergeleitet hingeschrieben werden können, sowie aus *Regeln*, die festlegen, wie man aus bereits hergeleiteten Formeln bzw. schon durchgeführten Herleitungen neue Formeln herleitet. Im einfachsten Fall hat eine Regel die Form

$$\frac{\text{Prämissen}}{\text{Konklusion}}.$$

Die *Prämissen* repräsentieren Formeln, die zur Anwendung der Regel bereits hergeleitet sein müssen; die Regel gestattet dann die Herleitung der *Konklusion*. Zusätzlich kann eine Regel *Seitenbedingungen* haben, also außerhalb der Syntax von Formeln ausgedrückte Bedingungen für die Anwendbarkeit der Regel, idealerweise einfache Zusatzforderungen an die syntaktische Struktur wie etwa das Vorkommen oder Nichtvorkommen gewisser Symbole. In komplizierteren Fällen können die Prämissen statt Formeln auch ganze Herleitungen sein; dazu sehen wir später Beispiele.

Man unterscheidet verschiedene Typen von formalen Deduktionssystemen u.a. nach der Gewichtung zwischen Axiomen und Regeln:

Hilbert Viele Axiome, wenig Regeln; meist nur *modus ponens*

$$(mp)\ \frac{\varphi \quad \varphi \to \psi}{\psi}$$

Gentzen Regeln für *Sequenten* $\varphi_1, \ldots, \varphi_n \vdash \psi_1, \ldots, \psi_k$, zu lesen als 'die Konjunktion der φ_i impliziert die Disjunktion der ψ_j. Axiome sind nur Sequenten der Form $\varphi, \cdots \vdash \varphi, \ldots$.

Natürliches Schließen Variante des Sequentenkalküls, in der nur eine Formel rechts von $\vdash$ steht und die linke Seite implizit gelassen wird; stattdessen hat man lokale Mengen von Annahmen, die in hierarchisch strukturierten Beweisen rekursiv aufgebaut werden.

Wir lernen im folgenden ein System natürlichen Schließens kennen, den Fitch-Kalkül.

Oft sind die Regeln eines Kalküls so organisiert, dass man für jedes logische Konnektiv Regeln sowohl zur Einführung (I wie *Introduction*) als auch zur Entfernung (E wie *Elimination*) bereitstellt. Diese Struktur weist auch das folgende einführende Beispiel auf.

Beispiel 16 (Regeln für eine auf Konjunktion beschränkte Logik)**.**

$$(\wedge I)\ \frac{\varphi \quad \psi}{\varphi \wedge \psi} \qquad (\wedge E1)\ \frac{\varphi \wedge \psi}{\varphi} \qquad (\wedge E2)\ \frac{\varphi \wedge \psi}{\psi}$$

Die Regel $(\wedge I)$ („Und-Introduktion") erlaubt es uns, sofern wir bereits φ und ψ hergeleitet haben, auch $\varphi \wedge \psi$ herzuleiten, die Regeln $(\wedge E1)$ und $(\wedge E2)$ („Und-Elimination") erlauben es, aus einer bereits hergeleiteten Konjunktion deren Konjunkte herzuleiten.

Notation: Wir schreiben $\Phi \vdash \varphi$, wenn φ per Regeln aus Annahmen in Φ (Φ Menge von Formeln) rein syntaktisch herleitbar ist.

Schreibweise:
Baumartig:
z.B. $\{\varphi \wedge \psi\} \vdash \psi \wedge \varphi$:

$$\frac{(\wedge E2)\frac{\varphi \wedge \psi}{\varphi} \qquad (\wedge E1)\frac{\varphi \wedge \psi}{\psi}}{\psi \wedge \varphi}(\wedge I)$$

Der Übersichtlichkeit halber verwenden wir im folgenden eine lineare Darstellung des Beweisablaufs:

1	$\varphi \wedge \psi$		
2	ψ	$(\wedge E2)$	(1)
3	φ	$(\wedge E1)$	(1)
4	$\psi \wedge \varphi$	$(\wedge I)$	$(2, 3)$

Wir notieren in jedem Schritt die hergeleitete Formel (erste Spalte) sowie die verwendete Regel und die Prämissen (zweite Spalte). Die erste Zeile enthält eine Annahme; Annahmen werden von Schlüssen durch einen waagerechten Strich getrennt.

Innerhalb eines Beweises ist es möglich, dass ein Unterbeweis als Prämisse verwendet wird, es ergibt sich daraus eine hierarchische Struktur. Unterbeweise haben *lokale* Annahmen, die ebenso wie oben durch einen waagerechten Strich abgetrennt werden. Die lokale Annahme verschwindet mit dem Ende des Unterbeweises (daher der Name), d.h. sie steht zum einen außerhalb des Unterbeweises nicht als Prämisse für Regelanwendungen zur Verfügung, zum anderen ist aber der äußere Beweis auch von ihr unabhängig, d.h. liefert Herleitungen mit entsprechend weniger Annahmen (also stärkere Aussagen). Solche Unterbeweise werden benötigt, wenn wir unser Regelwerk auf Negation und $\bot$ erweitern:

$$(\neg I)\ \dfrac{\begin{array}{|l} \varphi \\ \hline \vdots \\ \bot \end{array}}{\neg\varphi} \qquad (\bot\ \mathrm{I})\ \dfrac{\varphi \quad \neg\varphi}{\bot} \qquad (\bot\ \mathrm{E})\ \dfrac{\bot}{\Phi} \qquad (\neg\ \mathrm{E})\ \dfrac{\neg\neg\varphi}{\varphi}$$

Beispiel 17 $(\vdash \neg(\varphi \wedge \neg\varphi))$**.**

1			
2	$\varphi \wedge \neg\varphi$		
3	φ	$(\wedge E1), 1$	
4	$\neg\varphi$	$(\wedge E1), 1$	
5	$\bot$	$(\bot I), 2, 3$	
6	$\neg(\varphi \wedge \neg\varphi)$	$(\neg I), 1-4$	

Erweiterung auf Disjunktion und Implikation

$$(\vee I1)\dfrac{\varphi}{\varphi \vee \psi} \qquad (\vee I2)\dfrac{\psi}{\varphi \vee \psi} \qquad (\vee E)\dfrac{\begin{array}{ccc} \varphi & \psi & \\ \vdots & \vdots & \varphi \vee \psi \\ \chi & \chi & \end{array}}{\chi}$$

$$(\to E)\dfrac{\varphi \to \psi \quad \varphi}{\psi} \qquad (\to I)\dfrac{\begin{array}{|l} \varphi \\ \hline \vdots \\ \psi \end{array}}{\varphi \to \psi}$$

Herleitung von $(\to I)$ bei Codierung $\varphi \to \psi = \neg(\neg\neg\varphi \wedge \neg\psi)$

1	$\neg\neg\varphi \wedge \neg\psi$	
2	$\neg\neg\varphi$	$(\wedge E_1)$ 1
3	φ	$(\neg E)$ 2
4	$\vdots$	Annahme
5	ψ	
6	$\neg\psi$	$(\wedge E_2)$ 1
7	$\bot$	$(\bot I)$ 5,6
8	$\neg(\neg\neg\varphi \wedge \neg\psi)$	$(\neg I)$ 1–7

Herleitung von $(\to E)$ bei Codierung $\varphi \to \psi = \neg\varphi \vee \psi$

1	φ	
2	$\neg\varphi \vee \psi$	
3	$\neg\varphi$	
4	$\bot$	$(\bot I)$ 1, 3
5	ψ	$(\bot E)$
6	ψ	
7	ψ	
8	ψ	$(\vee E)$ 3–5, 6–7, 2

Satz 18. *(Korrektheit)*

$$\Phi \vdash \psi \Rightarrow \Phi \vDash \psi$$

Beweis. Formal gesehen müssen wir zunächst die Behauptung verallgemeinern auf Aussagen ψ, die in Unterbeweisen gefolgert werden; wir behaupten, dass für solche Aussagen $\Phi' \models \psi$ gilt, wobei Φ' aus allen in dem betreffenden Unterbeweis aktiven Annahmen besteht (inklusive Φ). Wir beweisen die verallgemeinerte Behauptung per Induktion über die Länge n der Herleitung von ψ. Hierbei verwenden wir *course-of-values induction*, d.h. wir beweisen, dass die Behauptung für $n = k$ gilt, wenn sie für alle $n < k$ gilt.

Wir unterscheiden dann nach der zuletzt angewandten Regel, z.B. a) $\wedge I$ (hier im Skript ausgelassen), b) $(\neg I)$: Der Unterbeweis in der Prämisse bedeutet, dass $\Phi \cup \{\varphi\} \vdash \bot$, wobei Φ die Menge der bei Anwendung der Regel aktiven Annahmen ist und im Unterbeweis eben die Annahme φ dazukommt. Da der Unterbeweis echt kürzer ist als der Gesamtbeweis, können wir auf ihn die Induktionsvoraussetzung anwenden, d.h. es folgt $\Phi \cup \{\varphi\} \vDash \bot$; mit anderen Worten, $\Phi \cup \{\varphi\}$ ist unerfüllbar. Damit folgt wie verlangt $\Phi \vDash \neg\varphi$: Wenn $\kappa \vDash \Phi$, dann $\kappa \nvDash \psi$, also $\kappa \vDash \neg\psi$. Die anderen Regeln für $\wedge$, $\neg$ und $\bot$ sind noch wesentlich einfacher; die Regeln für $\vee$ und $\to$ sind dann ebenfalls korrekt, da wir sie ja aus denen für $\wedge$, $\neg$ und $\bot$ herleiten. $\square$

3.1 Vollständigkeit

In Umkehrung zur Korrektheit gilt auch

Satz 19. *Vollständigkeit*

$$\Phi \vDash \psi \Rightarrow \Phi \vdash \psi$$

Definition 20. Eine Formelmenge Φ heißt *konsistent*, wenn $\Phi \nvdash \bot$, d.h. wenn sich aus Φ kein Widerspruch herleiten lässt. Ferner ist Φ *maximal konsistent*, wenn Φ maximal bezüglich $\subseteq$ unter den konsistenten Mengen ist, d.h.

1. Φ ist konsistent, und
2. wenn Ψ konsistent ist und $\Phi \subseteq \Psi$, dann folgt $\Phi = \Psi$.

Beweis. Strategie des Vollständigkeitsbeweises ist

(A) Zeige

$$\Phi \text{ konsistent} \Rightarrow \Phi \text{ erfüllbar.}$$

Damit folgt Vollständigkeit: Sei $\Phi \vDash \psi$; dann ist $\Phi \cup \{\neg\psi\}$ unerfüllbar, also nach obigem inkonsistent, also $\Phi \cup \{\neg\psi\} \vdash \bot$. Mittels Regel $(\neg I)$ folgt $\Phi \vdash \neg\neg\psi$, und mittels $(\neg E)$ $\Phi \vdash \psi$.

Slogan: Vollständigkeitsbeweise bestehen in *Modellkonstruktionen*.

(B) Zeige (A) zunächst für den Spezialfall, dass Φ *maximal* konsistent ist.

(C) Beweise das

Lindenbaumlemma: Φ konsistent $\Rightarrow$ es existiert $\overline{\Phi}$ max. kons mit $\Phi \subseteq \overline{\Phi}$.

Damit folgt dann (A) aus (B): Wenn Φ konsistent ist, dann existiert $\overline{\Phi}$ wie im Lindenbaumlemma. Nach (B) ist $\overline{\Phi}$ erfüllbar, und damit trivialerweise auch Φ.

□

Wir beginnen mit (C) und zeigen dazu:

Lemma 21 (Konsistenzlemma)**.** *Sei Φ konsistent, dann $\Phi \cup \{\psi\}$ konsistent oder $\Phi \cup \{\neg\psi\}$ konsistent.*

Beweis. Per Kontraposition: Seien $\Phi \cup \{\psi\} \vdash \bot$ und $\Phi \cup \{\neg\psi\} \vdash \bot$. Mittels $(\neg I)$ folgt $\Phi \vdash \neg\psi$ und $\Phi \vdash \neg\neg\psi$, also per $(\bot I)$ $\Phi \vdash \bot$, d.h. Φ ist inkonsistent. □

Beweis (Lindenbaumlemma). **a) Zorn** Vereinigungen aufsteigender Ketten von konsistenten Mengen sind konsistent, also hat die mittels $\subseteq$ geordnete Menge der konsistenten Mengen oberhalb einer gegebenen nach dem Zornschen Lemma maximale Elemente.

oder b) Sei $\varphi_1, \varphi_2, \varphi_3, \ldots$ Aufzählung aller Formeln. Konstruiere Kette $\Phi = \Phi_0 \subseteq \Phi_1 \subseteq \Phi_2 \subseteq \ldots$ konsistenter Formelmengen per

$$\Phi_{i+1} = \begin{cases} \Phi_i \cup \{\varphi_i\} & \text{wenn konsistent} \\ \Phi_i \cup \{\neg\varphi_i\} & \text{sonst (OK per Konsistenzlemma)} \end{cases}$$

Setze $\overline{\Phi} :- \bigcup_{i=0}^{\infty} \Phi_i \supseteq \Phi$. Zu zeigen:

1. $\overline{\Phi}$ ist konsistent: Nimm an $\overline{\Phi} \vdash \bot$. Der Beweis ist endlich, verwendet also nur endlich viele Annahmen aus $\overline{\Phi}$. Jede dieser Annahmen kommt in einem Φ_i vor; durch Wahl des größten unter diesen endlich vielen Indizes i erhalten wir ein Φ_i, das *alle* verwendeten Annahmen enthält. Dann $\Phi_i \vdash \bot$, im Widerspruch zur Konsistenz von Φ_i.
2. $\overline{\Phi}$ maximal: Sei Ψ konsistent und $\overline{\Phi} \subseteq \Psi$. Zu zeigen ist dann $\Psi \subseteq \overline{\Phi}$. Sei also $\psi \in \Psi$. Es existiert n mit $\psi = \phi_n$. Da $\Phi_n \cup \{\phi_n\} \subseteq \Psi$, ist $\Phi_n \cup \{\phi_n\}$ konsistent, also $\phi_n \in \Phi_{n+1} \subseteq \Phi$.

□

Es bleibt Schritt (B) durchzuführen, d.h. wir müssen zeigen, dass jede maximal konsistente Menge Φ erfüllbar ist. Wir halten folgende Eigenschaften maximal konsistenter Mengen fest:

Lemma 22. *(Hintikka-Eigenschaften) Sei Φ maximal konsistent. Dann gilt*

1. $\bot \notin \Phi$
2. $\neg\psi \in \Phi \iff \psi \notin \Phi$
3. $\phi \wedge \psi \in \Phi \iff \phi \in \Phi$ *und* $\psi \in \Phi$

Beweis. ad (1): Klar.

ad (2): „$\Rightarrow$“ klar (verwendet $(\bot I)$), „$\Leftarrow$“: Sei $\psi \notin \Phi$. Dann gilt $\Phi \cup \{\psi\} \supsetneq \Phi$. Da Φ maximal konsistent ist, ist $\Phi \cup \{\psi\}$ inkonsistent; nach Konsistenzlemma folgt, dass $\Phi \cup \{\neg\psi\}$ konsistent ist, und per maximaler Konsistenz von Φ folgt $\Phi \cup \{\neg\psi\} \subseteq \Phi$, also $\neg\psi \in \Phi$.

ad (3): „$\Rightarrow$“: Wir verwenden bereits (2): Wenn z.B. $\phi \notin \Phi$, dann $\neg\phi \in \Phi$ per (2), im Widerspruch zur Konsistenz von Φ ($\bot$ herleitbar per $(\wedge E_1)$, $(\bot I)$.)

„$\Leftarrow$“ läuft analog mit $(\wedge I)$. □

Um nun die verlangte erfüllende Wahrheitsbelegung κ für eine maximal konsistente Menge Φ zu erhalten, setzen wir nun

$$\kappa(A) = \top \iff A \in \Phi.$$

Lemma 23. *(Wahrheitslemma) $\kappa \vDash \psi \iff \psi \in \Phi$.*

Beweis. Induktion über ψ per Definition und Hintikka-Eigenschaften, z.B.: $\kappa \vDash \neg\psi \iff \kappa \nvDash \psi \overset{\text{IV}}{\iff} \psi \notin \Phi \overset{\text{Hintikka}}{\iff} \neg\psi \in \Phi$ □

Korollar 24. *(Kompaktheit) Sei Φ eine Formelmenge, so dass alle endlichen Teilmengen $\Phi_0 \subseteq \Phi$ erfüllbar sind (so eine Formelmenge heißt* endlich erfüllbar*). Dann ist Φ erfüllbar.*

Beweis. Nach Vollständigkeit ist Erfüllbarkeit gleichbedeutend mit Konsistenz. Konsistenz hat offenbar die behauptete Eigenschaft: nach Negation beider Seiten ist zu zeigen, dass $\Phi \vdash \bot$ genau dann, wenn es eine endliche Teilmenge $\Phi_0 \subseteq \Phi$ gibt mit $\Phi_0 \vdash \bot$. Das ist aber klar, da ein Beweis von $\bot$ aus Φ nur endlich viele der Annahmen in Φ verwendet. □

3.2 Anwendungen des Kompaktheitssatzes

Definition 25. Ein (ungerichteter) Graph mit Knotenmenge V und Kantenmenge E heißt *k-färbbar*, wenn es eine Abbildung $c : V \to \{1, \ldots, k\}$ (*Colouring/Färbung*) gibt, so dass für jede Kante $\{v, w\} \in E$ $c(v) \neq c(w)$ gilt.

Satz 26. *Ein (möglicherweise unendlicher) Graph ist k-färbbar, wenn alle seine endlichen Untergraphen k-färbbar sind.*

Beweis. Sei V die Knotenmenge und E die Kantenmenge des Graphen. Wir führen Atome $A_{v,i}$ für $v \in V$, $i \in \{1, \ldots, k\}$ ein, mit der Lesart „Knoten v hat Farbe i“ (also $c(v) = i$). Wir definieren Formelmengen Φ_1, Φ_2, Φ_3 wie folgt:

$$\Phi_1 = \{\textstyle\bigvee_{i=1}^{k} A_{v,i} \mid v \in V\}$$

– d.h. jeder Knoten hat mindestens eine Farbe.

$$\Phi_2 = \{\neg(A_{v,i} \wedge A_{v,j}) \mid v \in V, i, j \in \{1, \ldots, k\}, i \neq j\}$$

– d.h. kein Knoten hat mehr als eine Farbe.

$$\Phi_3 = \{\neg(A_{v,i} \wedge A_{w,i}) \mid \{v, w\} \in E, i \in \{1, \ldots, k\}\}$$

– d.h. keine zwei adjazenten Knoten haben die gleiche Farbe.

Wir setzen nun $\Phi = \Phi_1 \cup \Phi_2 \cup \Phi_3$. Dann ist Φ endlich erfüllbar: Sei $\Psi \subseteq \Phi$ endliche Teilmenge; dann ist die Menge $V_0 = \{v \mid \exists i.\, A_{v,i} \in \mathrm{At}(\Psi)\}$ endlich. Sei G_0 der von V_0 aufgespannte Untergraph. Nach Annahme ist G_0 k-färbbar; sei c eine entsprechende Färbung. Wir definieren eine Wahrheitsbelegung κ_0 durch

$$\kappa_0(A_{v,i}) = \top \text{ gdw. } c(v) = i$$

für $v \in V_0$ (und beliebig auf anderen Atomen). Dann gilt $\kappa_0 \models \Phi_0$.

Nach dem Kompaktheitssatz ist also Φ erfüllbar; sei $\kappa \models \Phi$. Dann existiert für jedes v genau ein i, so dass $\kappa(A_{v,i}) = \top$; wir erhalten eine k-Färbung c des Graphen, indem wir $c(v) = i$ setzen. □

Sehr ähnlich erhält man z.B. einfache Beweise von Königs Lemma (jeder endlich verzweigende unendliche Graph hat einen unendlichen Pfad) oder der Aussage, dass man, gegeben ein Satz K von quadratischen Kacheln mit gefärbten Kanten, genau dann die $\mathbb{N} \times \mathbb{N}$-Ebene mit Kacheln aus K farblich passend anschließend überdecken kann, wenn dies für jede $n \times n$-Fläche geht.

4 Normalformen und Resolution

Wir entwickeln nunmehr ein Entscheidungsverfahren für die Erfüllbarkeit aussagenlogischer Formeln, das zwar immer noch in schlechten Fällen in exponentieller Zeit läuft (und wegen der NP-Vollständigkeit des SAT-Problems wird sich das wahrscheinlich, d.h. wenn P≠NP, grundsätzlich nicht vermeiden lassen), aber in der Praxis wesentlich schneller ist als das bisher verwendete Wahrheitstafelverfahren: das sogenannte Resolutionsverfahren. Dieses Verfahren erwartet die Eingabeformel in einem besonderen Format, der sogenannten *konjunktiven Normalform*, die wir als nächstes in zwei Schritten einführen.

4.1 Negationsnormalform (NNF)

Die ab jetzt betrachteten Normalformen sind dadurch definiert, dass sie eine bestimmte Reihenfolge der logischen Operatoren bei der Traversierung des Syntaxbaums einer Formel von der Wurzel zu den Blättern festlegen. Wir behandeln zunächst nur die Negation, von der wir verlangen, dass sie hierbei zuletzt kommt; da sich aufeinanderfolgende Negationen aufheben, läuft dies darauf hinaus, dass es Negationen nur kurz vor dem Blatt geben darf. Mit unserer bisherigen Auswahl an Basisoperatoren (nur $\neg$ und $\wedge$) lässt sich allerdings eine solche Normalform im allgemeinen nicht herstellen; wir betrachten daher ab jetzt auch $\top$ und die Disjunktion $\vee$ als Basisoperationen.

Definition 27 (Negationsnormalform)**.** Eine aus Atomen sowie $\neg, \wedge, \vee, \bot, \top$ gebildete Formel ϕ ist in *Negationsnormalform (NNF)* (oder *eine NNF*), wenn die Negation $\neg$ in ϕ nur direkt vor Atomen vorkommt, d.h. wenn ϕ von der Grammatik

$$\phi, \psi ::= \bot \mid \top \mid A \mid \neg A \mid \phi \wedge \psi \mid \phi \vee \psi \qquad (A \in \mathscr{A})$$

erzeugt wird. Eine NNF χ ist *NNF von* ϕ, wenn $\chi \equiv \phi$.

Durch wiederholte Anwendung der Gesetze

$$\begin{aligned} \neg\neg\phi &\equiv \phi \\ \neg(\phi \wedge \psi) &\equiv \neg\phi \vee \neg\psi \\ \neg(\phi \vee \psi) &\equiv \neg\phi \wedge \neg\psi \end{aligned}$$

lässt sich jeder Term, der keine Wahrheitskonstanten enthält, in NNF bringen; Details s. Übung. Ferner kann man offenbar aus jeder Formel ϕ mit $\mathrm{At}(\phi) \neq \emptyset$ alle Wahrheitskonstanten durch Umformungen wie $\phi \wedge \bot \equiv \bot$ etc. entfernen. Wir halten explizit fest:

Satz 28. *Jede Formel hat eine NNF.*

Beispiel 29 (Terme in NNF)**.** Die Formel $\neg A \vee (B \vee \neg C)$ ist in NNF, $\neg(A \vee \neg B)$ dagegen nicht.

Formal können wir eine NNF $\mathsf{NNF}(\phi)$ für eine Formel ϕ ohne Wahrheitskonstanten *rekursiv* definieren durch

$$\begin{aligned} \mathsf{NNF}(A) &= A \\ \mathsf{NNF}(\phi \wedge \pi) &= \mathsf{NNF}(\phi) \wedge \mathsf{NNF}(\pi) \\ \mathsf{NNF}(\phi \vee \pi) &= \mathsf{NNF}(\phi) \vee \mathsf{NNF}(\pi) \\ \mathsf{NNF}(\neg A) &= \neg A \\ \mathsf{NNF}(\neg\neg\phi) &= \mathsf{NNF}(\phi) \\ \mathsf{NNF}(\neg(\phi \wedge \pi)) &= \mathsf{NNF}(\neg\phi) \vee \mathsf{NNF}(\neg\pi) \\ \mathsf{NNF}(\neg(\phi \vee \pi)) &= \mathsf{NNF}(\neg\phi) \wedge \mathsf{NNF}(\neg\pi) \end{aligned}$$

Es lässt sich dann induktiv zeigen, dass $\mathsf{NNF}(\phi) \equiv \phi$.

Beispiel 30.

$$\begin{aligned} \mathsf{NNF}(\neg(\neg(A \vee \neg B) \wedge C)) &= \mathsf{NNF}(\neg\neg(A \vee \neg B)) \vee \mathsf{NNF}(\neg C) \\ &= \mathsf{NNF}(A \vee \neg B) \vee \neg C \\ &= A \vee \neg B \vee \neg C \end{aligned}$$

4.2 Konjunktive Normalformen

Wir wollen nun als nächstes zusätzlich darauf bestehen, dass bei Traversierung des Syntaxbaums eine Formel von der Wurzel zu einem Blatt stets die Konjunktionen vor den Disjunktionen kommen (und weiterhin zuletzt ggf. die Negation).

Formal definieren wir *Literale*, *Klauseln* und *konjunktive Normalformen (CNF)* durch die folgende Grammatik:

Literale

$$L ::= A \mid \neg A \qquad A \in \mathscr{A}$$

Klauseln

$$C ::= \bot \mid neC$$
$$neC ::= L \mid L \vee neC$$

CNFs

$$\phi ::= \top \mid \psi$$
$$\psi ::= C \mid C \wedge \psi$$

Hierbei sehen wir ab jetzt $\vee$ und $\wedge$ als kommutativ und assoziativ an; wir unterscheiden also z.B. nicht zwischen $(A \vee B) \vee C$ und $B \vee (C \vee A)$.

Eine CNF ist demnach eine Konjunktion von Disjunktionen von Literalen, hat also die allgemeine Form

$$\bigwedge_{i=1}^{n} (\bigvee_{j=1}^{k} L_{ij})$$

Meistens, insbesondere zum Zwecke der Repräsentation im Rechner, verwenden wir eine alternative Darstellung von CNFs als Mengen, d.h. wir abstrahieren in Konjunktionen und Disjunktionen von der Reihenfolge und Wiederholungen:

Klauseln sind endliche Mengen von Literalen, d. h. die Disjunktion $L_1 \vee \cdots \vee L_k$ wird repräsentiert als die Menge $\{L_1, \ldots, L_k\}$. Die leere Klausel repräsentiert $\bot$ und wird als $\square$ notiert.

CNFs sind endliche Mengen von Klauseln, wobei die leere Menge $\top$ repräsentiert: $\top \mathrel{\hat{=}} \emptyset$; $C_1 \wedge \cdots \wedge C_n \mathrel{\hat{=}} \{C_1, \ldots, C_n\}$

Definition 31 (CNF C einer Formel ϕ)**.** Sei ϕ eine Formel. Eine CNF ϕ' heißt *CNF von* ϕ, wenn $\phi \equiv \phi'$.

Lemma 32. *Sei ϕ eine Formel. Dann hat ϕ eine CNF* $\mathrm{CNF}(\phi)$.

Beweis. Wir können nach obigem annehmen, dass ϕ bereits in NNF ist.

Wir definieren dann $\mathsf{CNF}(\phi)$ rekursiv:

$$\mathsf{CNF}(\phi \wedge \psi) = \mathsf{CNF}(\phi) \wedge \mathsf{CNF}(\psi)$$
$$\mathsf{CNF}((\phi \wedge \psi) \vee \chi) = \mathsf{CNF}(\phi \vee \chi) \wedge \mathsf{CNF}(\psi \vee \chi)$$
$$\mathsf{CNF}(\phi) = \phi \qquad (\phi \text{ bereits CNF}).$$

Die Rekursion terminiert, weil die Argumente in den rekursiven Aufrufen stets kleiner sind als das Argument im gerade definierten Aufruf. Man zeigt durch Indukion über die Größe von ϕ, dass $\phi \equiv \mathsf{CNF}(\phi)$ und dass $\mathsf{CNF}(\phi)$ in der Tat eine CNF ist. Der einzig interessante Teil der Induktion ist der Induktionsschritt für $(\phi \wedge \psi) \vee \chi$, der auf dem Distributivgesetz

$$(\phi \wedge \psi) \vee \chi \equiv (\phi \vee \chi) \wedge (\psi \vee \chi)$$

beruht. □

Das Problem an der CNF ist, dass, wenn $\mathsf{CNF}(\phi)$ n Klauseln hat und $\mathsf{CNF}(\psi)$ k Klauseln hat, dann $\mathsf{CNF}(\phi \vee \psi)$ $n \cdot k$ Klauseln hat, was bei längeren Disjunktionen einen exponentiellen Zuwachs in der Größe bewirkt. Z.B. hat $\mathsf{CNF}((A_1 \wedge B_1) \vee \cdots \vee (A_n \wedge B_n))$ 2^n Klauseln. Durch Einführung zusätzlicher Atome lässt sich der polynomiell halten, dann ist aber $\overline{\mathsf{CNF}}(\phi)$ nur noch *erfüllbarkeitsäquivalent* zu ϕ ($\overline{\mathsf{CNF}}(\phi)$ erfüllbar $\Leftrightarrow$ ϕ erfüllbar).

Beispiel 33.

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

4.3 Resolution

Das Resolutionsverfahren ist ein Algorithmus zur Entscheidung der Erfüllbarkeit einer CNF, hier dargestellt als eine Klauselmenge ϕ. Er basiert auf der *Resolutionsregel*

$$(\text{Res})\ \frac{C_1 \cup \{A\} \qquad C_2 \cup \{\neg A\}}{C_1 \cup C_2}. \tag{1}$$

Diese Regel wird im Resolutionsverfahren auf die Menge der Klauseln in einer CNF angewendet, in dem Sinne, dass, wenn in der Menge alle Prämissen, d.h. die Klauseln über dem Strich, enthalten sind, die Konklusion, also die Klausel $C_1 \cup C_2$, hinzugefügt werden darf; letztere heißt *Resolvente* von $C_1 \cup \{A\}$ und $C_2 \cup \{\neg A\}$.

Beispiel 34.

$$\{D, B, \neg C\}, \{D, C\}, \{\neg D, B\}, \{\neg C, B, \neg A\}, \{C, B, \neg A\}, \{\neg B, \neg A\}, \{\neg B, A\}$$

Die Resolutionsregel ist im folgenden Sinn korrekt:

Lemma 35. *Sei $C_1 \cup \{A\}, C_2 \cup \{\neg A\} \in \phi$. Dann gilt: ϕ ist logisch äquivalent zu $\phi \cup \{C_1 \cup C_2\}$.*

Beweis. Zu zeigen ist hier nur, dass die Resolvente $C_1 \cup C_2$ aus ϕ folgt. Sei also $\kappa \models \phi$, dann $\kappa \models C_1 \cup \{A\}, \kappa \models C_2 \cup \{\neg A\}$. Wir führen eine Fallunterscheidung nach $\kappa(A)$ durch:

$\kappa(A) = \top$: Dann $\kappa \models C_2$, also $\kappa \models C_1 \cup C_2$

$\kappa(A) = \bot$: Dann $\kappa \models C_1$, also $\kappa \models C_1 \cup C_2$ □

Auf der Resolutionsregel basierent der folgende Algorithmus zur Entscheidung der Erfüllbarkeit einer CNF:

Algorithmus 36 (Resolutionsverfahren)**.** Eingabe: CNF ϕ. Ausgabe: „ja“, wenn ϕ erfüllbar, „nein“ sonst.

Verwende ϕ als globale Variable:

1. If $\square \in \phi$ return „nein“.
2. Suche $C_1 \cup \{A\}, C_2 \cup \{\neg A\} \in \phi$, $C_1 \cup C_2 \notin \phi$. Falls keine solchen C_1, C_2 existieren, return „ja“.
3. $\phi := \phi \cup \{C_1 \cup C_2\}$, gehe zu Schritt 1.

Beispiel 37. 1. $\{D, B, \neg C\}, \{D, C\}, \{\neg D, B\}, \{\neg C, B, \neg A\}, \{C, B, \neg A\}, \{\neg B, \neg A\}, \{\neg B, A\}$

2. $\{A, \neg B\}, \{A, B\}$

Wir beweisen die totale Korrektheit des Algorithmus:

Terminierung. Der Algorithmus arbeitet ausschließlich auf der Menge $\mathrm{At}(\phi)$ der in ϕ vorkommenden Atome, d.h. die Anwendung der Regel führt nie neue Atome ein. In Mengendarstellung gibt es nur endlich viele unterschiedliche Klauseln über $\mathrm{At}(\phi)$, nämlich $4^{|\mathrm{At}(\phi)|}$ (jedes Atom kann sowohl als positives als auch als negatives Literal jeweils vorkommen oder nicht vorkommen). Insbesondere kann also nur endlich oft eine Klausel hinzugefügt werden, die nicht bereits in der aktuellen CNF enthalten ist. □

Korrektheit (Antwort „nein“ ist richtig). Klar per Korrektheit der Resolutionsregel (Lemma 35). □

Wir sagen, dass ϕ *resolutionsabgeschlossen (ra.)* ist, wenn mit $C_1 \cup \{A\}, C_2 \cup \{\neg A\} \in \phi$ stets $C_1 \cup C_2 \in \phi$ gilt. Der Algorithmus bricht also genau dann bei Schritt 2 ab, wenn ϕ ra. ist.

Vollständigkeit (Antwort „ja“ ist richtig). Sei ϕ ra., $\square \notin \phi$. ZZ: ϕ erfüllbar.

Induktion über $|\mathrm{At}(\phi)|$:

Induktionsanfang: Ohne Atome kann ϕ als Klauselmenge nur leer (also wahr) sein, da die einzige Klausel ohne Atome, nämlich $\square$, n.V. nicht in ϕ enthalten ist.

Induktionsschritt: Wähle $A \in \mathrm{At}(\phi)$. Sei

$$\phi/A = \{\, D \setminus \{\neg A\} \mid A \notin D \in \phi \,\}$$
$$\phi/\neg A = \{\, D \setminus \{A\} \mid \neg A \notin D \in \phi \,\}.$$

ϕ/A ist logisch äquivalent zu ϕ, wenn wir A annehmen, entsprechend für $\phi/\neg A$ und $\neg A$. Es gilt $\mathrm{At}(\phi/A) \not\ni A \notin \mathrm{At}(\phi/\neg A))$.

Dann ist entweder $\square \notin \phi/A$ oder $\square \notin \phi/\neg A$: sonst $\{\neg A\} \in \phi \ni \{A\}$; da ϕ ra., würde folgen $\square \in \phi$, Widerspruch.

Ohne Einschränkung[2] sei $\square \notin \phi/A$.

[2] „ohne Einschränkung“ ist gleichbedeutend mit (aber kürzer als) „ohne Beschränkung der Allgemeinheit“. Konkret heißt es hier, dass der Beweis für den Fall $\square \notin \phi/\neg A$ genauso geführt werden kann und deshalb weggelassen wird.

Nun ist auch ϕ/A ra.: Sei $E_1 \cup \{B\}, E_2 \cup \neg\{B\} \in \phi/A$ (insbesondere $B \neq A$). Dann ex. $C_1 \cup \{B\}, C_2 \cup \{\neg B\} \in \phi$ mit $E_1 \cup \{B\} = C_1 \cup \{B\} \setminus \{\neg A\}$, $E_2 \cup \{\neg B\} = C_2 \cup \{\neg B\} \setminus \{\neg A\}$. Da ϕ ra., folgt $C_1 \cup C_2 \in \phi$; da $A \notin C_1 \cup C_2$, folgt $E_1 \cup E_2 = C_1 \cup C_2 \setminus \{\neg A\} \in \phi/A$.

Da $|At(\phi/A)| < |\operatorname{At}(\phi)|$, folgt nach IV, dass ϕ/A erfüllbar ist, d.h. es existiert κ mit $\kappa \models \phi/A$. Wir behaupten nun, dass $\kappa[A \to \top] \models \phi$, womit dann ϕ wie verlangt erfüllbar ist. Sei also $D \in \phi$; zu zeigen ist $\kappa[A \to \top] \models D$.

Fall 1: $A \in D$ ✓

Fall 2: $A \notin D \Rightarrow D \setminus \{\neg A\} \in \phi/A \Rightarrow \kappa \models D \setminus \{\neg A\} \Rightarrow \kappa[A \to \top] \models D$.

□

5 Prädikatenlogik erster Stufe

Ein *Prädikat* ist eine Eigenschaft, die Individuen haben (oder nicht haben) können, d.h. eine Aussage mit Parametern, für die Individuen einzusetzen sind; Beispiel: $\mathsf{positiv}(n)$ oder $\mathsf{verheiratetMit}(\mathsf{x}, \mathsf{y})$. Diese Parametrisierung unterscheidet Prädikate von den Atomen der Aussagenlogik, die nur schlechthin wahr oder falsch sein können (wie z.B. $\mathsf{esRegnet}$). Logiken, die über Prädikate sprechen, heißen *Prädikatenlogiken*. Sie beinhalten typischerweise auch die Möglichkeit, Aussagen zu *quantifizieren*, typischerweise dahingehend, ob sie *für alle* möglichen Belegungen von Variablen oder für *mindestens eine* Belegung gelten. Wir beschäftigen uns hier nur mit dem Fall, in dem die betreffenden Variablen für Individuen stehen, also mit der Prädikatenlogik *erster Stufe*. (Es gibt auch Logiken, die Variablen für komplexere Gebilde, z.B. für Mengen von Individuen, zulassen; man spricht dann von Prädikatenlogik *höherer Stufe*, im Beispielfall *zweiter Stufe*.) Gelegentlich wird einfach der Begriff „Prädikatenlogik“ verwendet, womit meist die Prädikatenlogik erster Stufe gemeint ist. Letztere wird auf Englisch mit dem deutlich kürzeren Begriff *first order logic* bezeichnet, abgekürzt *FOL*; der Knappheit halber werden wir oft diese Abkürzung verwenden.

Definition 38 (Syntax der Prädikatenlogik erster Stufe)**.** Die Syntax der Prädikatenlogik hängt ab von einem Vorrat an Symbolen für Konstanten sowie für Prädikate und Funktionen gegebener Stelligkeit. Dieser Symbolvorrat wird oft als die *Sprache* bezeichnet; wir bevorzugen hier zur Vermeidung von Verwechslungen den Ausdruck *Signatur*. Formal besteht eine Signatur $\Sigma = (P_\Sigma, F_\Sigma)$ aus

- einer Menge P_Σ von *Prädikatensymbolen* und
- einer Menge F_Σ von *Funktionssymbolen*.

Jedes Funktions- oder Prädikatensymbol s hat eine endliche *Stelligkeit* $\operatorname{ar}(s) \geq 0$. Für $s \in P_\Sigma \cup F_\Sigma$ und $\operatorname{ar}(s) = n$ schreiben wir oft $s/n \in \Sigma$, wobei wir typischerweise dadurch die Unterscheidung zwischen Funktionen- und Prädikatensymbolen sicherstellen, dass wir erstere mit Kleinbuchstaben und letztere mit Großbuchstaben bezeichnen. Ein nullstelliges Funktionssymbol $c/0 \in \Sigma$ heißt auch *Konstante*; nullstellige Prädikatensymbole entsprechen gerade den Atomen der Aussagenlogik.

Diese Daten bestimmen nun zunächst den Begriff des *Terms*: Wir unterstellen einen Vorrat V an Variablen; Terme $E, D, \ldots$ sind dann gegeben durch die BNF

$$E ::= x \mid f(E_1, \ldots, E_n) \qquad (x \in V, f/n \in \Sigma).$$

Die Syntax von Formeln ist dann gegeben durch die folgende BNF:

$$\phi ::= (E = D) \mid P(E_1, \ldots, E_n) \mid \neg\phi \mid \phi_1 \wedge \phi_2 \mid \forall x.\phi \qquad (x \in V, P/n \in \Sigma).$$

Die durch die ersten zwei Klauseln konstruierten Formeln heißen *atomare Formeln*. Der Existenzquantor $\exists$ wird dann definiert durch $\exists x.\phi := \neg\forall x.\neg\phi$, außerdem $\bot \equiv \neg\forall x.x = x$. Weitere aussagenlogische Junktoren wie Implikation, Disjunktion etc. werden wie gewohnt definiert.

Die Sprechweise für $\forall x.\phi$ ist „für alle x gilt ϕ“, und für $\exists x.\phi$ „es existiert ein x, so dass ϕ“ oder „es existiert ein x mit ϕ“.

Beispiel 39 (Wissenswertes über Erlangen)**.** Der Satz „Jeder Erlanger mag einen anderen Erlanger“ wird in Logik erster Stufe ausgedrückt als

$$\forall x.\,(E(x) \to \exists y.\,(E(y) \wedge L(x, y)))$$

(mit E für „Erlanger“ und L für „likes“). Der etwas unwahrscheinlicher klingende Satz „Es gibt einen Erlanger, den alle Erlanger mögen“ dagegen wird formalisiert als

$$\exists y.\,(E(y) \wedge \forall x.\,(E(x) \to L(x, y))).$$

Diese Beispiele illustrieren zum einen die typische und meist allein sinnvolle Kombination von $\forall$ mit $\to$ und von $\exists$ mit $\wedge$ gemäß den klassischen *aristotelischen Formen*:

$$\begin{aligned} \text{Alle Ps sind Qs:} &\qquad \forall x.\,(P(x) \to Q(x)) \\ \text{Einige Ps sind Qs:} &\qquad \exists x.\,(P(x) \wedge Q(x)). \end{aligned}$$

Zum anderen sieht man hier, obwohl wir natürlich noch keinerlei formale Bedeutung von Formeln diskutiert haben, schon anhand der Lesart, dass man $\exists$ und $\forall$ im allgemeinen keineswegs bedeutungserhaltend vertauschen kann.

Wir sehen die Variable x in $\forall x.\,\phi$ als durch den Quantor *gebunden* an, d.h. ihre Lebensdauer ist begrenzt auf die Unterformel ϕ. Wenn $\forall x.\,\phi$ Teil einer größeren Formel ist, die außerhalb von $\forall x.\,\phi$ die Variable x bereits erwähnt, so wird die äußere Verwendung von x durch den Quantor verschattet. Diese Phänomene sind letztlich ähnlich der Verwendung lokaler Variablen in üblichen Programmiersprachen.

Formal definieren wir die *freien Variablen* in einem Term oder einer Formel wie folgt; alle Variablen, die nicht nach dieser Definition frei sind, sind *gebunden*.

Definition 40 (Freie Variable)**.** Die Mengen $FV(E)$ und $FV(\phi)$ der freien Variablen eines Terms E bzw. einer Formel ϕ sind rekursiv definiert durch

$$\begin{aligned} FV(x) &= \{x\} \\ FV(f(E_1, \ldots, E_n)) &= \bigcup_{i=1}^{n} FV(E_i) \\ FV(E = D) &= FV(E) \cup FV(D) \\ FV(P(E_1, \ldots, E_n)) &= \bigcup_{i=1}^{n} FV(E_i) \\ FV(\neg\phi) &= FV(\phi) \\ FV(\phi \wedge \psi) &= FV(\phi) \cup FV(\psi) \\ FV(\forall x.\phi) &= FV(\phi) \setminus \{x\} \end{aligned}$$

Z.B. gilt $FV(x = y \wedge \forall y.y = w) = \{x, y, w\}$.

Definition 41 (Satz)**.** Eine Formel ϕ heißt *Satz*, wenn $FV(\phi) = \emptyset$, wenn also alle ihre Variablen gebunden sind.

Wir sehen die Namen gebundener Variablen als unwichtig an und sehen daher zwei Formeln, die sich nur durch die Namen gebundener Variablen unterscheiden (also α-*äquivalent* sind) als im wesentlichen gleich an; z.B. sind $\forall x.\, x = z$ und $\forall y.\, y = z$ (nicht aber $\forall z.\, z = z$!) im wesentlichen dieselbe Formel.

Definition 42 (Substitution, Umbenennung)**.** Eine *Substitution* ist eine Abbildung σ, die jeder Variablen x einen Term $\sigma(x)$ zuordnet, so dass die Menge

$$\mathrm{Dom}(\sigma) = \{x \mid \sigma(x) \neq x\}$$

endlich ist, d.h. σ verändert nur eine endliche Anzahl an Variablen. $\mathrm{Dom}(\sigma)$ („Domain“, „Bereich“ von σ) ist die Menge aller Variablen, mit denen σ „etwas tut“, d.h. denen σ einen anderen Term zuordnet. Mit $E\sigma$ bezeichnen wir die Anwendung von σ auf den Term E. Dabei wird jede Variable wie in σ angegeben ersetzt: $x\sigma = \sigma(x)$, $f(E_1, \ldots, E_n)\sigma = f(E_1\sigma, \ldots, E_n\sigma)$. Die Anwendung $\phi\sigma$ einer Substitution σ auf eine Formel ϕ ist wegen eventuell gebundener Variablen komplizierter:

$$\begin{aligned}(E = D)\sigma &= (E\sigma = D\sigma)\\ P(E_1, \ldots, E_n)\sigma &= P(E_1\sigma, \ldots, E_n\sigma)\\ (\neg\phi)\sigma &= \neg(\phi\sigma)\\ (\phi \wedge \psi)\sigma &= \phi\sigma \wedge \psi\sigma\\ (\forall x.\, \phi)\sigma &= \forall y.\, \phi\sigma',\end{aligned}$$

wobei $\sigma'(x) = y$, $\sigma'(z) = \sigma(z)$ für jedes $z \neq x$, und wobei ferner y so gewählt ist, dass $y \notin FV(\sigma(z))$ für alle $z \in FV(\forall x.\, \phi)$ („y ist frisch“).

Eine Substitution σ heißt eine *Umbenennung*, wenn $\sigma(x)$ für alle x eine Variable ist.

Die Substitution $[E_1/x_1, \ldots, E_n/x_n]$ ist die Substitution σ mit

$$\sigma(x) = \begin{cases} E_i & \text{wenn } x = x_i \\ x & \text{sonst} \end{cases}$$

Die Identität wird denotiert durch die leere Substitution $[\,]$, also $E\,[\,] \equiv E$. Für Substitutionen σ, τ bezeichnet $\sigma\tau$ die Substitution mit $(\sigma\tau)(x) = (\sigma(x))\tau$, d.h. die Substitution, die entsteht, wenn man zuerst σ und dann τ ausführt.

Beispiel 43. Mit $\sigma = [add(z, y)/x, suc(x)/y]$ haben wir $\mathrm{Dom}(\sigma) = \{x, y\}$ und z.B.

$$add(add(x, z), add(x, y))\sigma = add(add(add(z, y), z), add(add(z, y), suc(x)).$$

Dieses Beispiel illustriert insbesondere, dass die Ersetzung von x und y durch σ definitionsgemäß *gleichzeitig* stattfindet; wenn man zuerst die Substitution $[add(z, y)/x]$ und anschließend die Substitution $[suc(x)/y]$ anwendet, kommt etwas anderes heraus, da dann auch im für x eingesetzten Term $add(z, y)$ die Variable y durch $suc(x)$ ersetzt wird.

Als weiteres Beispiel betrachten wir die Formel $\phi = \forall x.\, (x = y)$; wenn wir uns x, y etc. (nur für dieses Beispiel!) als Zahlen vorstellen, ist dies die (natürlich nicht übermäßig sinnvolle) Aussage,

dass y die einzige Zahl ist. Ferner sei σ die Substitution $[z/x, x+y/y]$. Die Seitenbedingung in der Definition der Anwendung $\phi\sigma$ von σ auf ϕ verhindert, dass in dieser Situation y als neue Variable gewählt wird, da y nicht frisch ist: y kommt frei im Term $x+y$ vor, den σ für die freie Variable y in $\forall x.\,(x = y)$ substituiert. Wenn wir uns über diese Beschränkung hinwegsetzen, ergibt sich denn auch eine Formel, die offenbar nicht im beabsichtigten Sinne eine Instanz von ϕ ist (formaler wird es gerade noch nicht, da wir noch keine Semantik definiert haben): Mit $\sigma'(x) = y$ und $\sigma'(y) = \sigma(y) = x + y$ erhielten wir hier als Resultat der Substitution

$$\forall y.\,((x = y)\sigma') = \forall y.\,(y = x + y),$$

also die gänzlich unverwandte Aussage, dass x linksneutrales Element der Addition ist.

Wählen wir dagegen wie vorgeschrieben eine frische Variable, beispielsweise z (das ist wirklich frisch im verlangten Sinn: z kommt zwar im Term $\sigma(x)$ vor, aber x ist nicht frei in ϕ), so erhalten wir mit $\sigma'(x) = z$, $\sigma'(y) = \sigma(y) = x + y$ korrekterweise

$$\phi\sigma = (\forall x.\,(x = y))\sigma = \forall z.\,((x = y)\sigma') = \forall z.\,(z = x + y),$$

also die Aussage, dass $x + y$ die einzige Zahl ist.

5.1 Natürliches Schließen in Prädikatenlogik

Wir erweitern nun das System des natürlichen Schließens um Regeln für die neuen Ausdrucksmittel $\exists$, $\forall$ und $=$. Die Regeln für Gleichheit sind

$$(= I)\ \frac{}{E = E} \qquad (= E)\ \frac{\phi[E/x] \qquad E = D}{\phi[D/x]}.$$

Damit beweist man andere erwartete Eigenschaften von Gleichheit, etwa Symmetrie

1	$E = D$	
2	$E = E$	(=I)
3	$D = E$	(=E) 1,2

und Transitivität

1	$E = D$	
2	$D = F$	
3	$E = F$	(=E) 1,2

Die Regeln für $\forall$ formalisieren bekannte Schlussweisen: Die $\forall$-Elimination lautet

$$\forall\mathrm{E}\ \frac{\forall x.\,\phi}{\phi[E/x]}$$

also umgangssprachlich „Wenn für alle x ϕ gilt, dann gilt ϕ auch für E“.

Die Einführungsregel liest sich umgangssprachlich „Wenn man ϕ für ein beliebiges c zeigen kann, dann gilt ϕ für alle x“, formal

$$(\forall\mathrm{I})\ \frac{\begin{array}{|l} \boxed{c} \\ \hline \vdots \\ \phi[c/x] \end{array}}{\forall x.\,\phi}$$

wobei c eine *frische* Konstante ist, d.h. außer in dem betreffenden Unterbeweis nirgendwo vorkommt (dies wird durch die Box um das c angedeutet).

Die Einführungsregel für $\exists$ ist ebenso intuitiv:

$$(\exists \mathrm{I})\ \frac{\phi[E/x]}{\exists x.\, \phi}$$

In Worten: Wenn ϕ für E gilt, dann gibt es ein x, für das ϕ gilt.

Etwas gewöhnungsbedürftiger ist die Eliminationsregel:

$$(\exists \mathrm{E})\ \frac{\exists x.\phi \qquad \begin{array}{|l} \boxed{c}\ \phi[c/x] \\ \hline \vdots \\ \psi \end{array}}{\psi}$$

wiederum mit der Seitenbedingung, dass c frisch ist; dies bedeutet hier insbesondere, dass c nicht in ψ vorkommen darf. In Worten: Wenn es ein x gibt, für das ϕ gilt, und wenn ich für beliebiges c eine Aussage ψ daraus folgern kann, dass ϕ für c gilt, dann gilt ψ.

Herleitung der Regen für $\exists$ Als erste Anwendung der Regeln für $\forall$ können wir die Regeln für $\exists$ bei Kodierung von $\exists$ per $\exists x.\, \phi \equiv \neg\forall x.\, \neg\phi$ herleiten:

Beweis von ($\exists I$)

1	$\phi[c/x]$	
2	$\quad \forall x.\, \neg\phi$	
3	$\quad \neg\phi[c/x]$	($\forall$E, 2)
4	$\quad \bot$	($\bot$I 1, 3)
5	$\neg\forall x.\, \neg\phi$	($\neg$I 1-4)
6	$\exists x.\, \phi$	

Beweis von $(\exists E)$

1	$\neg\forall x.\,\neg\phi$	
2	$\quad\neg\psi$	
3	$\quad\boxed{\text{c}}$	
4	$\quad\phi[c/x]$	
5	$\quad\vdots$	
6	$\quad\psi$	(Unterbeweis aus Prämisse)
7	$\quad\bot$	($\bot$I) 2,6
8	$\neg\phi[c/x]$	($\neg$I) 4–7
9	$\forall x.\,\neg\phi$	($\forall$I) 3–8
10	$\bot$	($\bot$I) 1,9
11	$\neg\neg\psi$	($\neg$I) 2–10
12	ψ	($\neg$E) 11

Weitere Beispiele für Herleitungen, jetzt im vollen Kalkül (also mit den Regeln für $\exists$):

Folgerung der Existenz aus der Allquantifiziertheit $(\forall x.\,P(x)) \to (\exists x.\,P(x))$

1	$\forall x.\,P(x)$	
2	$P(c)$	($\forall$E, 1)
3	$\exists x.\,P(x)$	($\exists$I, 2)

Transitivität der Subsumption

1	$\forall x.\,(P(x) \to Q(x))$	
2	$\forall z.\,(Q(z) \to R(z))$	
3	$\quad\boxed{\text{d}}\quad P(d)$	
4	$\quad P(d) \to Q(d)$	($\forall$E) 1
5	$\quad Q(d)$	($\to$E) 3, 4
6	$\quad Q(d) \to R(d)$	($\forall$E) 2
7	$\quad R(d)$	($\to$E) 5, 6
8	$\forall x.\,(P(x) \to R(x))$	($\forall$I*) 3–7

(Nicht-)-Vertauschung von $\forall$ **und** $\exists$ Wir können aus $\exists x.\,\forall y.\,P(x,y)$ die Formel $\forall y.\,\exists x.\,P(x,y)$ folgern. Die Umkehrung gilt nicht, wie man sich anhand der bisher nur informell beschriebenen Bedeutung der Formeln klarmacht; wir werden dies später auch formal zeigen.

1	$\exists x.\, \forall y.\, P(x,y)$	
2	$\boxed{\text{c}}$ $\forall y.\, P(c,y)$	
3	$\boxed{\text{d}}$	
4	$P(c,d)$	($\forall$E) 2
5	$\exists x.\, P(x,d)$	($\exists$I) 4
6	$\forall y.\, \exists x.\, P(x,y)$	($\forall$I) 3–5
7	$\forall y.\, \exists x.\, P(x,y)$	($\exists$E) 2–6

Noch mehr Beispiele:

$$\begin{aligned}
\exists x.\, (\phi \vee \psi) &\equiv \exists x.\, \phi \vee \exists x.\, \psi \\
\phi \wedge \exists x.\, \psi &\equiv \exists x.\, (\phi \wedge \psi) \qquad (x \notin FV(\phi) \\
\phi \vee \exists x.\, \psi &\equiv \exists x.\, (\phi \vee \psi) \qquad (x \notin FV(\phi)
\end{aligned}$$

5.2 Semantik

Wir legen nunmehr formal die *Bedeutung* prädikatenlogischer Formeln fest. Ein verbreitetes Prinzip der Logik ist, dass Bedeutung hierbei die Gestalt eines Begriffs von *Erfülltheit* von Formeln relativ zu einem *Modell* annimmt, d.h. die Semantik ist eine binäre Relation $\models$ zwischen Modellen und Formeln. Im Falle der Aussagenlogik war ein Modell schlicht und einfach eine Wahrheitsbelegung κ. Abhängig von einer gegebenen Signatur Σ definieren wir nun den Begriff des Σ-*Modells* (weitere geläufige Begriffe: Σ-Struktur, Σ-Algebra, FO-Modell, FO-Struktur, Interpretation). Ein solches Modell muss offenbar hinreichend Struktur für die Interpretation von Funktionen- und Prädikatensymbolen zur Verfügung stellen.

Definition 44 (Σ-Modell)**.** Ein Σ-*Modell* $\mathfrak{M}$ besteht aus

- einer Menge M (dem *Universum*, *(Grund)bereich* oder *Träger*);
- einer *Interpretation* in $\mathfrak{M}$ für jedes n-stellige Funktionssymbol $f/n \in \Sigma$, gegeben durch eine Funktion $\mathfrak{M}[\![f]\!] : M^n \to M$
- einer *Interpretation* in $\mathfrak{M}$ für jedes n-stelliges Prädikatensymbol $P/n \in \Sigma$, gegeben durch eine Teilmenge $\mathfrak{M}[\![P]\!] \subseteq M^n$.

Eine *Umgebung* η (in $\mathfrak{M}$) ist eine Abbildung $\eta : V \to M$.

Da Konstanten einfach nullstellige Funktionssymbole sind, werden sie durch Abbildungen $M^0 \to M$ interpretiert. Da es genau ein leeres Tupel ( ) gibt, läuft dies einfach auf die Auswahl eines Elements von M hinaus; wir werden daher im folgenden so tun, als würden Konstanten einfach durch Elemente von M interpretiert, wenn dies die Schreibweise vereinfacht.

Definition 45 (Interpretation von Termen, Erfülltheit)**.** Zu einem Term E definieren wir seine *Interpretation*

$$\mathfrak{M}[\![E]\!]\eta \in M$$

rekursiv durch

$$\begin{aligned}\mathfrak{M}[\![X]\!]\eta &= \eta(X)\\ \mathfrak{M}[\![f(E_1,\ldots,E_n)]\!]\eta &= \mathfrak{M}[\![f]\!]\,(\mathfrak{M}[\![E_1]\!]\eta,\ldots,\mathfrak{M}[\![E_n]\!]\eta)\end{aligned}$$

Die Erfülltheit einer Formel φ (bestehend aus n Termen) auf einem Modell und einer Umgebung $\mathfrak{M},\eta \models \varphi$ ist rekursiv definiert durch

$$\begin{aligned}\mathfrak{M},\eta \models (E=D) &\iff \mathfrak{M}[\![E]\!]\eta = \mathfrak{M}[\![D]\!]\eta\\ \mathfrak{M},\eta \models P(E_1,\ldots,E_n) &\iff (\mathfrak{M}[\![E_1]\!]\eta,\ldots,\mathfrak{M}[\![E_n]\!]\eta) \in \mathfrak{M}[\![P]\!]\\ \mathfrak{M},\eta \models \forall X.\varphi &\iff \text{für alle } m\in M \text{ gilt } \mathfrak{M},\eta[X\mapsto m]\models\varphi\end{aligned}$$

und durch die erwarteten Klauseln für die booleschen Fälle. Dabei bezeichnet $\eta[X\mapsto m]$ die Umgebung mit

$$\eta[X\mapsto m](Y) = \begin{cases} m & (Y=X)\\ \eta(Y) & (\text{sonst}).\end{cases}$$

Beispiel 46. Wir betrachten $\Sigma = \{z/0, s/1, O/1\}$, also eine Signatur, die aus einer Konstante z (*zero*), einem einstelligen Funktionssymbol s (*successor*) und einem einstelligen Prädikat O (*odd*) besteht, und suchen nach Σ-Modellen, die den Satz

$$\phi = (O(z) \wedge \forall x.\, O(x) \leftrightarrow \neg O(s(x))$$

erfüllen. (Hierfür spielt die Umgebung offenbar zunächst keine Rolle; wir beweisen dies weiter unten auch formal.)

- $M=\mathbb{N}$, $\mathfrak{M}[\![z]\!]=0$, $\mathfrak{M}[\![s]\!](n)=n+1$, $\mathfrak{M}[\![O]\!]=\{2n+1 \mid n\in M\}$.
- $M=\{0,\ldots,2n-1\}$, $\mathfrak{M}[\![z]\!]=0$, $\mathfrak{M}[\![s]\!](x)=x+1$ für $x<2n-1$, $\mathfrak{M}[\![2n-1]\!]=0$, $\mathfrak{M}[\![O]\!]=\{2i-1 \mid 1\le i\le n\}$.
- $M=\mathbb{N}\times\mathbb{N}$, $\mathfrak{M}[\![z]\!]=(0,0)$, $\mathfrak{M}[\![s]\!](n,k)=(n+1,k)$, $\mathfrak{M}[\![O]\!]=\{(n,k) \mid n+k \text{ ungerade }\}$.

Nun wollen wir zeigen, dass die freien Variablen das tun, was wir haben wollten, als wir sie definiert haben, dass nämlich eine Formel höchstens von den in ihr vorkommenden freien Variablen abhängt. („Höchstens“, weil es z.B. auch Tautologien gibt, die von ihren freien Variablen nicht abhängen, wie etwa $\forall X.\, X=Y \vee \neg X=Y$. [3])

Lemma 47. *Sei $\eta_1(X)=\eta_2(X)$ für alle $X\in FV(\varphi)$; dann gilt $\mathfrak{M},\eta_1\models\varphi \iff \mathfrak{M},\eta_2\models\varphi$.*

Beweis. Da in Formeln Terme vorkommen, müssen wir zunächst eine entsprechende Aussage für Terme beweisen, nämlich, dass für alle Terme E und alle Umgebungen η_1, η_2 mit $\eta_1(X)=\eta_2(X)$ für alle $X\in FV(E)$

$$\mathfrak{M}[\![E]\!]\eta_1 = \mathfrak{M}[\![E]\!]\eta_2 \tag{2}$$

[3]Hier ist als Klammerung natürlich $\neg(X=Y)$ zu lesen, die Klammerung $(\neg X)$ wäre syntaktisch nicht wohlgeformt.

gilt. Der Beweis per Induktion über E wird dem Leser überlassen.

Wir beweisen nun die Aussage des Lemmas durch Induktion über φ: Atomare Formeln werden per (2) abgehandelt, z.B. $\mathfrak{M}, \eta_1 \models E = D \iff \mathfrak{M}[\![E]\!]\eta_1 = \mathfrak{M}[\![D]\!]\eta_1 \overset{(2)}{\iff} \mathfrak{M}[\![E]\!]\eta_2 = \mathfrak{M}[\![D]\!]\eta_2 \iff \mathfrak{M}, \eta_2 \models E = D$. Die Booleschen Fälle sind trivial. Es bleibt der Allquantor:

$$\begin{aligned} \mathfrak{M}, \eta_1 \models \forall X.\psi &\iff \text{für alle } m \in M \text{ gilt } \mathfrak{M}, \eta_1[X \mapsto m] \models \psi \\ &\overset{\text{IV}}{\iff} \text{für alle } m \in M \text{ gilt } \mathfrak{M}, \eta_2[X \mapsto m] \models \psi \\ &\iff \mathfrak{M}, \eta_2 \models \forall X.\psi \end{aligned}$$

Zur Anwendung der Induktionsvoraussetzung brauchen wir hierbei, dass

$$\eta_1[X \mapsto m](Y) = \eta_2[X \mapsto m](Y) \text{ für alle } Y \in FV(\psi) \subseteq FV(\forall X.\psi) \cup \{X\}.$$

Dies zeigt man durch Fallunterscheidung über Y: Wenn $Y \in FV(\forall X.\psi)$, dann gilt $\eta_1[X \mapsto m](Y) = \eta_1(Y)$, da dann $Y \neq X$, entsprechend für η_2, und $\eta_1(Y) = \eta_2(Y)$ gilt nach Voraussetzung. Falls $Y = X$, so gilt $\eta_1[X \mapsto m](Y) = m = \eta_2[X \mapsto m](Y)$. □

Damit ist für einen *Satz* φ (der ja keine freien Variablen hat) $\mathfrak{M}, \eta \models \varphi$ von η unabhängig; wir schreiben in diesem Fall kurz $\mathfrak{M} \models \varphi$ für $\mathfrak{M}, \eta \models \varphi$.

Analog wie schon für aussagenlogische Formeln schreiben wir für eine Menge Φ von Formeln $\mathfrak{M}, \eta \models \Phi$, wenn $\mathfrak{M}, \eta \models \varphi$ für alle $\varphi \in \Phi$. Logische Konsequenz, Erfüllbarkeit, Gültigkeit und logische Äquivalenz werden analog wie bisher definiert, bis auf Ersetzung der Wahrheitsbelegung κ durch $\mathfrak{M}, \eta$. Z.B. gilt per Definition $\Phi \models \psi$ (ψ ist logische Folgerung aus Φ) dann, wenn aus $\mathfrak{M}, \eta \models \Phi$ stets $\mathfrak{M}, \eta \models \psi$ folgt. Wie bisher gilt $\Phi \models \psi$ genau dann, wenn $\Phi \cup \{\neg\psi\}$ unerfüllbar ist.

Beispiel 48 (Logische Konsequenz)**.** $\exists X.\, \forall Y.\, \mathrm{Loves}(Y, X) \models \forall Y.\, \exists X.\, L(Y, X)$

Beweis. Aus der Annahme folgt, dass $m \in M$ existiert mit

$$\mathfrak{M}, [X \mapsto m] \models \forall Y.\, L(Y, X). \tag{3}$$

Sei $n \in M$; zu zeigen ist $\mathfrak{M}, [Y \mapsto n] \models \exists X.\, L(Y, X)$. Wegen (3) gilt $\mathfrak{M}, [Y \mapsto n, X \mapsto m] \models L(Y, X)$, woraus die Behauptung folgt. □

Andererseits gilt $\forall Y.\, \exists X.\, L(Y, X) \not\models \exists X.\, \forall Y.\, L(Y, X)$: Ein Gegenmodell ist z.B. $M = \{0, 1\}$, $\mathfrak{M}[\![L]\!] = \{(1, 0), (0, 1)\}$.

In Beispielen wie dem obigen ist eine Sichtweise hilfreich, in der man die Erfülltheit einer Formel über das Gewinnen eines Spiels zwischen zwei Spielern, *Eloise* (∃) und *Abaelard* (∀) definiert. Eloise versucht zu zeigen, dass eine Formel erfüllt ist, und Abaelard, dass sie nicht erfüllt ist. Spielpositionen sind Paare (η, ϕ), wobei η eine Umgebung in $\mathfrak{M}$ und ϕ eine Formel in NNF (also mit Negation nur vor atomaren Formeln) ist; das Spiel ist beendet, wenn eine atomare Formel oder deren Negation erreicht ist (für die man dann unmittelbar im Modell nachsehen kann, wer gewonnen hat). Regeln des Spiels sind z.B.

- $\phi \wedge \psi$: ∀ ist dran und wählt ϕ oder ψ als neue Formel
- $\phi \vee \psi$: ∃ ist dran und wählt ϕ oder ψ als neue Formel

- $\forall X.\,\phi$: $\forall$ ist dran und wählt ein Element des Grundbereichs als neuen Wert für X (in η)
- $\exists X.\,\phi$: $\exists$ ist dran und wählt ein Element des Grundbereichs als neuen Wert für X

Atomare Formeln und deren Negationen sind Gewinnpositionen für einen der Spieler, je nachdem, ob die Formel eben gerade gilt oder nicht; z.B. gewinnt Eloise die Position $(\eta, L(X,Y))$, wenn $\mathfrak{M}, \eta \models L(X,Y)$ (d.h. wenn $(\eta(X), \eta(Y)) \in \mathfrak{M}[\![L]\!]$). Der entscheidende Unterschied zwischen $\forall Y.\,\exists X.\,L(Y,X)$ und $\exists X.\,\forall Y.\,L(Y,X)$ ist in dieser Sichtweise, dass im Erfülltheitsspiel für $\forall Y.\,\exists X.\,L(Y,X)$ Abaelard den ersten Zug macht, in dem für $\exists X.\,\forall Y.\,L(Y,X)$ dagegen Eloise.

$\rightarrow$ Welche Regeln müsste man für nichtatomare Negation einführen?

Beispiel 49 (Logische Äquivalenz)**.**

$$\neg\forall X.\varphi \equiv \exists X.\neg\varphi \qquad \neg\exists X.\varphi \equiv \forall X.\neg\varphi$$

Mittels der Äquivalenzen im Beispiel lässt sich die Konstruktion der NNF auf prädikatenlogische Formeln verallgemeinern; dazu müssen natürlich $\forall$ und $\exists$ als vollberechtigte Bestandteile der Grammatik von Formeln angesehen werden (ebenso wie schon im aussagenlogischen Fall $\wedge$ und $\vee$). Wir werden später noch weitergehende Normalformresultate behandeln.

Leicht zu zeigen, aber zentral ist

Lemma 50 (Substitutionslemma)**.** *Es gilt*

$$\mathfrak{M}, \eta \models \varphi\sigma \iff \mathfrak{M}, \eta_\sigma \models \varphi,$$

wobei $\eta_\sigma(X) = \mathfrak{M}[\![\sigma(X)]\!]\eta$ *für* $X \in V$*.*

(Man beachte, dass $\eta_\sigma(X) = \eta(X)$ für $X \notin \mathrm{Dom}(\sigma)$: dann haben wir nämlich $\mathfrak{M}[\![\sigma(X)]\!]\eta = \mathfrak{M}[\![X]\!]\eta = \eta(X)$.)

Beweis. Induktion über φ. Die benötigte Variante der Aussage für Terme ist

$$\mathfrak{M}[\![E\sigma]\!]\eta = \mathfrak{M}[\![E]\!]\eta_\sigma.$$

□

Wir halten abschließend fest, dass unser formales Deduktionssystem zur gerade definierten Semantik passt:

Satz 51 (Korrektheit)**.** *Wenn* $\Phi \vdash \psi$*, dann auch* $\Phi \models \psi$*.*

Beweis. Wie schon im Fall der Aussagenlogik führen wir den Beweis über die Länge der Herleitung von ψ aus Φ, mit Fallunterscheidung über die zuletzt angewandte Regel und Verallgemeinerung der Induktionsbehauptung auf in Unterbeweisen aus lokalen Annahmen hergeleitete Aussagen. Die Regeln für Gleichheit sind einfach; die Regeln für $\exists$ sind aus denen für $\forall$ hergeleitet, so dass wir sie hier ignorieren können. Wir behandeln die Eliminationsregel für $\forall$, um die Verwendung des Substitutionslemmas zu illustrieren: Wenn im letzten Beweisschritt bei lokalen Annahmen Φ' die Formel $\psi[X \mapsto E]$ aus $\forall X.\,\psi$ hergeleitet wird, dann gilt per Induktionsannahme $\Phi' \models \forall X.\,\psi$. Sei nun $\mathfrak{M}, \eta \models \Phi'$; zu zeigen ist $\mathfrak{M}, \eta \models \psi[X \mapsto E]$. Dies ist per Substitutionslemma äquivalent zu $\mathfrak{M}, \eta[X \mapsto \mathfrak{M}[\![E]\!]\eta] \models \psi$, was aber unmittelbar nach Definition der Semantik aus $\mathfrak{M}, \eta \models \forall X.\,\psi$ folgt. □

6 Unifikation

Unifikation bezeichnet dass Problem, wie – und ob überhaupt – zwei Terme strukturell gleich gemacht, sprich *unifiziert* werden können. Dieser Begriff und die entsprechenden Verfahren spielen eine zentrale Rolle in de automatischen Deduktion und in der logischen Programmierung.

Definition 52 (Gleichung, Unifikatoren, Allgemeinheitsvergleich)**.** Eine *Gleichung* $E \doteq D$ ist ein Paar (E, D) von Termen. Ein *Gleichungssystem* ist eine Menge von Gleichungen. Eine Substitution σ ist ein *Unifikator* von $E \doteq D$, wenn $E\sigma = D\sigma$, wobei wir (wie auch bisher schon) mit dem undekorierten Gleichheitszeichen '=' *syntaktische* Gleichheit bezeichnen, d.h. mit der voranstehenden Gleichung meinen wir, dass $E\sigma$ und $D\sigma$ wörtlich identische Terme sind. Ein *Unifikator* eines Gleichungssystem S ist eine Substitution, die Unifikator aller Gleichungen in S ist. Das System S ist *unifizierbar*, wenn es einen Unifikator hat. Wir bezeichnen mit

$$\mathsf{Unif}(S) = \{\sigma \mid \sigma \text{ ist Unifikator von } S\}$$

die Menge aller Unifikatoren von S. Eine Substitution σ_1 ist *allgemeiner als* σ_2, und σ_2 heißt dann umgekehrt eine *Spezialisierung von* σ_1, wenn eine Substitution τ existiert mit $\sigma_1\tau = \sigma_2$, also wenn σ_2 durch Einsetzen aus σ_1 hervorgeht.

Ein Unifikator σ von S ist *allgemeinster Unifikator* (MGU, für *most general unifier*) *von* S ($\sigma = mgu(S)$), wenn σ allgemeiner als jeder Unifikator von S ist.

Beispiel 53. Zur Gleichung

$$add(suc(x), y) \doteq add(y, suc(z))$$

haben wir z.B. den Unifikator $\sigma = [suc(x)/y, x/z]$. Man kann sich überzeugen, dass dies sogar ein mgu ist. In jedem Fall ist z.B. $\sigma' = [suc(suc(z))/y, suc(z)/z, suc(z)/x]$ ein weiterer Unifikator, und σ ist allgemeiner als σ': wir haben

$$\sigma' = \sigma[suc(z)/x].$$

Unifikatoren sind abgeschlossen unter Spezialisierung, d.h. eine Spezialisierung eines Unifikators ist selbst wieder ein Unifikator:

$$\sigma \in \mathsf{Unif}(S) \Rightarrow \sigma\tau \in \mathsf{Unif}(S).$$

Dies können wir nutzen, um alle Unifikatoren eines Gleichungssystems zu finden, wenn wir einen *allgemeinsten Unifikator* gefunden haben:

Lemma 54. *Wenn* $\sigma = mgu(S)$*, dann gilt* $\mathsf{Unif}(S) = \{\sigma\tau \mid \tau \text{ Subst.}\}$.

Kennt man also den allgemeinsten Unifikator von S, so erhält man durch Spezialisierung gerade alle Unifikatoren von S.

Setze nun für Substitutionen σ

$$FV(\sigma) = \bigcup_{x \in V} FV(\sigma(x)).$$

Lemma 55 (Eindeutigkeit des MGU)**.** *Der MGU ist eindeutig bis auf eindeutige und bijektive Umbenennung von Variablen, d.h. wenn* σ, σ' *allgemeinste Unifikatoren von* S *sind, dann gilt:*

1. *Es existiert ein auf $FV(\sigma)$ eindeutig bestimmtes τ mit $\sigma' = \sigma\tau$;*

2. *dieses τ ist eine Umbenennung und bijektiv als Abb. $FV(\sigma) \to FV(\sigma')$.*

Beweis. τ existiert, da σ MGU und σ' Unifikator, ebenso existiert τ' mit $\sigma'\tau' = \sigma$.

τ *eindeutig auf* $x \in FV(\sigma(y))$: wenn auch $\sigma' = \sigma\overline{\tau}$, dann
$\sigma(y)\tau = \sigma(y)\overline{\tau}$, also notwendig $\tau(x) = \overline{\tau}(x)$ (formal: Induktion über $\sigma(y)$).

$\tau : FV(\sigma) \to FV(\sigma')$ *bijektiv:* Es gilt

$$\sigma\tau\tau' = \sigma,$$

also per Eindeutigkeit $\tau\tau'(x) = x$ für $x \in FV(\sigma)$, analog $\tau'\tau(x) = x$ für $x \in FV(\sigma')$. Es folgt, dass τ und τ' Umbenennungen sind, und dann gegenseitig invers als Abbildungen zwischen $FV(\sigma)$ und $FV(\sigma')$. □

Wir beweisen die Existenz eines MGU durch Angabe eines Algorithmus:

6.1 Unifikationsalgorithmus von Martelli/Montanari

Definition 56 (Gelöstes Gleichungssystem)**.** Ein Gleichungssystem S heißt *gelöst*, wenn jede Gleichung in S von der Form $x \doteq E$ ist, wobei x nur dieses eine Mal in S vorkommt. Zwei Gleichungssysteme S_1, S_2 sind *äquivalent*, wenn $\mathsf{Unif}(S_1) = \mathsf{Unif}(S_2)$. Ein zu S äquivalentes gelöstes Gleichungssystem heißt *gelöste Form* von S.

Lemma 57. *Wenn $S' = \{x_1 \doteq E_1, \ldots, x_n \doteq E_n\}$ gelöste Form von S ist, dann ist $\sigma = [E_1/x_1, \ldots, E_n/x_n$ MGU von S.*

Beweis. Da S' gelöst ist, ist σ Unifikator von S' und damit auch von S. Wenn σ' Unifikator von S ist, dann auch von S'; es gilt damit $\sigma'(x_i) = E_i\sigma' = \sigma(x_i)\sigma'$ für alle i. Wenn $y \notin \{x_1, \ldots, x_n\}$, dann gilt $\sigma(y)\sigma' = y\sigma' = \sigma'(y)$. Insgesamt gilt also $\sigma' = \sigma\sigma'$, d.h. σ ist allgemeiner als σ'. □

Der Algorithmus (ursprünglich Herbrand, später Martelli/Montanari) besteht nun in erschöpfender Anwendung der folgenden Umformungsregeln:

(delete):
$S \cup \{x \doteq x\} \quad \to S$

(decomp):
$S \cup \{f(E_1, \ldots, E_n) \doteq f(D_1, \ldots, D_n)\} \quad \to S \cup \{E_1 \doteq D_1, \ldots, E_n \doteq D_n\}$

(conflict) :
$S \cup \{f(E_1, \ldots, E_n) \doteq g(D_1, \ldots, D_k)\} \quad \to \bot \qquad (\text{für } f \neq g)$

(orient) :
$S \cup \{E \doteq x\} \quad \to S \cup \{x \doteq E\} \qquad (E \text{ keine Variable})$

(occurs)/(elim) :

$$S \cup \{x \doteq E\} \quad \to \begin{cases} \bot & (x \in FV(E), x \neq E) \\ S[E/x] \cup \{x \doteq E\} & (x \notin FV(E), x \in FV(S) \end{cases}$$

Wenn $\perp$ erreicht wird, gibt der Algorithmus „nicht unifizierbar“ aus; ansonsten berechnet er, wie wir noch zeigen, eine gelöste Form und damit einen MGU von S.

Beispiele:

- $f(x, g(y)) \doteq f(g(z), z)$
- $f(x, g(x), h(y)) \doteq f(k(y), g(z), z)$
- $f(x, g(x)) \doteq f(z, z)$

Wir zeigen zunächst Terminierung. Dazu erinnern wir an den Begriff des *Terminationsmaßes*: Wir ordnen jedem Zustand (hier: jedem Gleichungssystem S so wie aktuell umgeformt) ein Maß $\nu(S)$ zu, so dass $\nu(S)$ in jedem Schritt abnimmt. Wenn wir als Wertebereich des Maßes eine geordnete Menge wählen, in der es keine unendlichen absteigenden Ketten

$$x_0 > x_1 > x_2 > \dots$$

gibt, beweist das, dass immer nur endlich viele Zustände durchlaufen werden, bevor der Prozess stoppt. Ein Beispiel einer solchen Menge ist die Menge $\mathbb{N}^n$ aller n-Tupel von natürlichen Zahlen in *lexikographischer Ordnung* $\prec$. Diese definieren wir durch Rekursion über n wie folgt: Wir setzen

- $() \leq ()$
- für $n > 0$: $(a_1, \dots, a_n) \preceq (b_1, \dots, b_n)$ genau dann, wenn $a_1 \leq b_1$ und, falls $a_1 = b_1$, außerdem $(a_2, \dots, a_n) \preceq (b_2, \dots, b_n)$.

Es gilt nun in der Tat

Satz 58. *Die lexikographische Ordnung ist eine* Wohlordnung*, d.h. es gibt keine unendliche absteigende Kette*

$$\vec{a}^0 \succ \vec{a}^1 \succ \vec{a}^2 \succ \dots \tag{4}$$

Beweis. Induktion über n, mit trivialem Induktionsanfang. Wir nehmen zwecks Herleitung eines Widerspruchs an, wir hätten eine unendliche absteigende Kette (4), mit $\vec{a}^i = (a_1, \dots, a_n)$. Dann gilt

$$a_1^0 \geq a_1^1 \geq a_1^2 \geq \dots.$$

Diese Kette wird, da es in $\mathbb{N}$ keine unendlichen absteigenden Ketten gibt, irgendwann *stationär*, d.h. es existiert k mit $a_1^i = a_1^k$ für alle $i \geq k$. Dann gilt aber per Definition

$$(a_2^k, \dots, a_n^k) \succ (a_2^{k+1}, \dots, a_n^{k+1}) \succ (a_2^{k+2}, \dots, a_n^{k+2}) \succ \dots,$$

im Widerspruch zur Induktionsvoraussetzung, die besagt, dass es in $\mathbb{N}^{n-1}$ keine unendlichen absteigenden Ketten gibt. □

Im vorliegenden Fall verwenden wir als Terminationsmaß das Tripel $(n_0 - n_1, n_2, n_3)$ in lexikographischer Ordnung, wobei

- n_0 = Anzahl der *ursprünglich* im System vorkommenden Variablen

- n_1 = Anzahl der Variablen x, die *erledigt* sind, d.h. als linke Seite einer Gleichung vorkommen und sonst nirgends.
- n_2 = Anzahl Symbole
- n_3 = Anzahl Gleichungen der Form $E = x$ mit E keine Variable.

Diese Maß nimmt offenbar in jedem Umformungsschritt ab, so dass der Algorithmus terminiert. Die erste Komponente $n_0 - n_1$ ist nichtnegativ, da der Algorithmus nie neue Variablen hinzufügt, also alle erledigten Variablen von vornherein im System vorgekommen sind.

Wenn der Algorithmus ohne Erreichen von $\bot$ terminiert, also keine Regel mehr anwendbar ist, dann ist das so erreichte System S gelöst: wenn $E \doteq D$ eine Gleichung in S ist, dann kann E nicht zusammengesetzt sein (sonst greift eine der Regeln *(decomp)*, *(conflict)* oder *(orient)*). Also ist E eine Variable x. Wegen der Regeln *(delete)*, *(occurs)* und *elim)* kommt dann x weder in D noch in den restlichen Gleichungen vor.

Es bleibt zu zeigen, dass die Regeln das gegebene Gleichungssystem stets in ein äquivalentes transformieren, wobei $\bot$ das unlösbare System repräsentiert. Das ist klar in allen Fällen bis vielleicht auf *(occurs)*. Eine Gleichung der Form $x \doteq E$ mit $x \in FV(E)$ ist aber offenbar nicht unifizierbar: der Term $E\sigma$ ist stets größer als der Term $x\sigma$.

7 Normalformen in Logik erster Stufe

7.1 Pränexe Normalform

Definition 59. Eine *pränexe Normalform* ist eine Formel der Form

$$Q_1 X_1 . \ldots Q_n X_n . \phi$$

mit $Q_1, \ldots, Q_n \in \{\forall, \exists\}$ und ϕ quantorenfrei.

Satz 60. *Zu jeder Formel in Prädikatenlogik erster Stufe lässt sich eine äquivalente pränexe Normalform berechnen.*

Beweis. Man eliminiert zunächst Implikationen und Biimplikationen. Die Berechnung erfolgt dann durch erschöpfende Anwendung der Umformungsregeln

$$\begin{array}{ccc} \neg\exists X.\,\phi \equiv \forall X.\,\neg\phi & & \neg\forall X.\,\phi \equiv \exists X.\,\neg\phi \\ \phi \wedge \exists X.\,\psi \equiv \exists X.\,(\phi \wedge \psi) & & \phi \wedge \forall X.\,\psi \equiv \forall X.\,(\phi \wedge \psi), \phi \vee \exists X.\,\psi \quad \equiv \exists X.\,(\phi \vee \psi) \\ \phi \vee \forall X.\,\psi & \equiv \forall X.\,(\phi \vee \psi), & \end{array}$$

wobei in der zweiten Zeile $X \notin FV(\phi)$ vorausgesetzt wird, was durch geeignete Umbenennung der gebundenen Variablen X stets erreicht werden kann. Es sind bei den Fällen für $\vee$ und $\wedge$ jeweils die symmetrischen Fälle mit gemeint. Jede Umformung beseitigt einen *Fehlstand*, d.h. ein Vorkommen eines Booleschen Konnektivs oberhalb eines Quantors, so dass das Verfahren mit einer Formel, auf die keine Umformungsregel mehr anwendbar ist, terminiert; eine solche ist eine pränexe Normalform. □

Beispiel 61. Die Formel

$$\forall X.\,((\forall Y.\,L(Y,X)) \to \exists Y.\,M(X,Y))$$

wird wie folgt umgeformt:

$$\begin{aligned}
&\forall X.\,((\forall Y.\,L(Y,X)) \to \exists Y.\,M(X,Y))\\
&\equiv \forall X.\,(\neg(\forall Y.\,L(Y,X)) \vee \exists Y.\,M(X,Y))\\
&\equiv \forall X.\,(\exists Y.\,((\exists Y.\,\neg L(Y,X)) \vee\ M(X,Y)))\\
&\equiv \forall X.\,(\exists Y.\,((\exists Y'.\,(\neg L(Y',X) \vee\ M(X,Y)))))
\end{aligned}$$

7.2 Skolemform

Definition 62. Eine *Skolemform* ist eine pränexe Normalform, die nur Allquantoren enthält.

Es gibt ersichtlich nicht zu jeder Formel eine äquivalente Skolemform; z.B. ist jede Skolemform über der leeren Signatur Σ äquivalent zu entweder $\top$ oder $\bot$ oder $\forall X, Y.\,X = Y$ (warum?), und damit nicht äquivalent zu $\exists X.\,\exists Y.\,X \neq Y$. Es gilt jedoch

Definition 63. Formeln ϕ, ψ heißen *erfüllbarkeitsäquivalent*, wenn ϕ erfüllbar ist genau dann, wenn ψ erfüllbar ist.

Satz 64. *Zu jeder Formel in Prädikatenlogik erster Stufe lässt sich eine erfüllbarkeitsäquivalente Skolemform berechnen.*

Diese *Skolemisierung* beruht auf der Ersetzung von Existenzquantoren durch frische Konstanten, basierend auf der Beobachtung, dass $\exists X.\,\phi$ genau dann erfüllbar ist, wenn $\phi[c/X]$ für eine frische Konstante c (eine *Skolemkonstante*) erfüllbar ist. Im allgemeinen hängen die für X einzusetzenden Werte noch von vorhergehenden allquantifizierten Variablen ab, so dass man es mit Skolemfunktionen statt nur Skolemkonstanten zu tun hat. Formal:

Beweis. Man berechnet zunächst eine pränexe Normalform

$$\phi = Q_1 X_1 . \ldots Q_n X_n .\,\phi_0,$$

streicht dann alle Existenzquantoren $\exists X_j$ und substituiert jede existenzquantifizierte Variable X_j durch

$$f_j(X_{j_1}, \ldots, X_{j_k}),$$

wobei f_j ein jeweils neu eingeführtes k-stelliges Funktionssymbol ist und $j_1, \ldots, j_k$ diejenigen Indizes $\leq j$ sind, für die $Q_j = \forall$. Die so erhaltene Skolemform $\bar{\phi}$ ist erfüllbarkeitsäquivalent zu ϕ: wenn $\bar{\phi}$ in einem Modell $\mathfrak{M}$ erfüllt ist, dann auch ϕ, da jeweils die Interpretation von $f_j(X_{j_1}, \ldots, X_{j_k})$ einen Zeugen für den Existenzquantor für X_j liefert. Wenn umgekehrt ϕ in einem Modell $\mathfrak{M}$ erfüllt ist, dann kann man ein Modell $\bar{\mathfrak{M}}$ konstruieren, das $\bar{\phi}$ erfüllt, indem man die neuen Symbole f_j so interpretiert, dass sie jeweils Zeugen für die Existenz von X_j mit der verlangten Eigenschaft auswählen. (Das hängt allerdings am sogenannten *Auswahlaxiom* der Mengentheorie.) □

Mit anderen Worten transformieren wir eine pränexe Normalform in Skolemform durch wiederholte Anwendung der Umformung

$$\forall X_1. \ldots \forall X_n. \exists Y. \phi \rightsquigarrow \forall X_1. \ldots \forall X_n. \phi[f(X_1, \ldots, X_n)/Y]$$

(Achtung: Nur auf ganze Formeln, nicht auf Teilformeln innerhalb größerer Formeln!) mit jeweils einem frischen Funktionssymbol f. Bei der Transformation von Formelmengen darf jedes f nur zur Transformation einer Formel verwendet werden.

Beispiel 65. Wir führen die Normalisierung aus Beispiel 61 fort:

$$\begin{aligned} &\forall X. (\exists Y. ((\exists Y'. (\neg L(Y', X) \vee\ M(X, Y))))) \\ \equiv\ &\forall X. (\neg L(f_3(X), X) \vee\ M(X, f_2(X))). \end{aligned}$$

Bemerkung 66. Das im Beweis von Satz 60 verwendete Verfahren zur Berechnung einer pränexen Normalform beinhaltet Wahlfreiheiten hinsichtlich der jeweils als nächstes anzuwendenden Umformung, und verschiedene Umformungsabfolgen können verschiedene pränexe Normalformen liefern. Z.B. kann man $(\forall x. P(x)) \vee \exists y. Q(y)$ zu sowohl $\forall x. \exists y. P(x) \vee Q(y)$ also auch $\exists y. \forall x. P(x) \vee Q(y)$ umformen. Zweiteres ist für Zwecke der Skolemisierung günstiger, da dann die Skolemfunktion für den Existenzquantor ein Argument weniger (nämlich keins) hat. Aus ähnlichen Gründen kann es vorteilhaft sein, vor der Bildung von pränexen Normalformen die Reihenfolge in Konjunktionen oder Disjunktionen geeignet zu vertauschen.

7.3 Klauselform

Man kann offenbar jede pränexe Normalform $Q_1 X_1 \ldots Q_n X_n. \phi$ so weiter normalisieren, dass ϕ in CNF ist. Wenn die Formel zusätzlich in Skolemform ist, also $Q_i = \forall$ für alle i, dann kann man die führenden Quantoren in der Notation weglassen, so dass man also alle Variablen als implizit allquantifiziert ansieht. Für die so erhaltene CNF verwendet man dann üblicherweise die bereits eingeführte Schreibweise als Menge von Klauseln. Eine solche Klauselmenge heißt *Klauselform* bzw. *ist in Klauselform*.

Beispiel 67. Die Formel aus Beispiel 65 hat bereits eine CNF als quantorenfreien Anteil; sie lautet in Klauselform (unter Weglassung der äußeren Mengenklammern)

$$\{\neg L(f_3(X), X),\ M(X, f_2(X))\}$$

(besteht also nur aus einer Klausel).

8 Resolution in Prädikatenlogik erster Stufe

Das Resolutionsverfahren für Aussagenlogik lässt sich auf Prädikatenlogik erster Stufe erweitern, ist dort allerdings nur noch ein Halbentscheidungsverfahren für Unerfüllbarkeit (d.h. für erfüllbare Eingabeformeln terminiert es möglicherweise nicht). Das Verfahren arbeitet mit Formeln in Klauselform.

Die sogenannte *generalisierte Reduktionsregel*, auf der das Verfahren beruht, kombiniert die aussagenlogische Resolutionsregel

$$\frac{C_1, A \qquad C_2, \neg A}{C_1, C_2}$$

(wobei wir nunmehr Mengenklammern einsparen und Mengenvereinigungen einfach mit Kommata andeuten) mit der Spezialisierungsregel

$$\frac{C}{C\sigma},$$

die aus einer Klausel C eine Substitutionsinstanz für eine Substitution σ folgert, sowie mit der *Faktorisierungsregel*

$$\frac{C, A, A}{C, A}.$$

Die kombinierte Regel, auch *Resolution mit impliziter Faktorisierung* genannt, verwendet Unifikation. Sie lautet

$$(RIF)\frac{C_1, A_1, \ldots, A_n \qquad C_2, \neg B}{C_1\sigma, C_2\sigma} \qquad (\sigma = mgu(A_1, \ldots, A_n, B)).$$

Hierbei schreiben wir kurz $mgu(A_1, \ldots, A_n, B)$ für den mgu des Systems $\{A_i \doteq B \mid i = 1, \ldots, n\}$. Vor Anwendung der Regel werden die Variablen in den beiden ursprünglichen Klauseln so umbenannt, dass die Klauseln disjunkte Variablenmengen haben (aus Lesbarkeitsgründen nehmen wir diese Umbenennung nicht in die Notation der Regel mit auf). Das ist offenbar zulässig, da die Klauseln ja allquantifizierte Formeln darstellen, und vergrößert die Menge der Unifikatoren.

Der Algorithmus zum Beweis der Gültigkeit einer Formel ϕ lautet damit

1. Bilde $\neg\phi$
2. Transformiere $\neg\phi$ in Klauselform
3. Wende (RIF) an, bis $\square$ erreicht ist.

(Achtung: anders als bei der aussagenlogischen Resolution kann man hier i.a. nicht erwarten, dass, wenn ϕ nicht gültig ist, irgendwann eine Formelmenge erreicht wird, auf die (RIF) nicht mehr anwendbar ist; d.h. der Algorithmus terminiert in diesem Fall eventuell nicht.)

Beispiel 68. Die Negation der Formel

$$P(a) \wedge (\forall X.\, (P(X) \to P(f(X)))) \to \exists X.\, P(f(f(X)))$$

wird in Klauselform zu

$$(1)\ \{P(a)\} \qquad (2)\ \{\neg P(X), P(f(X))\} \qquad (3)\ \{\neg P(f(f(X)))\}.$$

Resolution von (1) mit (2) liefert

$$(4)\ \{P(f(a))\},$$

Resolution von (4) mit (2) liefert

$$(5)\ \{P(f(f(a)))\},$$

und Resolution mit (3) liefert schließlich $\square$, womit die ursprüngliche Formel bewiesen ist.

8.1 Herbrand-Modelle

Wir zeigen nun die *Vollständigkeit* des Resolutionsalgorithmus, d.h. dass der Algorithmus auf jeder *unerfüllbaren* Klauselform mit der Antwort 'unerfüllbar' terminiert. (Auf *erfüllbaren* Klauselformen terminiert der Algorithmus eventuell nicht.) Der Schlüsselbegriff ist hierbei der des sogenannten *Herbrandmodells*; Herbrandmodelle zeichnen sich dadurch aus, dass in ihnen jedes Element durch einen Term ohne Variablen bezeichnet wird. Insbesondere sind Herbrandmodelle damit abzählbar; ihre Konstruktion impliziert daher den berühmten *Satz von Löwenheim-Skolem*, der besagt, dass jede erfüllbare Formelmenge in Prädikatenlogik ein höchstens abzählbares Modell hat.

Zur technischen Vereinfachung schränken wir ab jetzt auf Prädikatenlogik ohne „=“ ein; d.h. als atomare Formeln gibt es nur noch Prädikatenanwendungen $P(E_1, \dots, E_n)$. Für den Rest des Abschnitts sei eine Signatur Σ gegeben; o.E. nehmen wir an, dass Σ mindestens eine Konstante enthält (dass das o.E. ist, liegt daran, dass per Definition alle Modelle nichtleeren Träger haben).

Definition 69 (Herbrand-Universum)**.** Das *Herbrand-Universum* U_Σ ist die Menge aller *Grundterme* (Ground Terms), d.h. Terme ohne Variablen: $U_\Sigma = \{E \mid E\ \Sigma\text{-Term}, FV(E) = \emptyset\}$. Für $E \in U_\Sigma$ und ein Σ-Modell $\mathfrak{M}$ schreiben wir $\mathfrak{M}[\![E]\!]$ statt $\mathfrak{M}[\![E]\!]\eta$ für die Auswertung von E in $\mathfrak{M}$, da Umgebungen für die Auswertung geschlossener Terme irrelevant sind.

Beispiel 70. Bei $\Sigma = \{s, 0, \mathrm{odd}\}$ haben wir $U_\Sigma = \{0, s(0), s(s(0)), \dots\}$

Definition 71 (Herbrand-Modell)**.** Ein *Herbrand-(Σ-)Modell* ist ein Σ-Modell $\mathfrak{M}$ mit

- $M = U_\Sigma$
- Für $f/n \in \Sigma$ und $E_1, \dots, E_n \in U_\Sigma$ gilt

$$\mathfrak{M}[\![f]\!](E_1, \dots, E_n) = f(E_1, \dots, E_n)$$

Lemma 72 (Auswertungslemma für Terme)**.** *Sei $\mathfrak{M}$ ein Herbrand-Σ-Modell. Dann gilt*

$$\mathfrak{M}[\![E]\!]\eta = E\eta$$

für jeden Term E. Insbesondere gilt $\mathfrak{M}[\![E]\!] = E$ für Grundterme E.

Zum Verständnis der Aussage des Lemmas beachte man, dass eine Umgebung η in $\mathfrak{M}$ gleichzeitig eine Substitution ist, eben eine Abbildung von Variablen auf Terme (bei Unterschlagung der hier nicht so wichtigen Endlichkeitsbedingung an $\mathrm{Dom}(\eta)$).

Beweis. Induktion über E. □

Definition 73 (Grundinstanz)**.** Sei φ eine Formel. Eine *Grundinstanz* (Ground Instance) von φ ist eine Formel der Form $\varphi\sigma$, wobei σ eine *Grundsubstitution* (Ground Substitution) ist, d.h. für alle $x \in FV(\varphi)$ ist $\sigma(x)$ ein Grundterm. Damit ist insbesondere $\varphi\sigma$ eine geschlossene Formel, also $FV(\varphi\sigma) = \emptyset$. Für eine Menge Φ von Formeln bezeichnet $E(\Phi)$ die Menge der Grundinstanzen von Formeln aus Φ.

Lemma 74 (Auswertungslemma für Formeln)**.** *Sei φ eine Formel und $\mathfrak{M}$ ein Herbrand-Modell. Dann gilt, unter erneuter Verwechslung von Substitutionen und Umgebungen,*

$$\mathfrak{M}, \eta \models \varphi \iff \mathfrak{M} \models \varphi\eta.$$

Man beachte dabei, dass oben $\varphi\eta$ eine Grundinstanz von φ ist.

Beweis. Nach dem Substitutionslemma gilt zunächst

$$\mathfrak{M}, \eta' \models \varphi \iff \mathfrak{M} \models \varphi\eta,$$

wobei $\eta'(X) = \mathfrak{M}[\![\eta(X)]\!]$. Nach dem Auswertungslemma für Terme gilt aber $\eta' = \eta$. □

Definition 75 (Universeller Abschluss)**.** Der *universelle Abschluss* $\forall\varphi$ einer Formel φ mit $FV(\varphi) = \{x_1, \ldots, x_n\}$ ist die Formel $\forall x_1 \ldots \forall x_n . \varphi$. Wir dehnen diesen Begriff auf Formelmengen Φ aus per $\forall\Phi = \{\forall\varphi \mid \varphi \in \Phi\}$.

In Worten: Der universelle Abschluss von φ allquantifiziert alle freien Variablen in φ.

Lemma 76 (Grundinstanzlemma)**.** *Sei $\mathfrak{M}$ ein Herbrand-Σ-Modell und φ eine Formel. Dann gilt*

$$\mathfrak{M} \vDash \forall\varphi \iff \mathfrak{M} \vDash \varphi\sigma \text{ für jede Grundinstanz } \varphi\sigma \text{ von } \varphi.$$

Beweis. Klar nach dem Auswertungslemma. □

Entscheidend ist nunmehr

Satz 77 (Herbrand-Vollständigkeit)**.** *Sei Φ eine Menge von quantorenfreien Formeln über Σ. Dann sind äquivalent:*

1. *$\forall\Phi$ ist erfüllbar.*
2. *Es gibt ein ein Herbrandmodell $\mathfrak{M}$ mit $\mathfrak{M} \models \forall\Phi$.*
3. *$E(\Phi)$ ist aussagenlogisch erfüllbar.*

Dabei meinen wir in 3 mit *aussagenlogisch erfüllbar*, dass die Menge aussagenlogischer Formeln, die man erhält, wenn man alle atomaren Unterformeln und alle quantifizierten Unterformeln in $E(\Phi)$ als Atome mit merkwürdigen Namen ansieht, erfüllbar ist. Z.B. ist $(\forall x.\phi) \wedge \exists x.\neg\phi$ aussagenlogisch erfüllbar, da es strukturgleich zur aussagenlogischen Formel $A \wedge B$ ist; dagegen ist $(\forall x.\phi) \wedge \neg\forall x.\phi$ aussagenlogisch unerfüllbar, da es strukturgleich zur aussagenlogischen Formel $A \wedge \neg A$ ist.

Beweis. 1. $\implies$ 3. und 2. $\implies$ 1. sind trivial. Wir zeigen 3. $\implies$ 2. Sei $\kappa \models E(\phi)$. Wir konstruieren dann ein Herbrand-Modell $\mathfrak{M}$ mit $\mathfrak{M} \models \forall\Phi$ per

$$\mathfrak{M}[\![P]\!] = \{(E_1, \ldots, E_n) \mid \kappa(P(E_1, \ldots, E_n)) = \top\}$$

für $P/n \in \Sigma$. Um zu zeigen, dass tatsächlich $\mathfrak{M} \models \forall\Phi$ gilt, reicht es nach dem Grundinstanzlemma zu zeigen, dass $\mathfrak{M} \models E(\phi)$. Das zeigt man durch Induktion über quantorenfreies ϕ, dass für jede Grundinstanz $\phi\sigma$

$$\mathfrak{M} \models \phi\sigma \iff \kappa \models \phi\sigma \tag{5}$$

gilt. Damit folgt dann die Behauptung, da ja $\kappa \models E(\phi)$. Die Booleschen Fälle in der Induktion für (5) sind (wie meistens) leicht, und für atomare Formeln gilt die Äquivalenz nach der Konstruktion von $\mathfrak{M}$ und dem Auswertungslemma. □

Der Satz gilt nicht für beliebige Formelmengen, z.B.

Beispiel 78.

$$\begin{aligned}\Sigma &= \{a/1, P/1\}\\ \Phi &= \{\neg P(a), \exists x.P(x)\}\\ U_\Sigma &= \{a\}\end{aligned}$$

Φ ist offenbar in keinem einelementigen Modell erfüllbar, insbesondere also in keinem Herbrand-Modell, da hier $U_\Sigma = \{a\}$. Φ ist andererseits aber offenbar erfüllbar, z.B. wähle $M = \{0, 1\}$, $\mathfrak{M}[\![a]\!] = 0$, $\mathfrak{M}[\![P]\!] = \{1\}$.

Da Herbrandmodelle abzählbar sind, folgt hieraus wie angekündigt

Satz 79 (Löwenheim/Skolem)**.** *Jede erfüllbare Formelmenge hat ein höchstens abzählbares Modell.*

Beweis. Per Skolemisierung ist jede Formelmenge erfüllbarkeitsäquivalent zu einer der Form $\forall\Phi$ mit Φ quantorenfrei; die Behauptung folgt per Herbrand-Vollständigkeit. □

Satz 80. *Logik erster Stufe ist* kompakt*, d.h. wenn Φ eine Menge von Sätzen ist, so dass jede endliche Teilmenge von Φ erfüllbar ist, dann ist Φ erfüllbar.*

Beweis. Wir können annehmen, dass Φ aus Skolemformen besteht, d.h. $\Phi = \forall\Phi_0$ für eine Menge Φ_0 von quantorenfreien Formeln. Dann ist Φ erfüllbar genau dann, wenn $E(\Phi_0)$ aussagenlogisch erfüllbar ist. Jede endliche Teilmenge Ψ von $E(\Phi_0)$ ist enthalten in einer Menge der Form $E(\bar{\Psi})$ für eine endliche Teilmenge $\bar{\Psi}$ von Φ_0. Nach Voraussetzung ist $\bar{\Psi}$ erfüllbar, damit auch $E(\bar{\Psi})$ und somit erst recht auch Ψ, d.h. $E(\Phi_0)$ ist endlich erfüllbar, also nach Kompaktheit der Aussagenlogik auch $E(\Phi_0)$. □

Satz 81. *Es gibt einen Algorithmus, der alle gültigen Formeln in Logik erster Stufe aufzählt.*

Beweis. Dual zeigen wir, dass es einen Algorithmus gibt, der die *unerfüllbaren* Formeln auflistet. Wir können uns auf Skolemformen φ einschränken; es reicht, einen Algorithmus anzugeben, der mit der Antwort „unerfüllbar“ terminiert, wenn φ unerfüllbar ist, und ansonsten entweder „erfüllbar“ antwortet oder nicht terminiert. Wir erzeugen einfach nacheinander alle Grundinstanzen von φ; sobald wir eine aussagenlogisch unerfüllbare Menge erreichen, antworten wir „unerfüllbar“; falls die Menge der Grundinstanzen unendlich und aussagenlogisch erfüllbar ist, terminiert der Algorithmus also nicht. Im Sonderfall, dass die Menge der Grundinstanzen endlich und aussagenlogisch erfüllbar ist, antworten wir „erfüllbar“. □

8.2 Vollständigkeit der Resolution

Wir zeigen nunmehr wie angekündigt die Vollständigkeit der prädikatenlogischen Resolution. Wir reduzieren diese mittels Herbrandvollständigkeit auf die schon gezeigte Vollständigkeit der aussagenlogischen Resolution. Hierzu benötigen wir eine technische Aussage über das Verhältnis der beiden Resolutionsverfahren, für die wir wiederum die genaue Gestaltung der prädikatenlogischen Resolutionsregel eingehender diskutieren müssen.

Wir erinnern daran, dass die Anwendung von (RIF) auf den mgu als Substitution beschränkt ist. Wir bezeichnen mit (lRIF) die *liberalisierte RIF-Regel*, in der wir diese Einschränkung aufgeben

und stattdessen beliebige Unifikatoren von $A_1, \ldots, A_n, B$ zulassen. Dies ändert offenbar nicht die Stärke des Systems:

Lemma 82. *Wenn sich aus einer Klauselmenge mittels (lRIF) die leere Klausel herleiten lässt, dann auch mittels (RIF).*

Beweis. Man zeigt durch Induktion über die Länge der Herleitung, dass jede mittels (lRIF) herleitbare Klausel eine Substitutionsinstanz einer mittels (RIF) herleitbaren Klausel ist. Damit folgt die Behauptung, da eine Klausel, die die leere Klausel als Substitutionsinstanz hat, selbst leer sein muss. □

Wir bezeichnen ferner mit (RI2F) (*resolution with implicit two-sided factoring*) die Regel

$$\frac{C, A_1, \ldots, A_n \qquad \neg B_1, \ldots, \neg B_k, D}{C\sigma, D\sigma} \quad \sigma = mgu(A_1, \ldots, A_n, B_1, \ldots, B_k)$$

Lemma 83. *(RI2F) ist aus (lRIF) herleitbar.*

Beweis. Ohne Einschränkung bestehe $FV(\sigma)$ aus frischen Variablen, so dass $\sigma\sigma = \sigma$. Wir wenden (lRIF) mit der Substitution σ auf $A_1, \ldots, A_n, \neg B_1$ an und erhalten

$$C\sigma, \neg B_2\sigma, \ldots, \neg B_k\sigma, D\sigma.$$

Wir wenden dann (lRIF) mit der Substitution σ an auf $A_1, \ldots, A_n, B_2\sigma$ und erhalten (da $\sigma\sigma = \sigma$)

$$C\sigma, \neg B_3\sigma, \ldots, \neg B_k\sigma, D\sigma.$$

Wir fahren so fort und erhalten schließlich $C\sigma, D\sigma$. □

Lemma 84 (Lifting-Lemma)**.** *Seien $C, A_1, \ldots, A_n$ und $\neg B_1, \ldots, \neg B_k, D$ prädikatenlogische Klauseln, und sei σ eine Grundsubstitution mit $A_i\sigma = B_j\sigma$ für alle i, j. Dann ist die zugehörige aussagenlogische Resolvente $C\sigma, D\sigma$ Grundinstanz einer prädikatenlogischen Resolvente von $C, A_1, \ldots, A_n$ und $\neg B_1, \ldots, \neg B_k, D$ (nach (RI2F)).*

Beweis. Wähle $\sigma' = mgu(A_1, \ldots, A_n, B_1, \ldots, B_k)$. Dann gilt $\sigma = \sigma'\tau$ für eine Grundsubstitution τ, und $C\sigma, D\sigma$ ist unter τ Grundinstanz der prädikatenlogischen Resolvente $C\sigma', D\sigma'$. □

Damit gilt

Satz 85 (Vollständigkeit der prädikatenlogischen Resolution)**.** *Wenn Φ eine unerfüllbare Menge von prädikatenlogischen Klauseln ist, dann existiert eine Herleitung der leeren Klausel $\square$ aus Φ mittels prädikatenlogischer Resolution.*

Beweis. Nach Herbrandvollständigkeit ist $E(\Phi)$ aussagenlogisch unerfüllbar; nach der Vollständigkeit der aussagenlogischen Resolution existiert also eine Herleitung von $\square$ aus Grundinstanzen mittels aussagenlogischer Resolution. Indem wir das Lifting-Lemma auf alle Schritte dieses Resolutionsbeweises anwenden, erhalten wir eine Herleitung einer Klausel C aus Φ mittels (RI2F), und damit auch per (RIF), so dass $\square$ eine Grundinstanz von C ist. Dann ist aber $C = \square$. □

9 Quantorenelimination

Erfüllbarkeit von Formeln in Prädikatenlogik ist im allgemeinen unentscheidbar; dies zeigt man z.B. durch Reduktion des Postschen Korrespondenzproblems. Erfüllbarkeit in bestimmten Theorien kann dagegen in günstigen Fällen entscheidbar sein. Oft basieren Entscheidungsverfahren auf der Methode der *Quantorenelimnation*, d.h. der äquivalenten Ersetzung von Formeln durch quantorenfreie Formeln.

Definition 86 (Theorie)**.** Eine *Theorie* $\mathcal{T} = (\Sigma, \Phi)$ besteht aus einer Signatur Σ und einer Menge Φ von Σ-Sätzen, ihren *Axiomen*. Ein *Modell* von $\mathcal{T}$ ist ein Σ-Modell $\mathfrak{M}$ mit $\mathfrak{M} \models \Phi$. Eine Σ-Formel ψ heißt *erfüllbar über* $\mathcal{T}$, wenn ψ in einem Modell von $\mathcal{T}$ erfüllbar ist, d.h. wenn $\Phi \cup \{\psi\}$ erfüllbar ist. Wir schreiben $\mathcal{T} \models \psi$, wenn $\neg\psi$ unerfüllbar über $\mathcal{T}$ ist, d.h. wenn $\Phi \models \psi$.

Beispiel 87. Die Theorie $\mathcal{T}_<$ der *dichten linearen Ordnungen ohne Endpunkte* hat die Signatur

$$\Sigma_< = \{< /2\},$$

d.h. ein binäres Prädikat $<$, geschrieben in Infixnotation, sowie die Axiome

$$\begin{array}{ll}
\forall X.\, \neg(X < X) & \text{(Irreflexivität)} \\
\forall X, Y, Z.\, (X < Y \wedge Y < Z \to X < Z) & \text{(Transitivität)} \\
\forall X, Y.\, (X < Y \vee X = Y \vee Y < X) & \text{(Trichotomie)} \\
\forall X, Y.\, (X < Y \to \exists Z.\, (X < Z \wedge Z < Y)) & \text{(Dichte)} \\
\forall X.\, \exists Y.\, Y < X & \\
\forall X.\, \exists Y.\, X < Y & \text{(Endpunktfreiheit).}
\end{array}$$

Ein Modell dieser Theorie sind z.B. die rationalen Zahlen mir der üblichen Interpretation von $<$; ein weiteres Modell sind die reellen Zahlen.

Wir werden mittels Quantorenelimination zeigen, dass Erfüllbarkeit über $\mathcal{T}_<$ entscheidbar ist; man sagt kurz, dass $\mathcal{T}_<$ *entscheidbar* ist. Der allgemeine Ansatz ist hierbei wie folgt.

Definition 88. Eine Formel heißt *quantorenfrei*, wenn Sie, nun ja, keine Quantoren enthält. Eine Theorie $\mathcal{T} = \{\Sigma, \Phi\}$ *hat Quantorenelimination*, wenn für jede Σ-Formel ϕ eine quantorenfreie Σ-Formel ϕ' berechenbar ist, so dass $\mathcal{T} \models \phi \leftrightarrow \phi'$.

Fakt 89. *Wenn $\mathcal{T} = (\Sigma, \Phi)$ Quantorenelimination hat und Erfüllbarkeit von quantorenfreien Formeln über $\mathcal{T}$ entscheidbar ist, dann ist $\mathcal{T}$ entscheidbar.*

Quantorenelimination lässt sich auf einen Spezialfall reduzieren:

Lemma 90. *Wenn für jede Σ-Formel ϕ der Form $\exists X.\, (\alpha_1 \wedge \cdots \wedge \alpha_n)$ mit Literalen α_i eine quantorenfreie Formel ϕ' mit $\mathcal{T} \models \phi \leftrightarrow \phi'$ berechenbar ist, dann hat $\mathcal{T}$ Quantorenelimination.*

Beweis. Wir zeigen per Induktion über ψ, dass für jede Σ-Formel ψ eine äquivalente quantorenfreie Formel berechenbar ist. Wir nehmen dabei an, dass ψ nur $\exists$ enthält (d.h. $\forall$ durch $\exists$ definiert ist). Für Atome ist nichts zu zeigen, und die Booleschen Fälle sind trivial. Sei also ψ von der Form $\exists X.\, \chi$. Nach Induktionsvoraussetzung existiert ein quantorenfreies χ' mit $\mathcal{T} \models \chi \leftrightarrow \chi'$. Wir bringen χ' in DNF $\chi' \equiv \chi'_1 \vee \cdots \vee \chi'_n$, wobei die χ'_i Konjunktionen von Literalen sind. Per Vertauschung von $\exists$ mit $\vee$ (s.o.) gilt dann $\mathcal{T} \models (\exists X.\, \chi) \leftrightarrow (\exists X.\, \chi'_1 \vee \cdots \vee \exists X.\, \chi'_n)$, und die Quantoren auf der rechten Seite sind nach Voraussetzung eliminierbar. □

Damit zeigen wir

Satz 91. *Die Theorie der dichten linearen Ordnungen ohne Endpunkte hat Quantorenelimination.*

Beweis. Sei $\phi = \exists X.\,(\alpha_1 \wedge \cdots \wedge \alpha_n)$ mit Literalen α_i. Wir gehen schrittweise vor:

1. Wir eliminieren zunächst Negation: Per Irreflexivität, Transitivität und Trichotomie gilt $\mathcal{T} \models (\neg Y = Z) \leftrightarrow (Y < Z \vee Z < Y)$ und $\mathcal{T} \models (\neg(Y < Z) \leftrightarrow (Y = Z \vee Z < Y)$.

2. Damit erreichen wir eine Formel, die keine Negationen mehr, dafür aber wieder Disjunktionen enthält. Wie im Beweis von Lemma 90 formen wir ϕ dann in eine Disjunktion von existenzquantifizierten Formeln um; es reicht offenbar aus, zu zeigen, dass wir aus einer solchen Formel den Quantor eliminieren können.

3. Damit erreichen wir eine Formel $\psi = \exists X.\,(\beta_1 \wedge \cdots \wedge \beta_n)$ mit *Atomen* β_i. Wir eliminieren nun $=$:

   - Wenn $\beta_i = (Y = Y)$, dann streichen wir β_i.
   - Wenn $\beta_i = (Y = X)$ oder $\beta_i = (X = Y)$, dann streichen wir β_i und ersetzen in die verbleibenden β_j jeweils durch $\beta_j[X \mapsto Y]$. Abschließend streichen wir den Existenzquantor (X kommt ja nun unter ihm nicht mehr vor) und sind fertig.

   Falls wir noch nicht fertig sind, ziehen wir von den verbleibenden Atomen nunmehr diejenigen, die X nicht erwähnen, vor den Existenzquantor (s.o.), unter dem dann kein $=$ mehr vorkommt.

4. Falls unter den verbleibenden β_j $X < X$ vorkommnt, ersetzen wir ψ durch $\bot$ (per Irreflexivität).

5. Der verbleibende existenzquantifizierte Anteil hat nunmehr (nach einfacher Umformung) die Form
$$\exists X.\,(\bigwedge_{i=1}^{n}(U_i < X) \wedge \bigwedge_{j=1}^{m}(X < V_j), \tag{6}$$
wobei die U_i und die V_j von X verschiedene Variablen sind. Diese Formel ist in $\mathcal{T}_<$ äquivalent zu
$$\bigwedge_{i=1}^{n}\bigwedge_{j=1}^{m} U_i < V_j. \tag{7}$$
Dabei ist die Implikation (6) $\Longrightarrow$ (7) klar per Transitivität. Für die Umkehrung bemerken wir, dass es per Trichotomie unter den U_i ein größtes geben muss; sei dies U_{i_0} (formal beweist man per Induktion eine Disjunktion mit n Disjunkten, eins für jedes in Frage kommende U_i). Entsprechend gibt es unter den V_j ein kleinstes, V_{j_0}. Dann gilt nach Voraussetzung $U_{i_0} < V_{j_0}$, so dass per Dichte X existiert mit $U_{i_0} < X$ und $X < V_{j_0}$; dieses X ist dann per Transitivität ein Zeuge für (6). Diese Argumentation setzt natürlich voraus, dass $n, m > 0$. Für $n > 0, m = 0$ und für $n = 0, m > 0$ führt man ein ähnliches Argument mittels Endpunktfreiheit durch; der Fall $n = m = 0$ ist trivial.

□

Um nun tatsächlich Entscheidbarkeit von $\mathcal{T}_<$ zu bekommen, fehlt uns noch die Entscheidbarkeit der Erfüllbarkeit quantorenfreier Formeln über $\mathcal{T}$. Wir beobachten zunächst, dass bei der Quantorenelimination für $\mathcal{T}_<$ auch $\neg$ eliminiert wird. Ferner können wir mit Disjunktionen stets umgehen, indem wir zunächst in DNF transformieren und dann die Disjunkte der DNF der Reihe nach auf Erfüllbarkeit prüfen (natürlich hören wir damit auf, sobald wir einen erfüllbaren Disjunkt gefunden haben). Das verbleibende Problem ist, die Erfüllbarkeit einer Konjunktion atomarer Formeln der Form $X = Y$ oder $X < Y$ zu zeigen. Da wir jetzt nur noch erfüllbarkeitsäquivalent umformen müssen, können wir atomare Formeln $X = Y$ jeweils entfernen und in der verbleibenden Formel X durch Y substituieren (das ist nicht mehr äquivalent, da ja die Aussage $X = Y$ verloren geht, aber offenbar erfüllbarkeitsäquivalent). Zur Vereinfachung der Notation schreiben wir die verbleibende Konjunktion einfach als Menge C von Formeln der Form $X < Y$. Folgender Algorithmus entscheidet Erfüllbarkeit von C über $\mathcal{T}_<$:

1. Wenn in C Formeln $X < Y$,$Y < Z$ enthalten sind mit $X < Z$ nicht in C, füge $X < Z$ zu C hinzu und mache bei Schritt 1 weiter.

2. Falls C eine Formel der Form $X < X$ enthält, antworte „unerfüllbar“, sonst „erfüllbar“.

Schritt 1 terminiert, da nur endlich (genauer: quadratisch) viele Formeln $X < Z$ hinzugefügt werden können. Per Transitivität und Irreflexivität ist klar, dass der Algorithmus recht hat, wenn er „unerfüllbar“ antwortet. Um zu sehen, dass er auch im anderen Fall recht hat, zeigen wir

Lemma 92. *Wenn mit $X < Y$ und $Y < Z$ stets auch $X < Z$ in C ist und C keine Formel der Form $X < X$ enthält, dann ist C erfüllbar über $\mathcal{T}_<$.*

Beweis. Ein schnelles Argument, das allerdings etwas Algorithmik voraussetzt, läuft wie folgt: Nach Voraussetzung ist $<$ eine strikte partielle Ordnung auf den Variablen. Man kann dann die Variablen topologisch sortieren, d.h. in einer Folge $X_1, \ldots, X_n$ anordnen, so dass $i < j$, wenn $X_i < X_j$ in C ist. Dann $(\mathbb{Q}, <), \eta \models C$ mit $\eta(X_i) = i$.

Argument ohne topologisches Sortieren: Induktion über die Anzahl Variablen in C. Wähle eine in C vorkommende Variable und setze $C \setminus X = \{Y < Z \in C \mid Y, Z \text{ verschieden von } X\}$. Damit erfüllt $C \setminus X$ offenbar die Voraussetzung des Lemmas, ist also nach Induktionsvoraussetzung erfüllbar über $\mathcal{T}_<$, da es eine Variable weniger als C enthält. Seien also $\mathfrak{M}$ ein Modell von $\mathcal{T}_<$ und η eine Umgebung in $\mathfrak{M}$ mit $\mathfrak{M}, \eta \models C \setminus X$. Seien nun $U_1, \ldots, U_n$ die Variablen mit $U_i < X \in C$ und $V_1, \ldots, V_m$ die mit $X < V_j \in C$. Unter den $\eta(U_i)$ gibt es per Trichotomie einen größten Wert $l \in M$, und unter den $\eta(V_j)$ einen kleinsten, $r \in M$. Es gilt dann per Transitivität $l < r$ in $\mathfrak{M}$ (wobei wir $\mathfrak{M}[\![<]\!]$ einfach als $<$ schreiben), so dass per Dichte $x \in M$ existiert mit $l < x$ und $x < r$. Da $X < X \notin C$, gilt dann per Transitivität $\mathfrak{M}, \eta[X \mapsto x] \models C$. □

10 Vollständigkeit der Prädikatenlogik erster Stufe

Zur Vereinfachung arbeiten wir in Prädikatenlogik ohne Gleichheit (=) und ohne Funktionssymbole positiver Stelligkeit (aber mit Konstanten). Wir beweisen die Umkehrung des Korrektheitssatzes, d.h.

Satz 93 (Vollständigkeit der Prädikatenlogik erster Stufe)**.** *Sei Φ eine Menge von Sätzen und ψ ein Satz über einer Signatur Σ in Prädikatenlogik erster Stufe. Wenn ψ logische Folgerung aus Φ ist ($\Phi \models \psi$), dann ist ψ aus Φ herleitbar ($\Phi \vdash \psi$).*

Bemerkung 94. Eine Version des Satzes für beliebige Formeln (d.h. solche mit freien Variablen) ist leicht reduzierbar auf die obige Version, indem man einfach Variablen durch Konstanten ersetzt.

Wie schon im Falle der Auszsagenlogik ist der Beweis der Vollständigkeit wesentlich schwieriger als der der Korrektheit. Der Beweis, wie wir ihn hier präsentieren, geht auf Leon Henkin zurück; er basiert auf einer Reduktion der Prädikatenlogik erster Stufe auf die Aussagenlogik, für die wir die Vollständigkeit ja schon gezeigt haben. Im Groben besteht der Beweis aus den folgenden Schritten:

(A) Wir erweitern zunächst die Signatur Σ zu einer Signatur Σ_H, die für jede Formel $\phi(X)$ über Σ_H eine *Zeugenkonstante* $c_{\varphi(X)}$ enthält, gedacht als Name für ein $\phi(X)$ erfüllendes Element, wenn ein solches existiert. Ebenso erweitern wir die gegebene Formelmenge Φ um eine unendliche Menge $\mathscr{H}$ zusätzlicher Axiome, die *Henkin-Theorie*. Diese enthält insbesondere für jede Formel $\phi(X)$ ein Axiom $(\exists X.\, \varphi(X)) \to \varphi(c_{\varphi(X)})$.

(B) (*Henkin-Elimination*) Wir beweisen dann für jede Formel ρ über Σ (die also keine Zeugenkonstanten erwähnt), dass aus $\Phi \cup \mathscr{H} \vdash \rho$ bereits $\Phi \vdash \rho$ folgt, in Worten: wenn sich ρ aus Φ unter Zuhilfenahme der Henkin-Theorie herleiten lässt, dann kann man ρ schon aus Φ alleine herleiten .

(C) (*Henkin-Konstruktion*) Wir fassen nun Formeln in Logik erster Stufe als bloße aussagenlogische Formeln auf, indem wir alle Teilformeln der Form $\forall X.\, \varphi$, $\exists X.\, \varphi$ oder $P(\dots)$ als Atome ansehen. Aus einer Wahrheitsbelegung κ mit $\kappa \vDash \Phi \cup \mathscr{H}$ (*aussagenlogische* Erfülltheit) konstruieren wir dann ein Σ-Modell $\mathfrak{M}_\kappa$ mit $\mathfrak{M}_\kappa \vDash \Phi$ (Erfülltheit *in Logik erster Stufe*).

Damit läuft der Beweis des Vollständigkeitssatzes dann wie folgt: Wie schon im Falle der Aussagenlogik reicht es zu zeigen, dass jede konsistente Menge Φ von Sätzen in Logik erster Stufe erfüllbar ist. Per Henkin-Elimination ist mit Φ auch $\Phi \cup \mathscr{H}$ konsistent — als Menge von Formeln in Logik erster Stufe, und damit erst recht als Menge von aussagenlogischen Formeln, da ja nach der Uminterpretation allenfalls weniger Information und weniger deduktive Mittel als vorher zur Verfügung stehen, um einen Widerspruch herzuleiten. Per Vollständigkeit der Aussagenlogik ist damit $\Phi \cup \mathscr{H}$ erfüllbar als Menge aussagenlogischer Formeln; nach der Henkinkonstruktion folgt dann, dass Φ als Menge von Formeln in Logik erster Stufe erfüllbar ist.

Im einzelnen werden die Schritte der Beweisstrategie wie folgt umgesetzt.

(A) Wir brauchen für jede Formel $\varphi(X)$ eine Zeugenkonstante $c_{\varphi(x)}$, *aber*: Zeugenkonstanten erzeugen neue Formeln, z.B. $\exists y.P\left(y, c_{\varphi(c)}\right)$. Die Lösung besteht darin, die Hinzufügung von Zeugenkonstanten zu iterieren. Wir definieren also eine aufsteigende Folge $\Sigma = \Sigma_0 \subseteq \Sigma_1 \subseteq \Sigma_2 \subseteq \dots$ von Signaturen und setzen $\Sigma_H = \bigcup_{i=0}^{\infty} \Sigma_i$. Wir können dann für jedes $c_{\varphi(x)}$ einen *Geburtstag*

$$\min\{i \mid c_{\varphi(x)} \in \Sigma_i\}$$

definieren. Damit sieht man auch, dass Σ_H in der Tat für jede Formel $\phi(X)$ über Σ_H eine Zeugenkonstante $c_{\phi(X)}$ enthält: wenn i der Geburtstag der jüngsten Zeugenkonstante in $\phi(X)$ ist, dann enthält $\Sigma_{i+1} \subseteq \Sigma_H$ eine Zeugenkonstante $c_{\phi(X)}$.

Die Henkin-Theorie $\mathscr{H}$ ist dann definiert als die (unendliche) Menge aller Instanzen der folgenden drei Axiomenschemata:

H1: $(\exists x.\varphi(x)) \to \varphi(c_{\varphi(c)})$

H2: $\varphi(c) \to \exists x.\varphi(x)$

H3: $\neg\forall x.\varphi(x) \leftrightarrow \exists x.\neg\varphi(x)$

(B) H2 und H3 sind beweisbar in Logik erster Stufe und können daher offensichtlich in Beweisen in Logik erster Stufe als Annahmen weggelassen werden. Es bleibt zu zeigen, dass alle Instanzen von H1 eliminierbar sind. Wir wählen in einem gegebenen Beweis von ρ die Instanz von H1 mit dem jüngstem $c_{\varphi(x)}$. Wir bezeichnen die Menge aller anderen im Beweis vorkommenden Instanzen von H1 mit $\mathscr{H}_0$, so dass

$$\Phi \cup \mathscr{H}_0 \cup \{(\exists X.\,\varphi(X)) \to \varphi(c_{\varphi(X)})\} \vdash \rho.$$

Da $(\exists X.\,\varphi(X)) \to \varphi(c_{\varphi(X)})$ sowohl aus $\neg\exists X.\,\varphi(X)$ als auch aus $\varphi(c_{\varphi(X)})$ herleitbar ist, folgt

$$\Phi \cup \mathscr{H}_0 \cup \{\neg\exists X.\,\varphi(X)\} \vdash \rho \text{ und } \Phi \cup \mathscr{H}_0 \cup \{\varphi(c_{\varphi(X)})\} \vdash \rho.$$

Da $c_{\phi(X)}$ in Φ, $\mathscr{H}_0$ und ρ nicht vorkommt (in $\mathscr{H}_0$ deswegen nicht, weil $c_{\phi(X)}$ die jüngste vorkommende Zeugenkonstante ist), folgt aus dem rechten Teil per ($\exists$E)

$$\Phi \cup \mathscr{H}_0 \cup \{\exists X.\,\varphi(X)\} \vdash \rho,$$

also per Fallunterscheidung über $\exists X.\,\varphi(X) \vee \neg\exists X.\,\varphi(X)$ letztlich $\Phi \cup \mathscr{H}_0 \vdash \rho$. Damit ist eine Instanz von H1 eliminiert; Iterieren des Verfahrens eliminiert alle Instanzen von H1.

(C): Wir setzen

$$\begin{aligned} M_\kappa &= \text{Menge der Konstanten in } \Sigma_{\mathscr{H}} \\ \mathfrak{M}_\kappa(c) &= c \\ \mathfrak{M}_\kappa(P) &= \{(c_1,\ldots,c_n) \mid \kappa(P(c_1,\ldots,c_n)) = \top\} \end{aligned}$$

Die Behauptung $\mathfrak{M}_\kappa \models \Phi$ folgt dann unmittelbar aus dem

Wahrheitslemma: Für jeden Satz ρ gilt $\mathfrak{M}_\kappa \vDash \rho$ genau dann, wenn $\kappa \vDash \rho$.

Beweis. Induktion über die *Größe* von ρ, die wir definieren, indem wir 3 für $\forall X$ und 1 für $\exists X, \neg, \wedge$ zählen. Der Induktionsanfang ist $\rho = P(c_1,\ldots,c_n)$; für solche Formeln gilt die Behauptung nach Konstruktion von $\mathfrak{M}_\kappa$. Die Booleschen Fälle ($\neg, \wedge$) sind klar. Der Fall $\rho \equiv \forall X.\,\varphi(x)$ wird unter Ausnutzung der Definition der Größe von ρ durch Umformung von $\forall$ in $\neg\exists\neg$ erledigt; im Detail:

$$\begin{aligned} \mathfrak{M}_\kappa \vDash \forall X.\,\varphi(X) &\Leftrightarrow \mathfrak{M}_\kappa \nvDash \underbrace{\overbrace{\exists X.\,\neg}^{2}\varphi(X)}_{\text{kleiner als } \underbrace{\forall X}_{3}.\,\varphi(X)} \\ &\overset{IV}{\Leftrightarrow} \kappa \nvDash \exists X.\,\neg\varphi(X) \\ &\overset{H3}{\Leftrightarrow} \kappa \nvDash \neg\forall X.\,\varphi(X) \\ &\Leftrightarrow \kappa \vDash \forall X.\,\varphi(X). \end{aligned}$$

Es bleibt der Fall $\rho \equiv \exists X.\,\varphi(x)$; wir handeln die Implikationen getrennt ab:

„⇒“: Sei $\mathfrak{M}_\kappa \models \exists X.\, \phi(X)$. Dann existiert $c \in \Sigma_{\mathscr{H}}$ mit $\mathfrak{M}_\kappa \vDash \varphi(c)$. Da die Formel $\varphi(c)$ kleiner als $\exists X.\, \varphi(X)$ ist, folgt nach Induktionsvoraussetzung $\kappa \vDash \varphi(c)$. Mit H2 erhalten wir schließlich $\kappa \vDash \exists X.\, \varphi(X)$.

„⇐“: Sei $\kappa \vDash \exists X.\, \varphi(X)$. Nach H1 folgt dann $\kappa \vDash \varphi(c_{\varphi(x)})$; da die Formel $\varphi(c_{\varphi(x)})$ kleiner als $\exists X.\, \phi(X)$ ist, folgt nach Induktionsvoraussetzung $\mathfrak{M}_\kappa \vDash \varphi(c_{\varphi(x)})$ und damit $\mathfrak{M}_\kappa \vDash \exists X.\, \varphi(X)$. □

Es folgt nunmehr erneut

Korollar 95 (Kompaktheit)**.** *Jede* endlich erfüllbare *Formelmenge* Φ *(d.h. jede endliche Teilmenge von* Φ *ist erfüllbar) ist erfüllbar*

Beweis. Da Beweise endliche Objekte sind, gilt die analoge Eigenschaft mit „konsistent“ statt „erfüllbar“; nach Vollständigkeit und Korrektheit sind aber Konsistenz und Erfüllbarkeit äquivalent. □

Beispiel 96. Sei Φ eine Axiomatisierung der natürlichen Zahlen in Logik erster Stufe (z.B. die Peano-Axiome, mit 0 und Sukzessorfunktion s sowie einem Axiomenschema für Induktion). Dann ist $\Phi \cup \{c > s^n(0) | n \in \mathbb{N}\}$ endlich erfüllbar, somit nach Kompaktheit erfüllbar — d.h. wir können in Logik erster Stufe durch keine noch so raffinierte Axiomatisierung die Existenz von Nicht-Standard-Zahlen ausschließen.

Bemerkung 97. Vollständigkeit ist durchaus keine selbstverständliche Eigenschaft einer Logik. Man unterscheidet im allgemeinen zwischen *starker Vollständigkeit*, wie wir sie oben für Prädikatenlogik erster Stufe bewiesen haben, und *schwacher Vollständigkeit*. Eine Logik, mit logischer Folgerungsrelation $\models$, ist per Definition stark vollständig, wenn aus $\Phi \models \psi$ stets $\Phi \vdash \psi$ folgt (*Korrektheit* ist die umgekehrte Implikation), und schwach vollständig, wenn jede gültige Formel herleitbar ist, d.h. wenn aus $\models \psi$ stets $\vdash \psi$ folgt. Wenn eine Logik korrekt und stark vollständig ist und außerdem das Beweissystem *finitär* ist, d.h. wenn jede Regel nur endlich viele Prämissen hat (das schließt Regeln aus, die z.B. aus $\phi(n)$ für alle konkreten natürlichen Zahlen n die Formeln $\forall x \in \mathbb{N}.\, \phi(x)$ schließen), dann ist die Logik, mit der gleichen Argumentation wie oben, *kompakt*, d.h. jede endlich erfüllbare Formelmenge ist erfüllbar.

In *monadischer Prädikatenlogik zweiter Stufe* (MSO) hat man zusätzlich zu Quantoren über Individuenvariablen wie in Prädikatenlogik erster Stufe noch Variablen für Teilmengen des Grundbereichs, also für unäre Prädikate, und Quantoren darüber. Wir nennen die neuen Variablen *Mengenvariablen* und schreiben sie $P, Q, \ldots$; wir verwenden sie ansonsten wie Prädikatensymbole. Sei dann Φ die Formelmenge über der Signatur $\Sigma = \{0, suc_\lrcorner, <\}$ bestehend aus den Formeln

$$\forall P.\, ((P(0) \wedge \forall x.(P(x) \to P(s(x)))) \to \forall x.\, P(x)$$
$$\forall x.\, \neg s(x) = 0$$
$$\forall x, y.\, (s(x) = s(y) \to x = y)$$
$$\forall x.\, ((0 < x \vee 0 = x) \wedge \neg x < 0)$$
$$\forall x, y.(s(x) < s(y) \leftrightarrow x < y).$$

Dann definiert Φ eindeutig (bis auf irrelevante Umbenennungen der Elemente des Grundbereichs) die natürlichen Zahlen mit der üblichen Lesart von $<$: das zweite und dritte Axiom stellen sicher, dass man lauter verschiedene Elemente $s^n(0)$ für $n \geq 0$ hat, das erste Axiom

sorgt dafür, dass dies tatsächlich alle Elemente sind, und die letzten beiden Axiom definieren $<$ per Rekursion. Damit ist dann offenbar

$$\Phi \cup \{x < s^n(0) \mid n \in \mathbb{N}\}$$

endlich erfüllbar, aber nicht erfüllbar. Somit hat MSO keine stark vollständige finitäre Axiomatisierung. (Nach dem deutlich schwerer zu beweisenden Gödelschen Unvollständigkeitssatz ist MSO auch nicht schwach vollständig.)