Logik für Studierende der Informatik

Wintersemester 2025
Albert-Ludwigs-Universität Freiburg

Heike Mildenberger

Fassung vom 4.2.2026

Warnung, das Skript kann Fehler enthalten. Ich danke Herrn Dodillet, Herrn Jakob und Herrn Klemm für wichtige Hinweise.

Inhaltsverzeichnis

Kapitel 1

Aussagenlogik

1.1 Literaturempfehlungen

Die folgenden Bücher stehen immer in der Mathematikbücherei und können nicht ausgeliehen werden. Sie sind eingeladen, Lesebesuche abzustatten. Außerdem gibt es alle drei Werke auch in der Bibliothek der Technischen Fakultät.
H.-D. Ebbinghaus, J. Flum, W. Thomas. *Einführung in die Mathematische Logik*. Hochschultaschenbuch, 4 edition, 1996. [1]
Herbert Enderton. *A Mathematical Introduction to Logic.* Academic Press, 3 edition, 2001.[2]
Michael Sipser. *Introduction to the Theory of Computation*, PWS Publishing Company, Boston, 1997. [8]

Folgendes ausgezeichnete Lehrbuch gibt es in der Bibliothek für Mathematik und in der Lehrbuchsammlung II:
Martin Ziegler *Mathematische Logik*, Birkhäuser, Mathematik kompakt, 2010. [9]

Außerdem stehen einige sehr gute Bücher und Artikel in den Literaturangaben. Sie sollten sich keinesfalls nur auf dieses Skript beschränken!

1.2 Die Bestandteile

Wir beginnen jetzt die formale Entwicklung der Aussagenlogik (propositional logic, sententional logic).

Die *natürlichen Zahlen* sind $\mathbb{N} = \{0, 1, 2, \dots\}$. Wir fassen jede natürliche Zahl n auch als Vertreter einer Menge mit n Elemente auf, und nehmen oft eine einfache Menge mit n Elementen, nämlich die Menge der Vorgänger von n, also $n = \{0, 1, \dots, n-1\}$.

Definition 1.1. Die *Symbole der Aussagenlogik* sind wie folgt:

(a) Klammern: ((linke Klammer, Linksklammer, öffnende Klammer) und) (rechte Klammer, Rechtsklammer, schließende Klammer)

(b) Junktoren (propositional connectives): $\neg$ (nicht), $\wedge$ (und), $\vee$ oder, $\rightarrow$ (wenn ... dann), $\leftrightarrow$ (genau dann, wenn).

(c) Satzsymbole, auch Variable genannt: A_0, A_1,... Wir lassen, wenn nicht anders genannt, immer abzählbar unendlich (genaueres zu diesem Begriff findet man in Definition 1.34 und kurz danach) viele, durch $n \in \mathbb{N} = \{0, 1, \dots\}$ indizierte Variable $\{A_n \mid n \in \mathbb{N}\}$ zu.

(d) Das Symbol Verum $\top$ und das Symbol Falsum $\bot$. Manche Autoren lassen diese weg, und ersetzen $\top$ durch $(A_0 \vee \neg A_0)$ and $\bot$ durch $(A_0 \wedge \neg A_0)$.

Definition 1.2. Ein *Ausdruck* ist eine endliche Folge von Symbolen.

Zum Beispiel ist $\rightarrow ((A_4$ ein Ausdruck. Aber nur gewisse Ausdrücke haben eine Bedeutung, und diese werden Formeln genannt.

Definition 1.3. Wir definieren die Menge der *Formeln (in der Infix-Notation) der Aussagenlogik* wie folgt:

(a) Jedes Satzsymbol ist eine Formel und auch $\top$ und $\bot$ sind Formeln.

(b) Wenn α und β Formeln sind, dann sind auch $\neg\alpha$ und $(\alpha \wedge \beta)$ und $(\alpha \vee \beta)$, $(\alpha \rightarrow \beta)$ und $(\alpha \leftrightarrow \beta)$ Formeln.

(c) Kein Ausdruck ist dann eine Formel, wenn er es nicht aufgrund von (a) oder (b) sein muss.

Wir sagen $(\alpha \wedge \beta)$ entsteht aus α und β durch Anwendung des Junktors $\wedge$. Indem wir die Junktoren (im Falle von $\neg$) als einstellige oder zweistellige Funktionen auffassen, können wir auch von unter diesen Funktionen abgeschlossenen Mengen sprechen.

Definition 1.4. Sei f eine n-stellige Funktion, deren Definitionsbereich irgendeine Menge ist. Die Menge M ist *unter f abgeschlossen*, genau dann, wenn für alle $\vec{m}$ in M^n, falls $f(\vec{m})$ definiert ist, auch $f(\vec{m})$ ein Element von M ist.

Wir schreiben
$$f[X] = f''X = \{f(x) \mid x \in X\}$$
und nennen diese Menge *die Bildmenge von f angewandt auf X*.

Bemerkung: Beide Schreibweisen, $f[X]$ und $f''X$ sind eingebürgert. Beachten Sie, dass $f(X) \neq f[X]$ sehr wohl möglich ist.

Lemma 1.5. *Sei M eine Menge, sei $n \in \mathbb{N}$ und sei f eine n-stellige Funkion. Dann gibt es eine kleinste unter f abgeschlossenen Obermenge von M.*

Beweis. Wir definieren induktiv über $k \in \mathbb{N}$ eine Folge $\langle M_k \mid k \in \mathbb{N}\rangle = (M_k)_{k\in\mathbb{N}}$: $M_0 = M$, $M_{k+1} = M_k \cup f''M_k^n$. Dann ist $M_\infty = \bigcup_{k\in\mathbb{N}} M_k$ unter f abgeschlossen. Für jede gegen f abgeschlossene Menge A gilt: $A \supseteq M_\infty$, denn induktiv folgt für alle k, $A \supseteq M_k$. $\dashv$

Entsprechend kann man „abgeschlossen unter einer Menge von Funktionen" definieren. Eine andere Art, Definition 1.3(c) auszudrücken, ist zu sagen, dass die Menge der Formeln dadurch erzeugt wird, dass die „Menge der Satzsymbole unter den Junktoren abgeschlossen wird". Da der (im Sinne von $\subseteq$) kleinste Abschluss, nämlich der mit obigem Schema gebildete, genommen wird, gilt folgendes:

Satz 1.6. Induktionsprinzip für Eigenschaften *von Formeln. Wenn eine Eigenschaft für die Satzsymbole wahr ist und bei Anwendung der Junktoren erhalten bleibt, so ist sie für jede Formel wahr.*

Zum Beispiel nehmen wir folgende Eigenschaft:

Lemma 1.7. *Jede Formel hat gleich viele Linksklammern wie Rechtsklammern.*

Beweis: Die Behauptung ist wahr für Satzsymbole, da diese keine Klammern enthalten. Wenn α und β Formeln mit der behaupteten Eigenschaft sind, dann haben auch $\neg\alpha$, $(\alpha \wedge \beta)$, $(\alpha \vee \beta)$, $(\alpha \to \beta)$ und $(\alpha \leftrightarrow \beta)$ die behauptetete Eigenschaft. Wegen des Induktionsprinzips hat jede Formel die gewünschte Eigenschaft. $\dashv$

Lemma 1.8. *Keine Formel ist echtes Anfangsstück einer anderen Formel.*

Beweis: Die Behauptung ist eine Aussage über alle Paare von Formeln. Wir führen den Beweis induktiv über den Aufbau der längeren Formel simultan für alle ihre Anfangsstücke. Die Behauptung ist wahr für Satzsymbole, da diese keine nicht-leeren echten Anfangsstücke haben. Wir zeigen induktiv:

($*$) Jedes nicht-leere echte Anfangsstück einer Formel hat strikt mehr Linksklammern als Rechtsklammern oder nur $\neg$-Zeichen.

Die Behauptung für $\neg\alpha$ folgt unmittelbar aus der Behauptung für α. Jedes echte Anfangsstück von $(\alpha \vee \beta)$ ist entweder ein echtes Anfangssück von α oder ragt in β hinein. Im ersten Fall folgt die Aussage aus der Induktionsvoraussetzung von ($*$) und der Tatsache, dass am Kopf des Anfangsstücks eine unpaarige Linksklammer steht. Im zweiten Fall, wenn das Anfangsstück in β hineinragt, dann gibt es nach dem vorigen Lemma zu Beginn von β im Anfangsstück eine Linksklammer mehr als Rechtsklammern. Der Teil des Anfangsstücks nach dem Beginn von β hat nach der Induktionsannahme ($*$) für β wieder echt mehr Linksklammern als Rechtsklammern, oder, falls es ganz β ist, gleich viele Linksklammern wie Rechtsklammern. Auf jeden Fall bleibt eine unpaarige Linksklammer übrig. $\dashv$

Übung 1.9. Überlegen Sie sich, dass man statt der Klammern auch eine *polnische Notation* einführen könnte: Die Regeln für Aussagenvariable und für die Negation sind auch in der polnischen Notation wie oben. Jedoch $\wedge\alpha\beta$ entspricht $(\alpha \wedge \beta)$, usf. Ist die polnische Notation eindeutig lesbar? Warum?

1.3 Wahrheitsbelegungen

Wir definieren, was es bedeutet, dass eine Formel aus anderen Formeln logisch folgt.

Definition 1.10. Es bedeuten W wahr und F falsch. Sie werden die *Wahrheitswerte* genannt.

Definition 1.11. Eine *Wahrheitsbelegung* v für eine Menge $\mathcal{S}$ von Satzsymbolen ist eine Funktion

$$v\colon \mathcal{S} \to \{W, F\}.$$

Sei $\bar{\mathcal{S}}$ der Abschluss von $\mathcal{S}$ unter den fünf Junktoren. Wenn $\mathcal{S} = \{A_i \mid i \in \mathbb{N}\}$ dann ist $\bar{\mathcal{S}}$ also die Menge aller Formeln.

Satz 1.12. Induktionsprinzip für Definitionen. *Induktiv über den Aufbau der Formeln lassen sich Eigenschaften von Formeln auf der Menge $\bar{\mathcal{S}}$ definieren, indem man festlegt, für welche $A \in \mathcal{S}$ die Eigenschaft zutrifft, und festlegt, wie sich das Zutreffen der Eigenschaft bei Anwendung eines Junktors fortpflanzt. Induktiv über den Aufbau der Formeln lassen sich Funktionen auf der Menge $\bar{\mathcal{S}}$ definieren, indem man die Funktion auf $\mathcal{S}$ definiert und festlegt, wie sich die Funktionswerte bei der Anwendung eines Junktors verhalten.*

Eine erste Anwendung ist:

Definition 1.13. Wir definieren eine *Erweiterung $\bar{v}$ von v auf $\bar{\mathcal{S}}$* wie folgt:

(1) $\bar{v}(A) = v(A)$ für $A \in \mathcal{S}$,

(2) $\bar{v}(\neg\alpha) = W$:gdw $\bar{v}(\alpha) = F$,

(3) $\bar{v}(\alpha \wedge \beta) = W$:gdw $(\bar{v}(\alpha) = W$ und $\bar{v}(\beta) = W)$,

(4) $\bar{v}(\alpha \vee \beta) = W$:gdw $(\bar{v}(\alpha) = W$ oder $\bar{v}(\beta) = W)$,

(5) $\bar{v}(\alpha \to \beta) = W$:gdw $(\bar{v}(\alpha) = F$ oder $\bar{v}(\beta) = W)$,

(6) $\bar{v}(\alpha \leftrightarrow \beta) = W$:gdw $(\bar{v}(\alpha) = W$ gdw $\bar{v}(\beta) = W)$.

$\bar{v}$ wird manchmal auch Wahrheitsbelegung genannt, so wie wir schon v Wahrheitsbelegung genannt haben. (Da $\bar{v}$ sich eindeutig aus v ergibt, ist diese Benennung nicht gefährlich.)

Die obige Definition lässt sich auch in den bekannten *Wahrheitstafeln für Junktoren* schreiben mit der Konvention, dass der Wahrheitswert von α in der linken Spalte steht und der von β in der oberen Zeile (dies ist bei asymmetrischen Junktoren wichtig):

$\neg$	W	F
	F	W

$\wedge$	W	F
W	W	F
F	F	F

$\vee$	W	F
W	W	W
F	W	F

$\to$	W	F
W	W	F
F	W	W

$\leftrightarrow$	W	F
W	W	F
F	F	W

Überlegen Sie sich: Nach dem Induktionsprinzip ist durch Definition 1.13 $\bar{v}$ auf ganz $\bar{\mathcal{S}}$ wohldefiniert.

Konvention. Wir lassen die äußerste Klammer um eine Formel machmal weg. Beim weiteren Zusammensetzen schreiben wir die Klammer dann natürlich, denn wir wollen ja die eindeutige Lesbarkeit garantieren.

Beispiel 1.14. Als Beispiel für das Ausrechnen von $\bar{v}$ betrachten wir folgende Formel α:

$$\alpha = ((A_2 \to (A_1 \to A_6)) \leftrightarrow ((A_2 \wedge A_1) \to A_6))$$

mit der Wahrheitsbelegung v für A_1, A_2, A_6: $v(A_1) = W$, $v(A_2) = W$, $v(A_6) = F$. Wir rechnen nun die Erweiterung $\bar{v}$ von v an der Stelle α aus: $\bar{v}(A_1 \to A_6) =$

F, $\bar{v}(A_1 \wedge A_2) = W$, $\bar{v}((A_2 \wedge A_1) \to A_6)) = F$, $\bar{v}((A_2 \to (A_1 \to A_6))) = F$, und wir erhalten schließlich $\bar{v}(\alpha) = W$.

Definition 1.15. Wir sagen, dass *eine Wahrheitsbelegung v eine Formel φ erfüllt*, wenn $\bar{v}(\varphi) = W$ ist. Dazu muss natürlich der Definitionsbereich von v jedes Satzsymbol in φ enthalten.

1.4 Tautologische Implikation und äquivalente Formeln

Nun betrachten wir eine Formelmenge Σ, die als Menge der Voraussetzungen fungiert, und eine weitere Formel τ, die eine Schlussfolgerung sein kann.

Definition 1.16. Wir sagen Σ *impliziert tautologisch* τ oder Σ *impliziert* τ, in Zeichen $\Sigma \models \tau$, gdw folgendes der Fall ist: Für jede Wahrheitsbelegung v aller Symbole, die in Σ oder in τ auftreten, gilt: Wenn $\bar{v}$ jedes Element von Σ erfüllt, dann erfüllt $\bar{v}$ auch τ.

Falls Σ die leere Menge ist, dann bedeutet das, dass jede Wahrheitsbelegung der Satzsymbole in τ die Formel τ erfüllt. In diesen Falle sagen wir, dass τ eine (aussagenlogische) Tautologie ist und schreiben $\models \tau$. Noch einmal explizit:

Definition 1.17. (a) τ ist *allgemeingültig oder eine Tautologie*, gwd für alle Wahrheitsbelegungen $v \colon S \to \{W, F\}$, $\bar{v}(\tau) = W$.

(b) τ ist *erfüllbar*, gwd es eine Wahrheitsbelegung $v \colon S \to \{W, F\}$ gibt, so dass $\bar{v}(\tau) = W$.

Beobachtung 1.18. *$\top$ is allgemeingültig. $\bot$ ist nicht erfüllbar.*

Wenn Σ nur ein Element enthält, dann schreiben wir $\sigma \models \tau$ statt $\{\sigma\} \models \tau$.

Definition 1.19. Wenn sowohl $\sigma \models \tau$ als auch $\tau \models \sigma$, dann sagen wir, dass σ und τ *(tautologisch) äquivalent* sind. Wir schreiben $\sigma \equiv \tau$.

Beobachtung 1.20. *(1) $\sigma \equiv \tau$ gdw $\sigma \leftrightarrow \tau$ allgemeingültig ist.*

(2) σ is allgemeingültig gdw $\sigma \equiv \top$.

(3) σ is erfüllbar gdw $\sigma \not\equiv \bot$.

(4) $\equiv$ ist eine Äquivalenzrelation auf der Menge der Formeln.

Einschub: Sie wundern sich vielleicht über den Spielraum der Festlegung, welche Junktoren zum Aufbau der aussagenlogischen Formeln zulässig sind. Die folgenden Übungsaufgaben zeigen, dass es viel Spielraum gibt:

Definition 1.21. Eine Junktorenmenge $\mathcal{J}$ heißt *vollständig*, gdw es zu jeder aussagenlogischen Formel σ eine (tautologisch) äquivalente aussagenlogische Formel τ gibt, die nur Junktoren aus $\mathcal{J}$ enthält.

Ü1. Für den Junktor $|$ („Scheffer stroke", „weder noch") gilt: $\bar{v}((\varphi|\psi)) = W$ gdw $\bar{v}(\varphi) = F$ und $\bar{v}(\psi) = F$ für alle Erweiterungen $\bar{v}$ von Wahrheitsbelegungen v. Zeigen Sie, dass $\{|\}$ eine vollständige Junktorenmenge ist.

Ü2. Zeigen Sie, dass $\{\neg, \rightarrow\}$ eine vollständige Junktorenmenge ist. Analoges gilt für $\vee$ anstelle von $\rightarrow$ und für $\wedge$ anstelle von $\rightarrow$. Wie ist die Lage für $\leftrightarrow$ anstelle von $\rightarrow$?

Ü3.* Für den Junktor $+$ („entweder oder") gilt: $\bar{v}((\varphi + \psi)) = W$ gdw $\bar{v}(\varphi) = W$ oder $\bar{v}(\psi) = W$, aber nicht beide wahr, für alle Erweiterungen $\bar{v}$ von Wahrheitsbelegungen v. Zeigen Sie, dass $\{\wedge, \leftrightarrow, +\}$ eine vollständige Junktorenmenge ist und — ist der *-Teil der Aufgabe — dass jedoch keine ihrer echten Teilmengen vollständig ist.

Für schlanke Beweise über Eigenschaften von Formeln, die bei äquivalenten Formeln gleich entschieden werden, bietet es sich daher an, mit der recht kleinen vollständigen Junktorenmenge $\{\neg, \wedge\}$ zu arbeiten. Dies werden wir in manchen Induktionsbeweisen aus Sparsamkeitsgründen tun.

Definition 1.22. Sei φ eine Formel, aufgebaut aus $\wedge$, $\vee$, $\neg$, $\top$ und $\bot$. Die *duale Formel* φ^* ensteht durch Vertauschen von $\vee$ und $\wedge$, $\top$ und $\bot$.

Lemma 1.23. *$\varphi \equiv \psi$ gdw $\varphi^* \equiv \psi^*$.*

Beweis durch Induktion über den Aufbau der Formeln. $\dashv$

Satz 1.24. *Für Variable A, B, C gelten die folgenden Grundäquivalenzen:*

$$\begin{array}{rcll}
A \wedge A & \equiv & A & \textit{Idempotenz} \\
A \wedge B & \equiv & B \wedge A & \textit{Kommutativität} \\
((A \wedge B) \wedge C) & \equiv & (A \wedge (B \wedge C)) & \textit{Assoziativität} \\
(A \wedge (A \vee B)) & \equiv & A & \textit{Absorption} \\
(A \wedge (B \vee C)) & \equiv & (A \wedge B) \vee (A \wedge C) & \textit{Distributivität} \\
\bot \wedge A & \equiv & \bot & \textit{Kleinstes Element} \\
\top \wedge A & \equiv & A & \textit{Größtes Element}
\end{array}$$

Da das Assoziativgesetz gilt, sind folgende Abkürzungen wohldefiniert: Sei $I = \{i_m \,|\, m < n\}$ eine nicht-leere endliche Menge. Wir schreiben

$$\begin{array}{rcl}
\bigwedge_{i \in I} \varphi_i & := & \varphi_{i_0} \wedge \cdots \wedge \varphi_{i_{n-1}}, \\
\bigvee_{i \in I} \varphi_i & := & \varphi_{i_0} \vee \cdots \vee \varphi_{i_{n-1}},
\end{array}$$

für irgendeine Klammerung des Ausdrucks auf der rechten Seite, die aus ihm eine Formel macht. Wieviele solche Klammerungen gibt es?

Wir setzen

$$\begin{array}{rcl}
\bigwedge_{i \in \emptyset} \varphi_i & := & \top, \\
\bigvee_{i \in \emptyset} \varphi_i & := & \bot,
\end{array}$$

Wir benutzen die Schreibweise auch für Formelmengen, die nicht durch Indizes beschrieben werden: $\bigwedge \Phi = \bigwedge_{\varphi \in \Phi} \varphi$ usf. Im Falle eines unendlichen Φ gehört

$\bigwedge \Phi$ allerdings nicht zur Aussagenlogik, sondern zu einer sogenannten infinitären Sprache.

Die Ersetzung ist eine Technik, Formeln von innen her komplexer zu machen:

Definition 1.25. Seien φ, ψ Formeln, und sei A eine Variable. Dann ist $\varphi(A\backslash\psi)$ (sprich φ, A ersetzt durch ψ) die Formel, die aus φ entsteht, indem jedes Vorkommen von A durch ψ ersetzt wird.

Wie zeigt man, dass $\varphi(A\backslash\psi)$ tatsächlich ein Formel ist?

Lemma 1.26. *Ersetzungslemma. Sei $\varphi \equiv \varphi'$ und sei A eine Variable und ψ eine Formel. Dann ist $\varphi(A\backslash\psi) \equiv \varphi'(A\backslash\psi)$ und $\psi(A\backslash\varphi) \equiv \psi(A\backslash\varphi')$.*

Lemma 1.27. *$\varphi \vee \neg\varphi$ is allgemeingültig. Diese Regel wird auch* tertium non datur *genannt. Englisch: Excluded middle.*

Satz 1.28. *Es gelten die de Morgan'schen Regeln:*

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

Beweis: Man rechnet man mit Definition 1.13 nach, dass $\bar{v}$, angewandt auf die linke Seite, mit $\bar{v}$, angewandt auf die rechte Seite, für alle Wahrheitsbelegungen v von A und B übereinstimmt $\dashv$

Bemerkung: Wir haben die de Morgan'schen Regeln separat von den Grundäquivalenzen geschrieben, weil sie die Negation beschreiben. Die Grundäquivalenzen beschreiben einen distributiven Verband mit größtem und kleinstem Element. Die Negation gibt die Komplementierung, die aus einem distributiven Verband eine Boole'sche Algebra (siehe Abschnitt 1.6) macht.

Definition 1.29. Ein *Literal* ist eine Variable oder eine negierte Variable.

Definition 1.30. Eine aussagenlogische Formel φ ist in *disjunktiver Normalform gdw es $k \in \mathbb{N}$ und $m_i \in \mathbb{N}$, $i < k$, und Literale $A_{i,j}$, $i < k$, $j < m_i$, gibt, so dass*

$$\varphi = \bigvee_{i<k} \bigwedge_{j<m_i} A_{i,j}.$$

Eine aussagenlogische Formel φ ist in konjunktiver Normalform gdw es $k \in \mathbb{N}$ und $m_i \in \mathbb{N}$, $i < k$, und Literale $A_{i,j}$, $i < k$, $j < m_i$, gibt, so dass

$$\varphi = \bigwedge_{i<k} \bigvee_{j<m_i} A_{i,j}.$$

Welcher Normalformtyp gestattet unmittelbares Ablesen der Erfüllbarkeit?

Satz 1.31. *Jede Formel ist zu einer Formel in konjunktiver Normalform und zu einer Formel in disjunktiver Normalform äquivalent.*

Beweis: Simultan für beide Aussagen, induktiv über den Aufbau der Formel. Da wir beide Normalformen gleichzeitig durch die Induktion hochtragen, ist der $\neg$-Schritt automatisch. Der $\wedge$-Schritt für die konjunktive NF ist einfach. Wir zeigen den $\wedge$-Schritt für die disjunktive NF $(\bigvee_{i<k} \bigwedge_{j<m_i} A_{i,j}) \wedge (\bigvee_{i'<k'} \bigwedge_{j'<m'_{i'}} B_{i',j'}) \equiv \bigvee_{i<k,i'<k'} (\bigwedge_{j<m_i} A_{i,j} \wedge \bigwedge_{j'<m'_{i'}} B_{i',j'})$. Die anderen Schritte zeigt man durch Negieren und Anwendung der de Morgan'schen Regeln und Rückgriff auf die Induktionsvoraussetzung. $\dashv$

Übung: Führen Sie den $\vee$-Schritt für die konjunktive NF durch.

Korollar 1.32. *Jede Äquivalenz lässt sich mit Hilfe des Ersetzungslemmas 1.23 aus den Grundäquivalenzen 1.24 und aus den de Morgan'schen Regeln formal herleiten.*

Beweisskizze: Der Beweis des vorigen Theorems wird aus den Grundaussagen und Ersetzungen in Grundaussagen geführt. Dann darf man jetzt schon verwenden, dass man mit Hilfe des Ersetzungslemmas 1.23 aus den Grundäquivalenzen 1.24 formal hergeleitet hat, dass jede der beiden zu untersuchenden Aussagen in disjunktiver Normalform dasteht. Nun dünnt man mit den Absorptionsgesetzen die Disjunktionsglieder in $\bigvee_{i<k} \bigwedge_{j<m_i} A_{i,j}$ aus, indem man die unerfüllbaren Disjunktionsglieder (bei denen es j, j' gibt mit $A_{i,j} = \neg A_{i,j'}$) und diejenigen, zu denen es noch ein schwächeres Disjunktionsglied gibt, weglässt. Dann ordnet man die Reihenfolge der minimal starken Disjunktionsglieder um und ordnet auch die Konjunktionsglieder innerhalb jedes Disjunktionsglieds beliebig um. Man prüft ob es eine Umordnung der ersten Formel gibt, so dass nach der Anwendung des Idempotenzgesetzes auf beiden Seiten, die erste Formel als Zeichenreihe wie die zweite Formel aussieht. $\dashv$

Natürlich kann man Äquivalenz von φ und ψ auch einfach semantisch prüfen: Man geht alle Belegungen der Variablen in beiden Formel durch, und für jede Belegung v muss gelten $\bar{v}(\varphi) = W$ gdw $\bar{v}(\psi) = W$. Die semantische Prüfung wird uns zum Beweis des Satzes 1.44 dienen.

Wir haben also $\varphi \equiv \psi$ gdw es eine formale Herleitung der Äquivalenz gibt.

Bemerkung 1.33. Die skizzierten Berechnungsverfahren sind NP-hart. (Beweis: Satz von Cook im Kapitel 2.7.) Es ist also unbekannt, ob es ein polynomiales Verfahren zum Feststellen logischer Äquivalenz gibt.

1.5 Kompaktheit und Entscheidbarkeit

In desem wichtigen Abschnitt beweisen wir eine Kompaktheitseigenschaft für die Aussagenlogik und die Entscheidbarkeit der Menge der erfüllbaren Formeln. Um nicht zu viel mengentheoretische Technik zu benötigen, beschränken wir uns auf abzählbar viele Satzsymbole. Die Analoga für größere Symbolmengen werden hier nicht bewiesen

Definition 1.34. Eine Menge X heißt *abzählbar* gdw es eine surjektive Funktion $f\colon \mathbb{N} \to X$ gibt.

Definition 1.35. Eine Menge X heißt *abzählbar'* gdw es eine injektive Funktion $f\colon X \to \mathbb{N}$ gibt.

Lemma 1.36. *Abzählbar und abzählbar' sind äquivalent für nicht leere Mengen.*

Beweis: „$\Rightarrow$": Sei $f\colon \mathbb{N} \to X$ surjektiv. Für $x \in X$ sei $g(x) := \min\{n \mid f(n) = x\}$. Dann ist $g\colon X \to \mathbb{N}$ injektiv, da f eine Funktion ist.

„$\Leftarrow$": Sei $g\colon X \to \mathbb{N}$ injektiv und es sei X nicht leer. Sei $x_0 \in X$. Wir definieren

$$f(n) = \begin{cases} x, & \text{wenn } g(x) = n; \\ x_0, & \text{wenn } n \notin g''X. \end{cases}$$

Dann ist $f\colon \mathbb{N} \to X$ surjektiv. ⊣

Definition 1.37. Seien A und B Mengen.

$$A \times B = \{(a, b) \mid a \in A, b \in B\}.$$

Sei $n = \{0, 1, \dots, n-1\} \in \mathbb{N}$.

$$A^n = \{f \mid f\colon n \to A\} = \{\{(0, f(0)), \dots, (n-1, f(n-1))\} \mid f\colon n \to A\}.$$

Man identifiziert oft den endlichen Graphen $\{(0, f(0)), \dots, (n-1, f(n-1))\}$ mit dem Tupel $(f(0), \dots, f(n-1))$. In diesem Sinne ist also $A \times A = A^2$.

Lemma 1.38. $\bigcup_{n\in\mathbb{N}} \mathbb{N}^n$ *ist abzählbar. Die Menge der endlich langen Zeichenreihen (Wörter) über einem abzählbaren Alphabet ist abzählbar.*

Beweisskizze: Man zeigt jeweils, z. B. durch ein diagonales Abzählverfahren:

(a) Wenn A und B abzählbar sind, dann ist auch $A \times B$ abzählbar.

(b) Wenn für $n \in \mathbb{N}$ die Menge A_n abzählbar ist, dann ist auch $\bigcup_{n\in\mathbb{N}} A_n$ abzählbar.

⊣

Korollar 1.39. *Die Menge der aussagenlogischen Formeln ist abzählbar.*

Eine wichtige Tatsache ist der Kompaktheitssatz der Aussagenlogik.

Definition 1.40. Eine Menge Σ von Formeln ist erfüllbar gdw es eine Wahrheitsbelegung gibt, die jedes Element von Σ erfüllt.

Satz 1.41. Kompaktheitssatz für die Aussagenlogik. *Sei Σ eine abzählbare Menge von Formeln, so dass jede endliche Teilmenge von Σ erfüllbar ist. Dann ist auch Σ erfüllbar.*

Beweis: Nennen wir Σ endlich erfüllbar gdw jede endliche Teilmenge von Σ erfüllbar ist. Unter Verwendung dieser Definition können wir den Kompaktheitssatz wie folgt formulieren: Wenn Σ endlich erfüllbar ist, dann ist Σ erfüllbar. Der Beweis besteht aus zwei Teilen. Im ersten Teil erweitern wir unsere gegebene endlich erfülbare Menge Σ zu einer maximalen endlich erfüllbaren Menge Δ. Im zweiten Schritt verwenden wir Δ, um eine Wahrheitsbelegung zu wählen, die Σ erfüllt.

Wir wählen eine Liste $\alpha_1, \alpha_2, \dots$ der Menge aller Formeln, die mit den natürlichen Zahlen indiziert ist.

Wir definieren nun $\Delta_0 = \Sigma$. Für jede natürliche Zahl n definieren wir nun $\Delta_{n+1} = \Delta_n \cup \{\alpha_{n+1}\}$, wenn dies endlich erfüllbar ist, sonst definieren wir $\Delta_{n+1} = \Delta_n \cup \{\neg\alpha_{n+1}\}$. Sei Δ die Vereinigung der Δ_n.

Wir beweisen nun durch Induktion über n: Jedes Δ_n ist endlich erfüllbar. Die Induktionsaussage gilt für $n = 0$, weil Σ nach Voraussetzung des Satzes endlich erfüllbar ist. Wir nehmen nun an, dass Δ_n endlich erfüllbar sei, und zeigen, dass dann auch Δ_{n+1} endlich erfüllbar ist. Wir müssen zeigen, dass $\Delta_n \cup \{\alpha_{n+1}\}$ oder $\Delta_n \cup \{\neg\alpha_{n+1}\}$ endlich erfüllbar ist. Wenn dies nicht der Fall ist, können wir endliche Teilmengen Σ_0 und Σ_1 von Δ_n wählen, so dass beide, $\Sigma_0 \cup \{\alpha_{n+1}\}$ und $\Sigma_1 \cup \{\neg\alpha_{n+1}\}$, nicht erfüllbar sind. Dann ist aber $\Sigma_0 \cup \Sigma_1 \subseteq \Delta_n$ endlich und nicht erfüllbar, weil jede Wahrheitsbelegung entweder α_{n+1} oder $(\neg\alpha_{n+1})$ erfüllen muss. Das steht im Widerspruch zu unserer Induktionsannahme.

Dann ist auch Δ, die aufsteigende Vereinigung der Δ_n, endlich erfüllbar. Und Δ ist maximal im folgenden Sinn: Für jede Formel α ist $\alpha \in \Delta$ oder $\neg\alpha \in \Delta$.

Nun definieren wir eine Wahrheitsbelegung v wie folgt: Für jedes Satzsymbol A setzen wir $v(A) = W$ gdw $A \in \Delta$. Nach Konstruktion ist Δ außerdem vollständig: $\alpha_i \in \Delta$ oder $\neg\alpha_i \in \Delta$, und $\Sigma \subseteq \Delta$.

Behauptung: Für jede Formel φ gilt: $\bar{v}(\varphi) = W$ gdw $\varphi \in \Delta$.

Wir beweisen die Behauptung durch Induktion über den Aufbau von φ. Wenn φ ein Satzsymbol ist, dann gilt die Behauptung aufgrund der Definition von v und von $\bar{v}$. Nun sei $\varphi = \neg\psi$ und die Behauptung gelte für ψ. Dann ist $\bar{v}(\varphi) = W$ gdw $\bar{v}(\psi) = F$ gdw $\psi \notin \Delta$ wegen der Induktionsannahme. $\psi \notin \Delta$ impliziert aber nun $\neg\psi = \varphi \in \Delta$, da Δ ja maximal ist. Umgekehrt gilt auch, wenn $\varphi \in \Delta$ dann $\psi \notin \Delta$, da Δ ja endlich erfüllbar ist. Dehalb haben wir $\bar{v}(\varphi) = W$ gdw $\varphi \in \Delta$, wie gewünscht. Die anderen Fälle, in denen φ von der Form $(\psi \wedge \chi)$, $(\psi \vee \chi)$, $(\psi \to \chi)$, $(\psi \leftrightarrow \chi)$ ist, sind ähnlich. Wenn Sie die Aufgabe Ü2 über die Junktoren gemacht haben, können Sie sich auf $\vee$ oder $\wedge$ oder $\to$ alleine beschränken (nicht jedoch auf $\leftrightarrow$, wie Ü3 zeigt).

Nun haben wir, dass v eine Wahrheitsbelegung ist, die jede Formel von Δ erfüllt und daher erst recht jede Formel von $\Sigma \subseteq \Delta$ erfüllt. Also ist Σ wie gewünscht erfüllbar. $\dashv$

Korollar 1.42. *Wenn $\Sigma \models \tau$, dann gibt es ein endliches $\Sigma_0 \subseteq \Sigma$, so dass $\Sigma_0 \models \tau$.*

Beweis: Wenn es keine endliche Teilmenge Σ_0 von Σ gibt, die φ impliziert, dann ist $\Sigma \cup \{\neg\varphi\}$ endlich erfüllbar, und daher nach dem Kompakheitssatz erfüllbar. ⊣

Ü4. Nun eine Aufgabe mit etwas Topologie: Zeigen Sie, dass die Anzahl der logisch nicht äquivalenten Vervollständigungen einer konsistenten Menge C aussagenlogischer Formeln entweder endlich oder 2^ω (dies ist die Mächtigkeit des Kontinuums) ist.

Hinweis: Wir führen eine Topologie τ ein auf der Menge aller Vervollständigungen. Eine Basis $\mathcal{B}$ für die Menge τ der offenen Mengen ist gegeben durch

$$\begin{aligned}\mathcal{B} &= \{[\varphi] \mid \varphi \text{ Formel}\},\\ [\varphi] &= \{\Sigma \mid \varphi \in \Sigma\}.\end{aligned}$$

Dies ist eine Basis aus clopen (abgeschlossenen und offenen) Mengen. Nach dem Kompaktheitssatz ist der Raum

$$(\{\Sigma \mid \Sigma \text{ Vervollständigung von } C\}, \tau)$$

kompakt. Kompakte Räume mit Basen aus clopen Mengen heißen auch kompakte nulldimensionale Räume, sind entweder endlich oder betten die Äste eines perfekten Baumes (ein perfekter Baum ist ein abzählbar unendlich hoher binärer Baum, auch $2^{<\omega}$ genannt) ein [3].

Man könnte natürlich Def. 1.1 dahingegehend ändern, dass man überabzählbar viele Variablen zuässt, z.B., A_r, $r \in \mathbb{R}$.

Satz 1.43. *Der Kompaktheitssatz gilt auch für überabzählbare Mengen aussagenlogischer Formeln.*

Beweis durch transfinite Induktion. Dies ist eine Technik aus der Mengenlehre.

Satz 1.44. *Die Menge der aussagenlogischen Tautologien ist entscheidbar.*

Beweis: Für eine gegebene Formel gibt es einen Algorithmus oder eine effektive Prozedur, die in endlicher Zeit entscheidet, ob φ eine Tautologie ist oder nicht. Ein mögliches Verfahren sieht wie folgt aus: Wir erstellen eine Liste aller Möglichkeiten, den in φ auftretenden Satzsymbolen Wahrheitswerte zuzuordnen. Für jede dieser Möglichkeiten v berechnen wir den sich ergebenden Wahrheitwert von φ, $\bar{v}(\varphi)$. Wenn dieser Wahrheitswert für alle v W ist, dann ist φ eine Tautologie, sonst nicht. Ein weiteres Verfahren kann man aus einen Beweis der Korollars 1.32 herleiten. ⊣

Wir werden im Kapitel über Komplexitätstheorie lernen, von welcher seitlichen Komplexität der gerade geschilderte Algorithmus ist.

1.6 Boole'sche Algebren

Die Strukturklasse der Boole'schen Algebren ist eng mit der Aussagenlogik verwandt.

Definition 1.45. Eine *Boole'sche Algebra* $(B, 0, 1, \sqcap, \sqcup, {}^c)$ ist eine Menge B mit zwei ausgezeichneten Elementen 0 und 1 und Operationen $\sqcap, \sqcup \colon B \times B \to B$ und ${}^c \colon B \to B$, für die folgenden Gleichungen gelten:

$$
\begin{array}{rclrcll}
a \sqcap a & = & a & a \sqcup a & = & a & \text{(1.1) Idempotenz} \\
a \sqcap b & = & b \sqcap a & a \sqcup b & = & b \sqcup a & \text{(1.2) Kommutativität} \\
(a \sqcap b) \sqcap c & = & a \sqcap (b \sqcap c) & (a \sqcup b) \sqcup c & = & a \sqcup (b \sqcup c) & \text{(1.3) Assoziativität} \\
a \sqcap (a \sqcup b) & = & a & a \sqcup (a \sqcap b) & = & a & \text{(1.4) Absorption} \\
a \sqcap (b \sqcup c) & = & (a \sqcap b) \sqcup (a \sqcap c) & a \sqcup (b \sqcap c) & = & (a \sqcup b) \sqcap (a \sqcup c) & \text{(1.5) Distributivität} \\
0 \sqcap a & = & 0 & 1 \sqcup a & = & 1 & \text{(1.6) Extrema} \\
a \sqcap a^c & = & 0 & a \sqcup a^c & = & 1 & \text{(1.7) Komplementierung}
\end{array}
$$

Die Axiome entsprechen den grundlegenden Äquivalenzen von Satz 1.24. Das zweite Distributivitätsaxiom ist überflüssig. In Boole'schen Algebren gelten die de Morgan'schen Regeln. Eine Struktur $(V, \sqcup, \sqcap)$, in der (1.1), (1.2), (1.3) und (1.4) gelten, ist ein *Verband*. Wenn $(P, \leq)$ eine partielle Ordnung ist, in der je zwei Elemente ein Supremum (oder Maximum) und ein Infimum (oder Minumum) haben, ist $(P, \inf, \sup)$ ein Verband. Jeder Verband hat diese Gestalt, wenn man $a \leq b$ durch $a \sqcup b = a$ oder, äquivalent, durch $a \sqcap b = b$ definiert. Die Gleichungen (1.6) bedeuten, dass 0 und 1 kleinstes und größtes Element sind; (1.7) drückt aus, dass a^c ein Komplement von a ist. In einem distributiven Verband, d.h. in einem Verband, in dem (1.5) gilt, gibt es immer nur höchstens ein Komplement. Eine Boole'sche Algebra ist also ein „komplementärer distributiver Verband".

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

Satz 1.46 (Marshall Harvey Stone, 1936)**.** *Der Stone'sche Darstellungssatz für Boole'sche Algebren. Jede endliche Boole'sche Algebra ist isomorph zu einer Potenzmengenalgebra. Jede unendliche Boole'sche Algebra ist isomorph zu einer Unteralgebra einer Potenzmengenalgebra.*

Beweis: Sei B eine endliche Boole'sche Algebra. Wie definieren: $x \in B$ ist ein *Atom*, genau dann wenn $(\forall y \leq x)(y = x \vee y = 0)$. Sei $A \subseteq B$ die Menge der Atome von B. Die Abbildung $\varphi \colon (B, \sqcup, \sqcap, {}^c, 0, 1) \to (\mathcal{P}(A), \cup, \cap, {}^c, \emptyset, A)$ mit $\varphi(b) = \{x \in A \mid x \leq b\}$ ist ein Isomorphismus der beiden Boole'schen Algebren, d.h. φ ist treu bezüglich aller Funktionen und Konstanten und bijektiv.

Für den allgemeinen Fall definieren wir:

(1) $u \sqsubseteq_B v$ gdw $u \sqcup_B v = u$.

(2) $U \subseteq B$ ist ein *Ultrafilter auf* B gdw U gegen $\sqcap_B$ abgeschlossen ist und $(\forall u \in U)(\forall v \in B)(u \sqsubseteq_B v \to v \in U)$ und $0 \notin U$ ist und U $\subseteq$-maximal mit diesen Eigenschaften ist.

Es sei $S(B) = \{U \mid U \text{ Ultrafilter auf } B\}$, der sogenannte *Stoneraum* von B. Wir definieren $\varphi\colon B \to \mathcal{P}(S(B))$ durch $\varphi(b) = \{U \in S(B) \mid b \in U\}$. Man rechnet wieder nach, dass φ eine Einbettung von $(B, 0, 1, \sqcup, \sqcap, {}^c)$ nach

$$(\mathcal{P}(S(B)), \emptyset, S(B), \cup, \cap, X \mapsto S(B) \setminus X)$$

ist (zu einem Isomorphismus fehlt nur die Surjektivität, und φ ist nicht surjektiv). $\dashv$

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

$$\begin{array}{rcl} 1 & = & \top/\equiv \\ 0 & = & \bot/\equiv \\ (\varphi/\equiv) \sqcap (\psi/\equiv) & = & (\varphi \wedge \psi)/\equiv \\ (\varphi/\equiv) \sqcup (\psi/\equiv) & = & (\varphi \vee \psi)/\equiv \\ (\varphi/\equiv)^c & = & \neg\varphi/\equiv, \end{array}$$

wird $(LA_n, \sqcup, \sqcap, {}^c, \top, \bot)$ zu einer Boole'schen Algebra. LA_n is isomomorph zur Potenzmengenalgebra einer Menge mit 2^n Elementen, hat also 2^{2^n} Elemente. LA_n wird von den $A_n/\equiv$ frei erzeugt, d.h. jede aus den A_i gebildeten erfüllbare Formel φ genügt in der Lindenbaumalgebra der Gleichung $(\varphi/\equiv) \neq 0$.

Kapitel 2

Komplexitätstheorie

2.1 Turingmaschinen

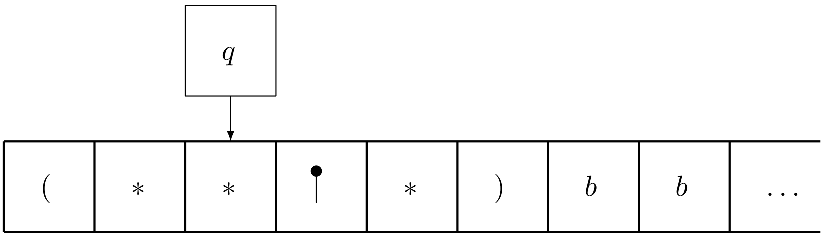
Das Gebiet „Rekursionstheorie", seit einigen Jahren auch „Berechenbarkeitstheorie" genannt, untersucht die prinzipielle Berechenbarkeit von Problemen aus der Informatik. Zum Beispiel nennen wir das Problem der Hamiltonpfade: Welche endlichen Graphen haben einen Pfad, der jeden Vertex genau einmal berührt? Ein Problem in diesen Sinne ist (nach einer geeigneten Kodierung, in unserem Beispiel kann man die Graphen als endliche Matrizen über $\mathbb{N}$ kodieren und diese Matrizen wiederum als naürliche Zahlen) eine Teilmenge von $\mathbb{N}$. Da es nur abzählbar viele Programme gibt, zeigt ein Abzählbarkeitsargument, dass es nicht berechenbare Teilmengen von $\mathbb{N}$ gibt.

Wenn man schon weiß, dass $A \subseteq \mathbb{N}$ berechenbar ist, kann man weiter fragen: Wieviel Zeit braucht ein Algorithmus, um die Frage „Ist $n \in A$?" zu beantworten? Fragen dieser Art gehören in das Gebiet „Komplexitätstheorie".

Es gibt viele Modelle, Algorithmen exakt zu fassen: C^{++}, Perlscripts, Ihre Lieblingsprogrammiersprache, Turing-Maschinen, Registermaschinen, Flussdiagramme. Alle diese Modelle werden hier jeweils auf Maschinen mit unbegrenztem Speicherplatz idealisiert. Mit viel Geduld und schrittweiser Übersetzungsarbeit kann man zeigen, dass alle gängigen Berechnungsmodelle gleich stark sind: Was unter einem Modell berechenbar ist, bleibt in jedem anderen Modell berechenbar. Dies ist die *Church'sche These*, dass es genau einen Begriff „rekursiv" (auch berechenbar, effektiv oder effektiv berechenbar oder entscheidbar genannt) gibt.

Und mit noch geduldigerer Übersetzungsarbeit kann man zeigen, dass auch die Zahl der Berechnungsschritte bis auf konstante Faktoren und kleine polynomiale Verzerrungen recht unabhängig vom gewählten (deterministischen oder nicht determinsistischen) Berechnungsmodell ist. Wir verlieren also nicht nicht viel, wenn wir uns zunächst auf das Modell der Turing-Berechnungen beschränken. Bei diesen betrachten wir deterministische und nicht deterministische Berechnungsmodelle. Einige der wichtigsten Komplexitätsklassen werden mit Hilfe der Zeitkomplexität nicht deterministischer Algorithmen definiert.

Bildlich gesprochen hat eine Turingmaschine ein nach rechts unendlich langes Band und einen Lesekopf, der auf das Band Symbole schreiben und sich nach links und nach rechts bewegen kann. Am Anfang enthält das Band eine endliche Inputfolge als Inschrift, die das Zeichen „leer" (wir nehmen b als Zeichen für „leer") nicht enthält, und ist sonst leer. Die Maschine rechnet, bis sie einen Output q_{ak} („akzeptieren", „ja") oder q_{ab} („ablehnen", „nein") liefert, oder rechnet unendlich lange, ohne zu stoppen.



Definition 2.1. Eine Turingmaschine (TM) ist ein 7-Tupel $M = (Q, \Sigma, \Gamma, \delta,$ $q_0, q_{ak}, q_{ab})$, das folgende Bedingungen erfüllt:

1. Q ist die endliche Menge der *Zustände*.
2. Σ ist das endliche *Eingabe-Alphabet*, das das Symbol b nicht enthält.
3. Γ ist das *Band-Alphabet*, das b enthält und Σ als Teilmenge hat.
4. $\delta\colon (Q \setminus \{q_{\mathrm{al}}, q_{\mathrm{ab}}\}) \times \Gamma \to Q \times \Gamma \times \{L, R\}$ ist die *Übergangsfunktion*, auch *Turingtafel* genannt.
5. $q_0 \in Q$ ist der Anfangszustand.
6. $q_{ak} \in Q$ ist der Akzeptierungszustand.
7. $q_{ab} \in Q$, $q_{ab} \neq q_{ak}$, ist der Ablehnungszustand.

Definition 2.2. Der sogenannte *Kleene-Stern* Sei Σ eine Menge. Dann ist

$$\Sigma^* = \{(a_0, \dots, a_{n-1}) \,|\, n \in \mathbb{N}, a_i \in \Sigma\}$$

die Menge der endlichen Tupel aus Σ, die auch *Wörter über* Σ genannt werden. Wir schreiben $\Box$ für das leere Wort. Statt $(a_0, \dots, a_{n-1})$ schreibt man auch $a_0 \dots a_{n-1}$. Die Konkatenation von Wörtern oder von einem Wort und einem Buchstaben wird einfach durch Hintereinanderschreiben ohne Trennungszeichen geschrieben.

Wir lassen üblicherweise Kommata und Klammern weg, wenn wir Wörter schreiben $(a_0, \dots, a_{n-1}) = a_0 \dots a_{n-1}$. Dies setzt natürlich die Wohlunterscheidbarkeit der Buchstaben voraus.

Eine Turingmaschine M rechnet anschaulich gesprochen wie folgt: Ein Input ist von der Form $w = w_1 w_2 \dots w_n \in \Sigma^*$. Am Anfang der Berechnung ist w auf die ersten n Zellen des Bandes geschrieben und der Rest des Bandes ist unbeschriftet. Der Lesekopf beginnt auf der ersten Zelle des Bandes. Dann verläuft die Berechnung nach der Übergangsfunktion wie in Definition 2.3 bis 2.5 beschrieben. Auf der ersten Zelle des Bandes ist die Bewegung L des Lesekopfes nicht erlaubt, er bleibt dann nur stehen. Die Berechnung fährt fort, bis M entweder in den Zustand q_{ab} oder q_{ak} eintritt. Genau in diesen beiden Zuständen hält M. Sonst fährt M auf immer fort.

Definition 2.3. Eine *Konfiguration* von M ist ein Tripel $(u, q, v) \in \Gamma^* \times Q \times \Gamma^*$.

Die Bedeutung ist: M ist im Zustand q, das Wort u steht links vom Lesekopf auf dem Band und und rechts davon steht das Wort v, der Lesekopf steht auf dem ersten Symbol von v. Wir verwenden hier a, b', c für Buchstaben und u, v für Wörter.

Definition 2.4. Seien u, v Wörter, und a, b', c, d Buchstaben. Eine Konfiguration $(ua, (q, b'), dv)$ hat als *Nachfolgerkonfiguration*

$$\begin{array}{lll} (u, (q', a), cdv) & \text{gdw} & \delta(q, b') = (q', c, L), \text{ und} \\ (uac, (q', d), v) & \text{gdw} & \delta(q, b') = (q', c, R). \end{array}$$

Das Band hat einen linken Rand, ist also wie $\mathbb{N}$ geordnet. Dies drückt sich in folgender Regel aus: Eine Konfiguration $(\square, (qb'), dv)$ hat als Nachfolgerkonfiguration

$$\begin{array}{rll} (\square, (q', c), dv) & \text{gdw} & \delta(q, b') = (q', c, L), \text{ und} \\ (c, (q', d), v) & \text{gdw} & \delta(q, b') = (q', c, R). \end{array}$$

Nun folgt eine sehr wichtige dreiteilige Definition:

Definition 2.5. (1) M *akzeptiert* den Input w gdw es ein $k \in \mathbb{N}$ und eine Folge von Konfigurationen C_0, C_1, …, C_k gibt, so dass die folgenden Bedingungen erfüllt werden:

1. $C_0 = (\square, q_0, w)$ ist die *Anfangskonfiguration* $q_0 w$ *von* M.
2. C_{i+1} ist die *Nachfolgerkonfiguration* von C_i für alle i.
3. C_k ist die Konfiguration mit dem Zustand q_{ak}.

(2) Sei M eine TM, die angesetzt auf jeglichen Input, immer nach endlich vielen Schritten hält. (Für Kenner: Wenn man diese Forderung weglässt, dann werden auch rekursiv aufzählbare Mengen Akzeptierungsmengen. Aber wir möchten hier, dass alle Akzeptierungsmengen rekursiv entscheidbar sind). Die Menge B der Folgen aus Σ, die M akzeptiert, wird die Akzeptierungsmenge $B = A(M)$ genannt. Wir sagen auch M akzeptiert B, wenn $B = A(M)$.

(3) Ein Menge $B \subseteq \Sigma^*$ heißt *Turing-berechenbar* gdw es eine auf jedem Input nach endlich vielen Schritten stoppende Turingmaschine M gibt, so dass $B = A(M)$.

(4) Eine Funktion $F\colon \Sigma^* \to \Sigma^*$ heißt *Turing-berechenbar* gdw es eine Turingmaschine gibt, die für jedes $w \in \Sigma^*$ mit der Anfangskonfiguration (q_0, w) startend nach endlich Schritten akzeptierend hält und im Haltezustand die Konfiguration $(q_{\mathrm{ak}}, F(w))$ hat.

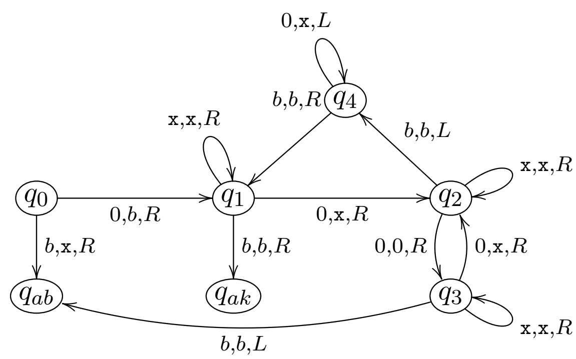
Nach der Church'schen These kann man diese exakte Definition nun als Definition von „berechenbar" nehmen. Wir werden das von nun an tun.

2.2 Einige Beispiele von Turingmaschinen

Unsere Beispiele werden nicht genau in den eben beschriebenen Rahmen eingepasst, sondern eher intuitiv beschrieben.

Beispiel 1: Ein M mit $A(M) = \{0^{2^n} \mid n \in \mathbb{N}\}$.
Sei ein Input w gegeben. Sei $\Sigma = \{0\}$, $\Gamma = \{0, \mathtt{x}, b\}$.
1. Gehe von nach rechts das Band entlang und setze jede zweite 0 durch x.
2. Wenn die Anzahl der Nullen ungerade und größer 1 ist, lehne w ab.
3. Wenn das Band eine einzige 0 enthält, dann akzeptiere den Input w.
4. Gehe zum Anfang des Bandes zurück.
5. Wiederhole Schritt 1.

Zustände $\{q_0, q_{ak}, q_{ab}, q_1, q_2, q_3, q_4\}$. Dank xy-pictures haben wir nun folgende Skizze



Die Skizze ist wie folgt zu lesen: Für $a \in \Sigma$ und $X \in \{L, R\}$ gilt: Ein Pfeil von q_i nach q_j mit Beschriftung abX steht für $\delta(q_i, a) = (q_j, b, X)$. Wir erhalten also die Übergangsfunktion δ von M:

q_0: Anfangszustand:
$\delta(q_0, 0) = (q_1, b, R)$
$\delta(q_0, b) = (q_{ab}, b, R)$
$\delta(q_0, \mathtt{x}) = (q_{ab}, \mathtt{x}, R)$

q_1: am Anfang des aktuellen Worts (keine b gerechnet), prüfen, ob noch andere Buchstaben außer x auf dem Band stehen, wenn keine anderen Buchstaben gefunden, akzeptieren:
$\delta(q_1, 0) = (q_2, \mathtt{x}, R)$
$\delta(q_1, b) = (q_{ak}, b, R)$
$\delta(q_1, \mathtt{x}) = (q_1, \mathtt{x}, R)$

q_2: vorher gerade Zahl an Nullen-Zustand, jetzt bei ungerader Null, falls jetzt eine Null da steht, aber nicht auf dem linkesten Platz, so bleibt diese 0:
$\delta(q_2, 0) = (q_3, 0, R)$
$\delta(q_2, b) = (q_4, b, L)$
$\delta(q_2, \mathtt{x}) = (q_2, \mathtt{x}, R)$

q_3: vorher ungerade Zahl an Nullen-Zustand, jetzt bei gerader Null, falls jetzt eine Null da steht, aber nicht am linkesten Platz, so wird die 0 gestrichen, q_3 wechselt sich mit q_2 ab:
$\delta(q_3, 0) = (q_2, \mathtt{x}, R)$
$\delta(q_3, b) = (q_{ab}, b, R)$
$\delta(q_3, \mathtt{x}) = (q_3, \mathtt{x}, R)$

q_4: Chance auf Akzeptieren noch nicht verworfen, gehe zurück nach links, bis zu b (das ja in der ersten Zelle steht):
$\delta(q_4, 0) = (q_4, 0, L)$
$\delta(q_4, b) = (q_1, b, R)$
$\delta(q_4, \mathtt{x}) = (q_4, \mathtt{x}, L)$

Jedes Mal, wenn M Schritt 1 wiederholt, dividiert M die Anzahl der Nullen durch 2. Die Maschine sieht, ob die Anzahl der Nullen gerade oder 1 ist, wenn nicht, lehnt sie ab. Wenn M genau eine Null sieht, dann akzeptiert M den Input.

Beispiel 2: Ein M mit $A(M) = \{w\#w \mid w \in \{0,1\}^*\}$.
Sei ein Input w gegeben.
1. M schaut sich w an, um zu überprüfen, ob w genau ein # Symbol enthält. Wenn nicht, lehnt M w ab.
2. M geht auf dem Band hin und her und überprüft, ob entprechende Zellen zu beiden Seiten von # dasselbe Symbol enthalten. Wenn nicht, lehnt M w ab. M löscht Symbole, nachdem sie überprüft wurden, um in Auge zu behalten, welche Symbole einander entsprechen. Vorschlag: $Q \supset \Sigma \times Q'$, um das Gelesene als Zustand im Gedächtnis zu behalten.
3. Wenn alle Symbole zur linken Seite von # gelöscht sind, überprüft M, ob kein Symbol auf der rechten Seite von # übrig bleibt. Wenn ja, abzeptiert M w.
Bemerkung: Ein Übergangsdiagramm findet sich im Buch von Sipser in Kapitel 3.1.

Beispiel 3: Ein M mit $A(M) = \{a^i b^j c^k \mid i \cdot j = k \text{ und } i, j, k > 0\}$.
Sei ein Input w gegeben.
1. M schaut sich w an, um zu überprüfen, ob w von der Form $a^i b^j c^k$ ist. Wenn nicht, lehnt M w ab.
2. Der Lesekopf geht zum Anfang des Bandes.
3. M löscht ein a, und sein Lesekopf geht nach rechts, bis ein b auftritt. Dann löscht M ein b und ein c, abwechselnd, bis kein b mehr übrigbleibt.
4. M schreibt die b wieder hin, und wiederholt Schritt 3. Wenn kein a mehr da ist, überprüft M on kein c mehr da ist. Wenn ja, akzeptiert M w, und im anderen Fall lehnt M w ab.

Beispiel 4: Ein M mit $A(M) = \{\#x_1\#x_2 \dots \#x_\ell \mid \forall i \neq j (x_i \in \{0,1\}^* \wedge x_i \neq x_j)\}$.
Angesetzt auf einen Input w, arbeitet M mit zwei Marken, d.h. zwei neuen Zeichen, die auch beide dasselbe Zeichen sein köennen. Ein markiertes # ist

eine Zelle, in der eine Marke (statt #) steht. Eine Marke bewegen heißt, dass an ihre alte Stelle wieder # geschrieben wird.

1. M schreibt eine Marke auf das erste Bandsymbol. Wenn das Symbol b ist, akzeptiert M w. Wenn das Symbol kein # ist, lehnt M w ab. Sonst geht M zum nächsten Schritt über.

2. M geht nach rechts zum nächsten # und schreibt eine Marke darauf. Wenn es kein zweites # gibt, abzepiert M w. Sonst gibt es nun zwei Marken, die linke und die rechte genannt.

3. M vergeicht die zwei Folgen auf den rechten Seiten der markierten #en. Wenn sie übereinstimmen, lehnt M w ab.

4. M bewegt die rechte Marke zum nächsten #-Symbol. Wenn die rechte Marke schon vor einem b steht, bewegt M die linke Marke zum ihr nächstfolgenden #-Symbol und holt die rechte Marke zurück auf das der linken Marke nächstfolgende #-Symbol. Wenn es kein solches nächstfolgendes # gibt, auf das die rechte Marke zurückgeholt werden kann, dann akzeptiert M w.

5. M wiederholt Schritt 3.

Übung Könnte man die Beispiele auch programmieren, wenn die Turingmaschine nicht schreiben darf? Welche Mengen kann eine Turingmaschine, die nicht schreiben darf, akzeptieren? Ungeduldige Leserinnen und Leser finden die Anwort mit „google" unter „Schleifenlemma" oder "Pumping Lemma".

2.3 Mehrbandturingmaschinen

Es gibt mehrere Varianten zu Turingmaschinen, die sich jedoch in in der Klasse der akzepierten Mengen nicht unterscheiden. Eine Mehrbandturingmaschine ist eine TM mit mehreren Bändern. Jedes Band hat einen eigenen Lesekopf. Am Anfang ist der Input auf Band 1 geschrieben und die anderen Bänder sind leer. Die Übergangsfunktion beschreibt die gleichzeitige Arbeit aller k Leseköpfe:

$$\delta\colon Q \times \Gamma^k \to Q \times \Gamma^k \times \{L, R\}^k,$$

wenn k die Anzahl der Bänder ist. Der Ausdruck

$$\delta(q_i, a_1, \dots, a_k) = (q_j, b_1, \dots, b_k, L, R, \dots L)$$

bedeutet: Falls die TM im Zustand q_i die Einträge $(a_1, \dots, a_k)$ liest, dann schreiben ihre Leseköpfe die Symbole $(b_1, \dots, b_k)$ in diese Zellen, und bewegen sich wie beschrieben nach rechts oder links und die Maschine geht in den Zustand q_j. Wir beschränken uns hier wieder auf Turingmaschinen, die auf jedem Berechnungsast nach endlich vielen Schritten stoppen.

Falls der Lesekopf auf dem ersten Feld des Bandes steht, bleibt er beim Befehl L einfach auf demselben Feld.

Ähnliches für die Bewegungen des Lesekopfes nach rechts. Die Nachfolgerkonfiguration kann länger sein als ihr Vorgänger, wenn der Lesekopf nach rechts in die Region, wo bis jetzt nur b's stehen, läuft.

Satz 2.6. *Zu jeder Mehrband-TM gibt es eine Einband-TM, die dieselbe Menge akzeptiert.*

Der Beweis wird durch Programmieren einer Turingtafel geführt. Siehe [8, Theorem 3.8]. Um den Kombinationen der Bewegungen der verschiedenen Leseköpfe Herr zu werden, wird der Lesekopf nach links bewegt, nur wenn alle Leseköpfe sich nach links bewegen. Sonst wird der Lesekopf nach rechts bewegt, und die Bänder, auf denen sich die jeweiligen Einzelleseköpfe nach links bewegen sollten, werden statt dessen um zwei Felder nach rechts gerückt.

Der Entscheidbarkeitsbegriff hängt also nicht von der Anzahl der Bänder ab, obwohl man mit höhere Bänderzahl vielleicht "eleganter" und "schneller" rechnen kann. Die Schnelligkeit werden wir mathematisch durch den Begriff "Zeitkomplexität" erfassen.

2.4 Nicht deterministische Turingmschinen

Bei nicht-deterministischen Maschinen ist der Zielbereich der Übergangsfunktion δ nun die Potenzmenge $\mathcal{P}$ des früheren Zielbereichs:

$$\delta\colon Q \times \Gamma \to \mathcal{P}(Q \times \Gamma \times \{L, R\}).$$

Die Interpretation einer Berechnung ist nun ein Baum, der sich von jeder Konfiguration aus genau in alle Konfigurationen, die durch Punkte in der Menge δ beschrieben werden, verzweigt. Wenn ein Ast dieses Baumes zum Akzeptierungszustand führt, dann akzeptiert die Maschine den Input.

Da obige Definition in Worten recht knapp ist, geben wir hier noch eine ausführliche Definitionen des nicht deterministischen Berechnens:

Definition 2.7. Eine nichdeterministische *Turingmaschine* ist ein 9-Tupel

$$M = (Q, \Sigma, \Gamma, b, q_0, q_{\mathrm{ab}}, S, F, \delta)$$

dessen Komponenten die folgenden Bedeutungen haben:

- Q: ist die nicht leere endliche Menge der Zustände,
- Γ: ist die nicht leere endliche Menge der Bandsymbole,
- Σ: $\Sigma \subseteq \Gamma$ ist die Menge der Eingabesymbole,
- b: das Leersymbol, $b \in \Gamma \setminus \Sigma$. Im Anfangszustand stehen in allen bis auf endlich vielen Feldern Leerzeichen.
- q_0: $\in Q$ ist der Anfangszustand,
- q_{ab}: $\in Q$ ist der Ablehnungszustand,
- S: $S \subseteq Q$, die Menge der Stoppzustände,
- F: $F = S \setminus \{q_{\mathrm{ab}}\}$ die Menge der akzeptierenden Zustände,
- δ: die Übergangsfunktion $\delta\colon (Q \setminus S) \times \Gamma \to \mathcal{P}(Q \times \Gamma \times \{R, L\})$. $(p, Y, D) \in \delta(q, X)$ bedeutet: im Zustand q wird X gelesen und nun wird zufällig in einen Nachfolgetripel aus der gegebenen Menge gewechselt. Wenn dieses gerade (p, Y, D) ist, so wird X durch Y überschrieben, der neue Zustand ist p, und der Lesekopf geht in Richtung D.

Wir betrachten nur nichtdeterministische Turingmaschinen, die längs jedes Berechnungsastes stoppen. (Für ExpertInnen: Andernfalls kommen wir wieder zu den rekursiv aufzählbaren Akzeptierungsmengen.)

Definition 2.8. Die Fortsetzung von δ auf Konfigurationen:

Konfigurationsbeschreibungen (IDs- instantaneous descriptions). Eine Konfiguration beschreibt die Stellung des Lesekopfes, den Zustand q und die Bandinschrift vom linken Rand bis zur Stellung des Lesekopfes oder bis zur ersten Stelle, rechts von der nur noch b's stehen, je nachdem, welches weiter rechts liegt. Wir schreiben Konfiguationen als Tripel: Zuerst die Bandinschrift vor dem Lesekopf, dann (q, X) auf das Feld, auf dem der Lesekopf steht, wenn dort der Buchstabe X steht, danach als dritte Komponente der Rest der schon einmal berührten Bandinschrift. So braucht man keine extra Koordinate für die Stellung des Lesekopfes. Es sei $w = a_0 a_1 \dots a_{k-1}$ das Eingabewort. Dann ist die *Anfangskonfiguration* C_0 *von* M *angesetzt auf* w folgendes 3-Tupel:

$$C_0 = (\Box, (q_0, a_0), (a_1, \dots a_{k-1})).$$

Wir schreiben $q(C) = q$, falls $C = ((a_0, a_1, a_2, \dots, a_{i-1}, a_i), (q, a_{i+1}), (a_{i+2}, \dots, a_{n-1}))$. Die Konfiguration

$$C = ((a_0, a_1, a_2, \dots, a_{i-1}, a_i), (q, a_{i+1}), (a_{i+2}, \dots, a_{n-1})$$

hat die Menge der (direkten) Nachfolgerkonfigurationen

$$\begin{aligned}
&\hat{\delta}((a_0, a_1, a_2, \dots, a_{i-1}), (q, a_i), (a_{i+2}, \dots, a_{n-1})) \\
&\quad = \{((a_0, a_1, a_2, \dots, a_{i-2}), (q', a_{i-1}), (a'_i, \dots, a_{n-1})) \mid (q', a'_i, L) \in \delta(q, a_i)\} \cup \\
&\qquad\qquad \{(a_0, a_1, a_2, \dots, a'_i, (q', a_{i+1}), \dots, a_{n-1}) \mid (q', a'_i, R) \in \delta(q, a_i)\}.
\end{aligned}$$

Die Nachfolgermenge einer Konfigurationenmenge ist die Vereinigung der Mengen der Nachfolgerkonfigurationen. Wir schreiben $\hat{\delta}$ auch für diese auf die Potenzmenge des Konfigurationenraums geliftete Funktion, also $\hat{\delta}(\mathscr{C}) = \{\hat{\delta}(C) \mid C \in \mathscr{C}\}$. Für $C' \in \hat{\delta}^{(n)}(\{C\})$ (n Iterationen, für ein beliebiges n) schreibt man $C \vdash C'$.

Eine *Berechnungsast* ist eine endliche Folge $C_0, \dots, C_n$ von Konfigurationen mit $C_{i+1} \in \hat{\delta}^{(n)}(\{C_i\})$ für $i < n$.

Eine Berechnung ist die Bestimmung des endlichen Baums, der durch alle Berechnungsäste gegeben ist.

Die Berechnung beginnt mit der Einermenge der Anfangskonfiguration.

Definition 2.9. Die Menge der akzeptierten Wörter ist

$$\begin{aligned}
A(M) = \{&w \in \Sigma^* \mid \text{ es gibt ein } n \in \mathbb{N}, \text{ es gibt } C_0, \dots, C_n, \\
&C_0 \text{ ist der Anfangskonfiguration}, C_i \vdash C_{i+1}, q(C_n) \in F\}
\end{aligned}$$

Satz 2.10. *Zu jeder nicht deterministischen TM gibt es eine deterministische TM, die dieselbe Menge akzeptiert.*

Beweis [8, Theorem 3.10]. Für Turingmaschinen, die längs jedes Berechnungsastes stoppen (dies trifft nach unserer Definition immer zu, doch viele Autoren nehmen eine schwächere Definition), kann man die deterministische Abarbeitung aller Äste der nicht deterministischen Berechnung nach Belieben anordnen. ⊣

Bemerkung. Wir geben einen Ausblick auf Turingmaschinen in anderer Interpretation, die nicht immer halten müssen. Dann erhält man gerade die rekursiv aufzählbaren Mengen als Akzeptierungsmengen. Auch in dieser Situation können die nicht deterministischen Maschinen durch deterministische simuliert werden. Die deterministische Simulation muss "breadth first" durchgeführt werden und resultiert dann in einer deterministischen TM, die nicht notwendigerweise stoppt. Dies heißt, dass in jedem Block von Simulationsschritten alle Äste der nicht deterministischen Berechnung gleichzeitig um einen Schritt länger simuliert werden. Man simuliert also den Niveaus nach (der Breite nach) durch einen gedachten Suchbaum: Jedes Niveau wird vor dem Aufstieg zu einem höheren Knoten im Baum abgearbeitet. Da jedes Niveau endlich ist, geht das.

Korollar 2.11. *Sei $A \subseteq \Sigma^*$. Die folgenden Aussagen sind äquivalent:*

1. *A ist von der Form $A(M)$ für eine TM M, also Turing-berechenbar.*
2. *A ist von der Form $A(M)$ für eine Mehrband-TM M.*
3. *A ist von der Form $A(M)$ für eine nicht-deterministische TM M.*

2.5 Zeitkomplexität

Innerhalb der berechenbaren Mengen treffen wir nun Einteilungen nach der Komplexität der berechnenden Algorithmen. Die Zeitkomplexität bezieht sich auf die Zahl der Rechenschritte. Es gibt auch noch die Raum-Komplexität, die die Anzahl der gebrauchten Speicherplätze misst.

Definition 2.12. Sei M eine deterministische oder nichtdeterministische TM. Die *Zeitkomplexität* von M ist die Funktion $f\colon \mathbb{N} \to \mathbb{N}$, bei der $f(n)$ das Maximum der Schrittzahl von M für einen Input der Länge n ist. In nicht deterministischen Fall wird zusätzlich noch das Maximum über alle Berechnungspfade gebildet Ein Schritt ist ein Übergang von eine Konfiguration zu ihrer Nachfolgerkonfiguration. Wir sagen „M läuft in Zeit f" oder „M ist ein f-Zeit-TM" (statt f-Zeit auch $f(n)$-Zeit, wenn $f(n)$ ein Term von n ist).

Definition 2.13. Seien $f, g\colon \mathbb{N} \to \mathbb{N}$. Wir schreiben $f = O(g)$, wenn es ein $c \in \mathbb{N} \setminus \{0\}$ gibt, so dass für alle hinreichend großen n, $f(n) \leq c \cdot g(n)$. Wir schreiben $f = o(g)$ wenn es für alle $\varepsilon > 0$ ein n_0 gibt, so dass für alle $n \geq n_0$, $f(n) \leq \varepsilon \cdot g(n)$.

Beispiel: Wir analysieren nun die Zeitkomplexität folgender TM M, die $A(M) = \{0^k 1^k \mid k \geq 0\}$ hat.

Sei ein Input w der Länge n gegeben:

1. Gehe das Band entlang, und lehne w ab, wenn es ein 0 rechts von einer 1 gibt.

2. Wenn es sowohl eine 0 als auch eine 1 auf dem Band gibt, gehe das Band entlang und lösche sowohl eine einzige 0 als auch eine einzige 1. Wiederhole gegebenenfalls Schritt 2.

3. Wenn eine 0 aber keine 1 übrig bleibt oder umgekehrt, lehnt w ab. Wenn weder eine 0 noch eine 1 übrigbleibt, dann akzeptiert w.

Anweisung 1 braucht n Schritte. M braucht dann noch einmal n Schritte, um den Lesekopf wieder zurück auf die erste Zelle zu bewegen. Insgesamt sind es also $O(n)$ Schritte. In jeder Runde der Anweisung 2 muss M das Band lesen und zwei Symbole löschen. Es kann höchstens $n/2$ Runden gehen, und deshalb ist die Anzahl der Schritte für Anweisung 2 höchstens $n/2 \cdot O(n) = O(n^2)$. Anweisung 3 braucht noch einmal n Schritte. Alle Anweisungen zusammen brauchen also $O(n) + O(n^2) + O(n) = O(n^2)$ Schritte. Die Zeitkomplexität von M ist daher $O(n^2)$.

Definition 2.14. Sei $t\colon \mathbb{N} \to \mathbb{N}$. Die *Zeitkomplexitätsklasse* $[N]ZEIT(t)$ ist die Menge

$$\{A \mid A = A(M) \text{ für eine [nicht-deterministische] } O(t)\text{-Zeit TM } M\}.$$

Wir zeigten, dass $A_0 = \{0^k 1^k \mid k \in \mathbb{N}\}$ zu $ZEIT(O(n^2))$ gehört. Kann man dies verbessern? Tatsächlich gibt es eine $O(n \cdot \lg(n))$-TM mit Akzeptierungsmenge A_0. Außerdem gibt es eine $O(n)$-TM mit zwei Bändern und Akzeptierungsmenge A_0. Die Zeitkomplexität kann also von der Wahl des Berechnungsmodells abhängen. Für die verschiedenen Turingmaschinen, die wir bis jetzt betrachteten, gilt:

Satz 2.15. *Sei $t\colon \mathbb{N} \to \mathbb{N}$ so dass für alle n, $t(n) \geq n$.*

1. *Jede t-Zeit-Mehrband-TM ist äquivalent zu einer $O(t^2)$-Zeit-TM mit einem Band. Äquivalent heißt: haben dieselbe Akzeptierungsmenge.*
2. *Jede nicht-deterministische t-Zeit-TM ist äquivalent zu einer (deterministischen) $2^{O(t)}$-Zeit-TM*

Beweis: Man schaut sich die in den Beweisen von Satz 2.6 und Satz 2.10 skizzierten Algorithmen an und schätzt deren Schrittzahl ab.

Deshalb unterscheiden sich Einband- und Mehrband-Turingmaschinen höchstens um die Anwendung eines Polynoms, während sich deterministische und nicht-deterministische Turingmaschinen höchstens um eine Exponentiation unterscheiden. Wir sehen Polynome als klein und Exponentiationen als groß an. Alle vernünftigen deterministischen Berechnungsmodelle sich polynomial äquivalent, d.h. ihre Arbeitsschritte können in polynomialer Zeit ineinander übersetzt werden und ihre Zeitkomplexitäten unterscheiden sich höchstens um die Anwendung einer polynomialen Funktion.

Definition 2.16. Folgende Menge heißt „Komplexitätsklasse P":

$$P = \{A(M) \mid \text{ für ein } k \in \mathbb{N} \text{ ist } M \text{ eine deterministische } n^k\text{-Zeit-TM}\}.$$

D.h., $P = \bigcup_k ZEIT(n^k)$.

Definition 2.17. Folgende Menge heißt „Komplexitätsklasse NP”:

$$NP = \{A(M) \mid \text{ für ein } k \in \mathbb{N} \text{ ist } M \text{ eine nicht-deterministische } n^k\text{-Zeit-TM}\}.$$

D.h., $NP = \bigcup_k NZEIT(n^k)$.

P is unabhängig von der Wahl des Berechnungsmodells, solange es deterministisch ist. Nach heutiger Auffassung entspricht P der Klasse der Probleme (= Akzeptierungsmengen), die prinzipiell mit einem Computer berechnet werden können.

2.6 Beispiele für Probleme in P

Definition 2.18. (1) Ein gerichteter Graph ist eine Stuktur (V, E) aus einer endlichen Menge V und aus einer Relation $E \subseteq V \times V \setminus \{(v, v) \mid v \in V\}$. V steht für vertex, vertices, E steht für (directed) edge, (gerichtete) Kante.

(2) Seien $s, t \in V$. Ein Pfad in (V, E) von s nach t ist eine Folge $s_0, \ldots, s_n$, so dass $s_0 = s$, $s_n = t$ und für alle i $(s_i, s_{i+1}) \in E$.

$$PFAD = \{(V, E, s, t) \mid (V, E) \text{ ist ein gerichteter Graph mit einem Pfad von } s \text{ nach } t\}.$$

Satz 2.19. *$PFAD \in P$.*

Beweis: Wenn ein Input (V, E, s, t) der Größe n gegeben ist, gehe wie folgt vor:

1. Setze eine Marke auf s.

2. Betrachte die Kanten in E. Wenn a markiert ist, und $(a, b) \in E$, dann markiere auch b.

3. Wiederhole Punkt 2. bis kein weiterer Punkt mehr markiert werden kann.

4. Wenn t eine Marke hat, akzeptiere (V, E, s, t). Sonst lehne (V, E, s, t) ab.

Wir kodieren einen Graphen (V, E) mit $|V| = m$ durch einen $m \times m$-Matrix, haben also Input-Größe $n = m^2$ und $|V| = O(\sqrt{n})$. Anweisungen 1 und 4 brauchen jeweils nur einen Schritt. Anweisung 2 kann höchstens $|V|$ Mal ausgeführt werden. Jede einzelne Anwendung benötigt höchstens $O(n)$ Schritte. Somit ist die Gesamtzahl der Schritte höchstens $1 + 1 + |V| \cdot O(n) = O(n^{\frac{3}{2}})$ und $PFAD \in P$. Hier ist $|V|$ die Mächtigkeit von V, d.h. die Anzahl seiner Elemente. $\dashv$

Definition 2.20. (1) Zwei natürliche Zahlen heißen *relativ prim* zueinander, wenn 1 ihr größter gemeinsamer Teiler ist.

(2)
$$RELPRIM = \{(n, m) \mid n \text{ und } m \text{ sind relativ prim}\}.$$

Satz 2.21. *$RELPRIM \in P$.*

Beweis: Wenn ein Input (x, y) der Größe n gegeben ist, gehe wie folgt vor:

1. Ersetze x durch $x \bmod y$.
2. Vertausche x und y.
3. Wiederhole Anweisung 1 und 2 bis $y = 0$.
4. Wenn $x = 1$, akzeptiere den Input, sonst lehne ihn ab.

Wir analysieren nun die Zeitkomplexität dieses Algorithmus. Nach einer Anwendung von 1 ist $x < y$. Nach einer Anwendung von 2 ist $x > y$, da sie vertauscht wurden. Jede Anwendung von 1 halbiert nun x (oder y, je nachdem, wer gerade dran ist): Wenn $x/2 \geq y$ ist, dann $x \bmod y < y \leq x/2$, und somit erhalten wir eine Zahl $< x/2$. Wenn $x/2 < y$ ist, dann $x \bmod y = x - y < x/2$ und somit erhalten wie wieder ein Zahl $< x/2$.

Mit jeden Anwendung von 2 werden x und y vertauscht, deshalb sind die ursprünglichen Werte von x und y nach je zwei Anwendungen von 1 und 2 halbiert. Daher könnnen 1 und 2 höchstens $2 \cdot \min(\log_2 x, \log_2 y)$ Mal angewendet werden. Da $\log_2 n = O(n)$ und da jede Anwendung von 1 oder 2 nur polynomiale Zeit dauert und zur Kodierung der Zahlen $\leq n$ höchstens noch einmal $\text{poly}(\log_2 n)$ braucht, ist die gesamte Berechnungszeit polynomial. $\dashv$

2.7 NP-Vollständigkeit und der Satz von Cook

An dem Beispiel $PFAD$ sahen wir, dass Mengen die auf den ersten Blick exponentiell erscheinen, dennoch polynomial sein können. Doch nicht alle Probleme sind so einfach wie $PFAD$: Es gibt zahlreiche Probleme, für die nicht bekannt ist, ob sie in P liegen. Viele dieser Probleme liegen in NP, der Klasse mit „Überprüfungen in polynomialer Zeit". Wir beschäftigen uns hier mit einigen berühmten Vertretern, die zudem noch NP-vollständig (s.u.) oder NP-hart sind.

Definition 2.22. Eine *Überprüfung* für eine Menge A ist ein Algorithmus V zur Untersuchung von Paaren (w, c), so dass

$$A = \{w \in \Sigma^* \mid V \text{ akzeptiert } (w, c) \text{ für ein } c\}.$$

Wenn V in polynomialer Zeit in Abhängigkeit von (w, c) läuft (also in Zeit $\leq p_1(|w|+|c|)$ für ein geeignetes Polynom p_1) und wenn die Länge von c polynomial von der Länge von w abhängt (also in Zeit $\leq p_2(|w|)$ für ein geeignetes Polynom p_2), dann sagen wir, dass V eine Überprüfung in polynomialer Zeit ist. Eine Folge c, so dass V (w, c) akzeptiert, heißt *Zertifikat* für w.

Die Überprüfung verwendet also als zusätzliche Eingabe ein $c \in \Gamma^{\leq p_2(|w|)}$ für ein (von A, aber nicht von w abhängendes) Polynom p_2. Hier steht für eine natürliche Zahl y und eine Menge Γ (im vorliegenden Fall das Arbeitsalphabet) der Ausdruck $\Gamma^{\leq y}$ für die Menge aller Folgen (= Wörter) über Γ der Länge höchstens y. Für Γ mit mindestens zwei Elementen ist das Abarbeiten aller Zusatzinformationen also von exponentieller Schrittzahl.

Satz 2.23 (Korollar zu Satz 2.27). *NP ist die Klasse (modulo Isomorphie keine echte Klasse sondern mengenklein) der Mengen, die eine Überprüfung in polynomialer Zeit haben.*

Beweis: Warum genügt einmal Raten (nämlich das Erraten eines Zertifikats) zu Anfang? Die Übergangstafel einer nicht deterministischen TM darf ja bei jedem Berechnungsschritt raten. Wir werden im Beweis des Satzes von Cook 2.27 sehen: Jedes Problem in NP lässt sich durch eine in polynomialer Zeit berechenbare Reduktion in das Problem $3-SAT$ übersetzen, und $3-SAT$ hat einen Algorithmus, der zu Anfang einmal ein Zertifikat polynomialer Länge errät. $\dashv$

Definition 2.24. Sei Σ ein Alphabet. Eine Funktion $f\colon \Sigma^* \to \Sigma^*$ heißt in polynomialer Zeit berechenbar, gdw es eine polynomiale TM gibt, die aus den Input w den Output $f(w)$ (auf den ersten Zellen des Bandes, wenn die Maschine in den Stoppzustand q_{ak} eintritt) liefert. q_{ab} kommt nicht mehr vor.

Definition 2.25. Eine Menge A heißt *in polynomialer Zeit auf eine Menge B reduzierbar*, und wir schreiben $A \leq_P B$ gdw es eine in polynomialer Zeit berechenbare Funktion f gibt so dass für alle $w \in \Sigma^*$:

$$w \in A \text{ gdw } f(w) \in B.$$

f wird eine Reduktion von A auf B in polynomialer Zeit genannt.

Definition 2.26. Eine Menge A heißt *NP-vollständig*, gdw $A \in NP$ und für jedes $B \in NP$, $B \leq_P A$.

Wir werden zeigen, dass es NP-vollständige Mengen gibt, Es ist $P = NP$ gdw eine (und damit jede) NP-vollständige Menge in P ist. Unser Beispiel eine NP-vollständigen Menge ist

$$SAT = \{\varphi \mid \varphi \text{ ist eine erfüllbare Formel der Aussagenlogik}\}.$$

Satz 2.27 (Stephen Cook, 1971, auch Satz von Cook-Levin genannt). *SAT ist NP-vollständig.*

Beweis: Es ist leicht zu sehen, dass SAT in NP ist: Ein Zertifikat für eine Formel φ ist eine Wahrheitsbelegung der Satzvariablen in φ, die φ erfüllt. Da

$$\varphi \in SAT \leftrightarrow \neg\varphi \text{ ist nicht allgemeingültig},$$

bezeugt unser im Beweis des Satzes 1.44 skizzierten Algorithmus, der tatsächlich in NP ist, die Behauptung.

Sei $A \in NP$. Wir zeigen, dass $A \leq_P SAT$ ist. Sei N eine nichtdeterministische n^k-Zeit-TM, so dass $A(N) = A$ für eine feste natürliche Zahl k. Wir schreiben N so, dass N auf jedem Input w genau $|w|^k$ Schritte läuft und dann stoppt und dazu nicht mehr als $|w|^k$ Zellen braucht. Eine Berechnungsmatrix für N mit Input w ist eine $n^k \times n^k$-Matrix, deren Zeilen die Konfigurationen eines Astes einer Berechnung von N auf den Input w hin sind. Wir nehmen an, dass jede Konfiguration mit dem Symbol # beginnt und endet. Die erste Zeile der Matrix ist die Anfangskonfiguration von N mit Input w, und jede weitere Zeile folgt aus ihrer Vorgängerzeile gemäß der nicht-deterministischen Übergangsfunktion von N. Die Berechnungsmatrix akzeptiert w gdw eine ihrer Zeilen eine Akzeptierungskonfiguration ist.

Wir beschreiben nun eine Reduktion f von A auf SAT in polynomialer Zeit. Auf den Input w liefert f eine Formel

$$f(w) = \varphi_w$$

der Aussagenlogik. Sei $C = Q \cup \Gamma$, Q die Zustandsmenge von N, Γ das Bandalphabet von N. Wir nehmen $Q \cap \Gamma = \emptyset$ an, und können dann jede Konfiguration einfacher durch Konkatenation der Tripel aus Definition 2.7 beschreiben. Diese Konfiguationen bilden die Zeilen einer Matrix. Es gibt so viele Zeilen, wielange die Turingmaschine arbeitet. Die Satzsmbole von φ sind von der Form $x_{i,j,s}$ mit $1 \leq i, j \leq n^k$ und $s \in C$. Die $n^k \times n^k$ Matrix hat $(n^k)^2$ Zellen. Die Zelle in Zeile i und Spalte j wird $z(i,j)$ genannt.

φ_w ist von der Form $\varphi_{\text{Zelle}} \wedge \varphi_{\text{Anfang},w} \wedge \varphi_{\text{Bewegung}} \wedge \varphi_{\text{Akzeptierung}}$. Wir beschreiben nun diese vier Teile einzeln.

Das Satzsymbol $x_{i,j,s}$ hat die Bedeutung: $z(i,j)$ enthält das Symbol s. φ_{Zelle} garantiert, dass $z(i,j)$ genau s enthält.

$$\varphi_{\text{Zelle}} = \bigwedge_{1 \leq i,j \leq n^k} [\bigvee_{s \in C} x_{i,j,s} \wedge \bigwedge_{s,t \in C, s \neq t} \neg(x_{i,j,s} \wedge x_{i,j,t})].$$

$\varphi_{\text{Anfang},w}$ garantiert, dass die erste Zeile der Matrix die Anfangskonfiguration von N mit Input w ist.

$$\varphi_{\text{Anfang},w} = x_{1,1,\#} \wedge x_{1,2,q_0} \wedge x_{1,3,w_1} \wedge \cdots \wedge x_{1,n+2,w_n} \wedge x_{1,n+3,\#} \wedge x_{1,n+4,b} \wedge \\ \cdots \wedge x_{1,n^k-1,b} \wedge x_{1,n^k,b}.$$

$\varphi_{\text{Akzeptierung}}$ garantiert, dass eine Akzeptierungkongfiguration in der Matrix auftritt zu einer Zeit vor einer Ablehnungskonfiguration.

$$\varphi_{\text{Akzeptierung}} = \bigvee_{1 \leq j \leq n^k} x_{n^k,j,q_{ak}}$$

Und schließlich garantiert $\varphi_{\text{Bewegung}}$, dass jede Zeile der Matrix eine Konfiguration enthält, die aus der vorherigen Zeile folgt. Eine 2×3-Untermatrix der Matrix heißt *zulässig*, gdw sie der Übergangsfunktion δ von N folgt: Z.B., wenn $\delta(q_1, a) = \{(q_2, b, R)\}$ und $\delta(q_1, b) = \{(q_2, c, L), (q_3, a, R)\}$, dann ist die Untermatrix $\begin{pmatrix} a, q_1, b \\ q_2, a, c \end{pmatrix}$ zulässig, und die Untermatrix $\begin{pmatrix} a, q_1, b \\ q_2, a, a \end{pmatrix}$ ist unzulässig. Auch bei Untermatrizen, die gerade nicht die Stelle des Lesekopfes in der Mitte enthalten, ist klar, wie man Zulässigkeit definiert. An den meisten Stellen ist eine Matrix der Form $\begin{pmatrix} a, b, c \\ a, b, c \end{pmatrix}$ zulässig. Folgendes ist leicht zu sehen: Wenn die erste Zeile der Matrix der Anfangskonfiguration entspricht und jede Untermatrix zulässig ist, dann folgt die Gesamtmatrix der Übergangsfunktion. Nun konstruieren wir die Formel $\varphi_{\text{Bewegung}}$. Die (i,j)-Untermatrix ist die Untermatrix in den Zeilen i und $i+1$ und in den Spalten $j-1, j, j+1$. Deren Zulässigkeit kann wie folgt ausgedrückt werden:

$$\varphi_{\text{zul}(i,j)} = \bigvee_{\binom{a_1,\,a_2,\,a_3}{a_4,\,a_5,\,a_6} \text{ ist zulässig}} (x_{i,j-1,a_1} \wedge x_{i,j,a_2} \wedge x_{i,j+1,a_3} \wedge x_{i+1,j-1,a_4} \wedge x_{i+1,j,a_5} \wedge x_{i+1,j+1,a_6}).$$

Nun beschreibt $\varphi_{\text{Bewegung}}$ die Zulässigkeit aller 2×3-Untermatrizen

$$\varphi_{\text{Bewegung}} = \bigwedge_{1 \le i < n^k, 1 < j < n^k} \varphi_{\text{zul}(i,j)}.$$

Schließlich betrachten wir die Komplexität dieser Reduktion. Die Größe von φ_{Zelle} ist $O(n^{2k})$, von $\varphi_{\text{Anfang},w}$ ist $O(n^k)$, von $\varphi_{\text{Bewegung}}$ und von $\varphi_{\text{Akzeptierung}}$ $O(n^{2k})$. Also erhalten wir insgesamt $O(n^{2k})$. Die Anzahl der Satzvariablen ist $O(n^{2k})$, und deshalb ist die Länge von φ in der Klasse $O(\log_2 n) \cdot O(n^{2k})$, also in $O(n^{2k+1})$ und somit ein Polynom in n. Wir brauchen $O(\log_2 n)$, um die wachsenden Indizes binär zu schreiben, denn das Alphabet ist ja fest, und darf nicht mit n wachsen.

Aus der Konstruktion von $w \mapsto \varphi_w$ folgt, dass es einen nichtdeterministischen akzeptierenden Lauf von N angesetzt auf w gibt genau dann, wenn es eine Belegung der Satzvariablen $x_{i,j,s}$, $i,j \le n^k$, $s \in C$ gibt, so dass φ_w wahr ist. $\dashv$

Der obige Beweis zeigt etwas mehr: Eine KNF-Formel (Formel in konjunktiver Normalform) ist eine Konjunktion von Disjunktionen der Form

$$\bigwedge_{i<k} \bigvee_{j<m_i} L_{i,j}$$

mit Literalen $L_{i,j}$ (siehe Def. 1.29). Wenn für alle i, $m_i \le 3$ ist, sagen wir, dass die Formel eine 3-KNF-Formel ist. 3-*SAT* ist die Menge der erfüllbaren 3-KNF-Formeln.

Korollar 2.28. *3-SAT ist NP-vollständig.*

Beweis. In polynomialer Zeit können wir die Formel φ des obigen Beweises in eine KNF-Formel umformen. Weiterhin können wir eine Disjunktion von Satzsymbolen und Negationen von Satzsymbolen in eine Konjunktion von Disjunktionen von höchtens drei Satzsymbolen und Negationen von Satzsymbolen umformen, z.B. $\varphi = B_1 \vee \cdots \vee B_n$ in

$$\begin{aligned}\varphi_{3-\text{KNF}} = ((B_1 \vee B_2 \vee C_1) \wedge (\neg C_1 \vee B_3 \vee C_2) \wedge (\neg C_2 \vee B_4 \vee C_3) \\ \wedge \cdots \wedge (\neg C_{n-3} \vee B_{n-1} \vee B_n).\end{aligned}$$

Man zeigt induktiv über n: Es gibt eine Wahrheitsbelegung v von $B_1,\dots B_n$, so dass $\bar{v}(B_1 \vee \dots \vee B_n) = W$ gdw es eine Wahrheitsbelegung v_* von $B_1, \dots, B_n$ und $C_1,\dots, C_{n-3}$ gibt, für die $\bar{v_*}(\varphi_{3-\mathrm{KNF}}) = W$.

So ist die Erfüllbarkeit (fast) KNF φ_w zu der Erfüllbarkeit der 3-KNF-Formel $\varphi_{w,3-\mathrm{KNF}}$ äquivalent. Die Formel $\varphi_{w,3-\mathrm{KNF}}$ kann aus φ_w in polynomialer Zeit in der Länge von φ berechnet werden. Es folgt, dass 3-SAT NP-vollständig ist. $\dashv$

Satz 2.29. *2-SAT ist in P.*

(Auch habe ich nirgends einen Beweis gefunden, das Folgende ist selbst ausgedacht. In Sipser steht dies als Übungsaufgabe. Ist 2-SAT P-vollständig?)

Am leichtesten geht es mit der Resolutionsmethode, die wir in Kapitel 6 kennenlernen werden. Es gibt hierzu auch Hinweise in Exercise 36 [6]. Formeln in KNF mit höchstens zwei Literalen pro Konjunktionsglied heißen *Krom-Formeln*. Horn-Formeln (siehe Kapitel 6) sind spezielle Krom-Formeln.

Skizze einer etwa $O(n^9)$-Zeit-Turingmaschine: Sei $\varphi = \bigwedge_{i<I}(a_{i,1} \vee a_{i,2})$ gegeben und $a_{i,j} = A_k$ oder $= \neg A_k$ für Satzvariablen A_0 bis A_n, $n \leq |\varphi|$. Wir berechnen einen gerichteten Graphen $(V, E) = (\{A_i \mid i \leq n\} \cup \{\neg A_i \mid i \leq n\}, E)$, so dass $((a, b) \in E$ und $(\neg b, \neg a) \in E)$ gdw $(\neg a \vee b)$ ein Konjunktionsglied von φ ist oder $(b \vee \neg a)$ ein Konjunktionsglied in φ ist. Die Idee dabei ist: $(\neg a \vee b)$ ist ein Konjunktionsglied, also impliziert φ die Formeln $a \to b$ und $\neg b \to \neg a$. Falls $(\neg)a \vee (\neg)a$ Konjuktionsglied ist, setzen wir $\bar{v_0}(a) = W(F)$ und $\bar{v_0}(\neg a) = F(W)$. (Mit $a = A_i$ oder $\neg A_i$ und dann $\neg a = A_i$.)

Wir erweitern nun in $(2n)^5$ Schritten die Kantenmenge um jeweils eine Kante oder um keine Kante.

1. Wir setzen $E = F_0$.

2. Sei vor dem Schritt s der Graph (V, F_s) gegeben, $F_s \supseteq E$, $v_s \supseteq v_0$. Wir wählen drei Vertizes $a, b, c \in V$, die durch eine Schrittnummerierung für den Schritt s bestimmt werden, so dass jedes der $(2n)^3$ Tripel $(2n)^2$ oft vorkommt. Dann fügen wir $(a, c) \in F_{s+1} \setminus F_s$ gdw $(a, b) \in F_s$ und $(b, c) \in F_s$ und $(a, c) \notin F_s$. Falls $v_s(a) = W$ und $(a, (\neg)b) \in F_{s+1}$ (also $a \to (\neg)b$) aus φ folgt, setzen wir $v_{s+1}(b) = W(F)$ und $v_{s+1}(\neg b)$ gespiegelt. Falls $v_s(a) = F$ und $(\neg a, (\neg)b) \in F_{s+1}$, setzen wir $v_{s+1}(b) = W(F)$ und $v_{s+1}(\neg b)$ gespiegelt.

3. Falls $F_{s+1} \supsetneq F_s$ und v_{s+1} eine partielle Wahrheitsbelegung ist, gehe zu Schritt 2.

4. φ is erfüllbar gdw für kein Literal a, $(\neg a, a)$ $(a, \neg a) \in F_s$ und für kein Literal a, $v_s(a) = W$ und $v_s(a) = F$.

Da $|F_s| \leq n^2$, braucht man $\leq n^4$ Schritte, um zu prüfen, ob für kein Literal a, $(\neg a, a) \in F_s$ $(a, \neg a) \in F_s$.

Wir beweisen die Korrektheit des Algorithmus, indem wir zeigen, dass wir nach positiver Prüfung einen weiteren Algorithmus anschließen können, der eine Wahrheitsbelegung ermittelt, die φ wahr macht.

Wir arbeiten nun $O(n^3)$-viele Schritte zur Erstellung einer Belegung: Sei $F_{s+1} = F_s$ und sei $v_{s+1} = v_s$.

A_i heißt positiv entschieden, wenn $(\neg A_i, A_i) \in F_s$, d.h. wenn $\varphi \models \neg A_i \to A_i$, oder wenn $v_{s+1}(A_i) = W$, und A_i heißt negativ entschieden, wenn $(A_i, \neg A_i) \in$

F_s oder $v_{s+1}(A_i) = F$. Sonst heißt A_i unentschieden. Im ersten Schritt setzen wir $v_{s+2} \supseteq v_{s+1}$ und $v_{s+2}(A_i) = W$ gdw A_i positiv entschieden, und $v_{s+2}(A_i) = F$ gdw A_i negativ entschieden. Danach nehmen wir die erste unentschiedene Satzvariable, sei dies A_j, und setzen $v_{s+3} \supseteq v_{s+2}$ und $v_{s+3}(A_j) = W$ und $v_{s+3}((\neg)A_k) = W$ gdw $(A_j, (\neg)A_k) \in F_s$. Wir behaupten, dass φ erfüllbar ist gdw $\varphi \wedge A_j W$ erfüllbar ist. Sonst gibt es ein A_k, so dass (A_j, A_k) und $(A_j, \neg A_k)$ beide in F_s vorkommen. Dann ist aber nach Schritt 2 $A_j \rightarrow A_k$ und $A_j \rightarrow \neg A_k$ und auch $A_k \rightarrow \neg A_j$ eine Konsequenz von φ, und somit auch $(A_j, \neg A_j) \in F_s$ und A_j negativ entschieden, im Widerspruch zu Unentschiedenheit von A_j mit v_{s+2}. So machen wir induktiv weiter mit der Erweiterung der Belegungen und der Erhaltung der Erfüllbarkeit, bis alle unentschiedenen Variablen belegt sind. Dies geht $r \leq n$ Schritte (und jeder einzelne Schritt hat $\leq 2n^2$ Teilaufgaben zum Eintrage aller Konsequenzen) und das Verfahren endet mit einer Belegung v_{s+1+r}, die φ wahr macht. $\dashv$

2.8 Beispiele von Mengen in NP

Weitere NP-vollständige Probleme sind $CLIQUE$, $HAMPFAD$, $TEILSUMME$.

Beweise zur Vollständigkeit findet man in Michael Sipser "Introduction to the Theorie of Computation" [8]. Wir beweisen nur, dass die Probleme in NP sind.

Definition 2.30. Sei $(V, E) = G$ ein gerichteter Graph. Ein Hamiltonpfad ist ein Pfad, der jeden Punkt genau einmal enthält.

$$HAMPFAD = \{(V, E, s, t) \mid \text{ es gibt einen Hamiltonpfad von } s \text{ nach } t \text{ in } (V, E)\}.$$

Folgender Algorithmus zeigt, dass $HAMPFAD \in NP$:

Für gegebenen Input $((V, E, s, t), c)$:

1. Prüfe, ob $c = (c_0, \ldots, c_{m-1})$ eine Folge der Länge $|V|$ von paarweise ungleichen Elementen aus V ist.

2. Überprüfe, ob s das erste und t das letzte Element von c sind.

3. Schaue, ob für jedes $i < m$, ob $(c_i, c_{i+1}) \in E$. Wenn dies für ein i fehlschlägt, lehne (V, E, s, t, c) ab. Sonst akzeptiere sie.

Nach unseren Vorarbeiten (z.B. auch die Maschine von Beispiel 4) sehen wir, dass dieser Algorithmus in polynomialer Zeit läuft.

Wir wenden uns nun den ungerichteten Graphen zu. Diese sind üblicher als die gerichteten Graphen und werden oft einfach nur Graphen genannt.

Definition 2.31. Ein (ungerichteter) Graph ist eine Paar (V, E), so dass V eine Menge ist und $E \subseteq P \times E \setminus \{(v, v) \mid v \in V\}$ eine symmetrische Relation ist, d.g., dass für alle $(v, w) \in E$ auch $(w, v) \in E$ ist. $(v, w) \in E$ heißt Kante von (V, E), und $v \in V$ heißt Knoten oder Vertex.

Definition 2.32. Sei $k \in \mathbb{N}$. Eine k-Clique in (V, E) ist eine Teilmenge C von V mit $|C| = k$, so dass für je zwei $x \neq y \in C$ gilt: $(x, y) \in E$.

$CLIQUE = \{(V, E, k) \mid (V, E)$ ist ein ungerichteter Graph mit einer k-Clique$\}$.

Folgender Algorithmus zeigt, dass $CLIQUE \in NP$:
Für gegebenen Input $((V, E, k), c)$:
1. Prüfe, ob $c = (c_0, \dots, c_{k-1})$ eine Menge von k Elementen aus V ist.
2. Schaue, ob für jedes $i < j < k$, ob $(c_i, c_j) \in E$.
3. Genau wenn die Antworten zu 1 und zu 2 positiv sind, akzeptiere (V, E, k, c).

Definition 2.33. $TEILSUMME = \{(S, t) \mid S = \{x_1, \dots, x_n\} \subseteq \mathbb{N}, \exists \langle y_1, \dots, y_\ell \rangle$, so dass $y_i \in S$ und $\sum_{i \leq \ell} y_i = t\}$. Die y_i müssen nicht paarweise verschieden sein.

Folgender Algorithmus zeigt, dass $TEILSUMME \in NP$:
Für gegebenen Input $((S, t), c)$:
1. Prüfe, ob c ein Folge von Zahlen mit Summe t ist.
2. Schaue, ob alle Einträge der Folge c Elemente aus S sind.
3. Genau wenn die Antworten zu 1 und zu 2 positiv sind, akzeptiere (S, t, c).

Sind diese Mengen nun auch in P oder nur in NP? Die Antworten auf diese Fragen sind nicht bekannt. Tatsächlich ist nicht bekannt, ob es überhaupt eine Menge in $NP \setminus P$ gibt. Man weiß nur

$$NP \subseteq EXPZEIT = \bigcup_{k \in \mathbb{N}} ZEIT(2^{n^k}).$$

Die Frage $P = NP$ wird heute als das wichtigste offene Problem in der theoretischen Informatik angesehen.

Kapitel 3

Die Logik der ersten Stufe

Wie betrachten jetzt eine Logik, die viel reicher als die Aussagenlogik ist, nämlich die Logik der ersten Stufe, auch Prädikatenlogik genannt. Insbesondere kann diese Logik Ideen ausdrücken, die in verschiedenen mathematischen Theorien vorkommen. Zuerst führen wie eine Beschreibung der Symbole einer Sprache erster Stufe ein:

a) Logische Symbole :

0. Klammern (und),
1. Junktoren $\wedge$, $\vee$, $\to$, $\leftrightarrow$ und $\neg$,
2. Variable $v_0, v_1, \ldots$
3. Gleichheitszeichen $=$,
4. Quantoren $\forall$, $\exists$.

b) Zur Menge τ der nichtlogischen Symbole gehören (je nach Auswahl)

0. Prädikatssymbole: Für jede natürliche Zahl $n > 0$ eine leere, endliche oder unendliche Menge n-stelliger Prädikatssymbole,
2. Konstantensymbole: Eine leere, endliche oder unendliche Menge von Symbolen,
3. Funktionssymbole: Für jede natürliche Zahl $n > 0$ eine leere, endliche oder unendliche Menge n-stelliger Funktionssymbole.

τ wird *Symbolmenge*, *Sprache*, similarity type, signature, Signatur genannt. Es ist auch $\tau = \emptyset$ gestattet. τ und $\mathscr{L}(\tau)$ (s.u.) werden *Sprache* genannt.

Bemerkungen: Das Gleichheitszeichen ist ein zweistelliges Prädikatssymbol, es wird jedoch im Gegensatz zu den anderen zweistelligen Prädikatssymbolen als logisches Symbol betrachtet. Die Wörter „logische Symbole” und „nichtlogische Symbole” sind einfach Namen, die sich, obwohl sie nicht sehr sinnvoll sind, etabliert haben.

Andernfalls sagt man explizit, dass man mit einer Sprache ohne Gleichheit arbeitet (non-equational language).

Beachten Sie, dass es tatsächlich viele verschiedene Sprachen der ersten Stufe gibt, die von der Wahl der Prädikats-, Konstanten- und Funktionssymbole abhängen. Zwei Beispiele sind:

In der *Sprache der Mengenlehre* gibt neben dem Gleichheitszeichen nur ein zweistelliges Prädikatssymbol: $\in$. Es gibt keine anderen Prädikatssymbole jeglicher Stelligkeit und keine Konstantensymbole und keine Funktionssymbole.

In der *Sprache der elementaren Zahlentheorie* gibt es genau ein zweistelliges Prädikatssymbol $<$, ein Konstantensymbol 0, ein einstelliges Funktionssymbol S (für die Nachfolgerfunktion) und drei zweistellige Funktionssymbole $+$, $\cdot$, E, die die Addition, die Multiplikation und die Exponentiation darstellen. Wir schreiben Exy für x^y, da wir alles auf einer Linie schreiben möchten. Die Exponentiation gehört manchmal nicht zur Sprache der Zahlentheorie.

3.1 Terme und Formeln

Wie bisher ist ein Ausdruck eine endliche Folge von Symbolen. Nur bestimmte Ausdrücke haben eine Bedeutung, und diese heißen Formeln. Zunächst legen wir fest, was Terme sind.

Definition 3.1. *τ-Terme.* Es sei τ eine Menge von Funktionssymbolen und Konstantenzeichen.

1. Jedes Konstantensymbol $c \in \tau$ und jede Variable ist ein τ-Term.
2. Wenn $f \in \tau$ ein n-stelliges Funktionssymbol ist und $t_1, \dots, t_n$ τ-Terme sind, dann ist auch $ft_1 \dots t_n$ ein Term. (Beachten Sie, dass wir keine Klammern und keine Kommata schreiben. Wenn Sie dies auf $+$, $\cdot$ und E anwenden, erhalten Sie die sogenannte polnische Notation. Eine Zeitlang wurde die umgekehrte polnische Notation $t_1 \dots t_n f$ auf Hewlett-Packard-Taschenrechnern benutzt.)
3. Nur die Zeichenreihen, die sich durch endlichmalige Anwendung von 1. und von 2. erzeugen lassen, sind τ-Terme.

Natürlich liegt uns hier ein weiteres Beispiel einer induktiven Definition vor: Ein Ausdruck ist demnach ein τ-Term genau dann, wenn er es aufgrund der Regeln 1 und 2 sein muss. Wir sehen, dass die Menge der Terme durch Abschluss der Menge der Konstantensymbole und der Variablen unter den Funktionssymbolen erzeugt wird. Wenn es keine Funktionssymbole in der Sprache gibt, dann sind die Konstantensymbole und die Variablen die einzigen τ-Terme und eine Definition durch Induktion ist nicht notwendig. Einige Beispiele für Terme in der Sprache der elementaren Zahlentheorie:
$+v_2Sv_1$, in verständlicherer Form $v_2 + Sv_1$,
$SSS0$,
$+Ev_1SSEv_300$, zurückübersetzt $v_1^{SSv_3^0} + 0$.

Lemma 3.2. *Kein Term ist echtes Anfangsstück eines anderen Terms.*

Beweis: Induktiv über den Aufbau der Terme. ⊣

Als nächstes kommen wir nun zu der einfachsten Art von Formeln, den sogenannten atomaren Formeln.

Definition 3.3. Es sei τ eine Symbolmenge. *Atomare Formeln, genauer atomare $\mathscr{L}(\tau)$-Formeln*, auch *Primformel* genannt, sind Ausdrücke der Form $Pt_1 \dots t_n$ mit einem n-stelligen Prädikatssymbol P oder dem Gleichheitssymbol $=$ (dann ist $n = 2$) und τ-Termen $t_1, \dots, t_n$.

Dieser Begriff wird also explizit definiert und nicht durch Induktion. Die atomaren Formel sind die Bausteine, aus denen kompliziertere Formeln aufgebaut werden. Die Rolle der atomaren Formeln ist also analog zur Rolle der Satzvariablen in der Aussagenlogik. Wenn die Symbolmenge groß ist, kann schon die Menge der atomaren Formel überabzählbar sein. Überlegen Sie sich ein Beispiel.

Definition 3.4. Nun definieren wir für jede Symbolmenge τ die Menge $\mathscr{L}(\tau)$, die *die Sprache der ersten Stufe zur Symbolmenge τ* oder *die Menge der $\mathscr{L}(\tau)$-Formeln, kurz Formeln* genannt wird: $\mathscr{L}(\tau)$ ist die kleinste Menge, für die folgendes gilt:

1. Jede atomare Formel ist eine Formel.
2. Wenn φ eine Formel ist, dann ich auch $\neg\varphi$ eine Formel.
3. Wenn φ und ψ Formeln ist, dann sind auch $(\varphi \wedge \psi)$, $(\varphi \vee \psi)$, $(\varphi \to \psi)$ und $(\varphi \leftrightarrow \psi)$ Formeln.
4. Wenn ψ eine Formel ist und v eine Variable ist, dann sind auch $\forall v\psi$ und $\exists v\psi$ Formeln.

Wir sagen, dass die Menge der Formeln dadurch erzeugt wird, dass die Menge der atomaren Formeln unter den Junktoren und den Quantoren abgeschlossen wird. Wieder gilt

Lemma 3.5. *Keine Formel ist echtes Anfangsstück einer anderen Formel.*

Wir geben als Beispiele zwei Formeln der Mengenlehre:
$(\forall v_2 \in v_2 v_1)$, in geläufigerer Form $(\forall v_2 v_2 \in v_1)$, auch $(\forall v_2)(v_2 \in v_1)$, $\forall v_2 v_2 \in v_1$ u.ä.,
$\neg\forall v_1 \neg\forall v_2 \in v_2 v_1$ auch $\exists v_1 \forall v_2 v_2 \in v_1$.

Diese erste Formel sagt: „Jede Menge ist ein Element von v_1", und die zweite drückt aus: „es gibt eine Menge, die jede Menge als Element enthält." Es gibt einen wichtigen Unterschied zwischen diesen beiden Beipielen: Im zweiten Beispiel haben wir einen vollständigen Satz, im ersten Beispiel hingegen haben wir eine Formel, deren Bedeutung von der Interpretation der Variablen abhängt. v_1 wird eine freie Variable der ersten Formel genannt. Wir geben jetzt eine genaue Definition dieses Begriffs:

Definition 3.6. Die Eigenschaft *„v tritt frei in der Formel φ auf"* wird induktiv über den Aufbau von φ definiert.

0. Wenn φ atomar ist, dann tritt v frei in φ auf, wenn v eine Variable in φ ist.
1. v tritt frei in $\neg\varphi$ auf, wenn v frei in φ auftritt.
2. Für jeden zweistelligen Junktor $*$ gilt: v tritt frei in $(\varphi * \psi)$ auf, wenn v in φ oder in ψ frei auftritt.

3. v tritt frei in $\forall v_i \varphi / \ \exists v_i \varphi$ auf, wenn v frei in φ auftritt und ungleich v_i ist.

Wir schreiben $\mathrm{fr}(\varphi)$ für die Menge aller Variablen, die frei in φ auftreten,

Definition 3.7. Wenn keine Variable frei in der Formel φ auftritt, dann sagen wir, dass φ ein *Satz* ist.

Die Sätze sind die Formeln mit einer vollständigen Bedeutung. Sie ist unabhängig von der Interpretion der Variaben. Dies werden wir im Koinzidenzlemma beweisen.

Definition 3.8. Die Variablen, die in φ überhaupt vorkommen, heißen die *Variablen von* φ. Die Variablen von φ, die nicht frei in der Formel φ auftreten, heißen die *gebundenen Variablen von* φ.

Wir werden die beiden Weisen des Vorkommens genauer unterscheiden, wenn wir die Semantik der Logik der ersten Stufe vorstellen.

3.2 Abkürzungen und Klammern

Um unsere Fomeln einfacher lesen zu können, erlauben wir uns, Ausdrücke zu schreiben, die streng genommen keine Formeln sind, aber einfach in Formeln übertragen werden können:

$t_1 \neq t_2$ für $\neg = t_1 t_2$

Klammern:
Wir lassen die äußersten Klammern weg. Wir lassen bei Konjunktionen auf mehr als zwei Gliedern die inneren Klammern weg. Selbes bei Disjunktionen.

Manche Autoren gestatten auch:
1. $\alpha \to \beta \to \gamma$ steht für $\alpha \to (\beta \to \gamma)$. Ich empfehle dies nicht. Im Programm (Proof Assistant) "lean" wird es so gemacht.

2. $\wedge$ (manchmal auch $\vee$) bindet stärker als $\to$. Auch das empfehle ich nicht. Wir schreiben $(\chi \wedge \psi) \to \varphi$.

3.3 Wahrheit und Modelle

In der Aussagenlogik weisen Wahrheitsbelegungen den Satzsymbolen und dann auch beliebigen Formeln Wahrheitswerte zu. In der Logik der ersten Stufe wird die Rolle der Wahrheitbelegung durch Strukturen und Belegungen der freien Variablen übernommen. Strukturen liefern eine Bedeutung für die Quantoren und die nichtlogischen Symbole der jeweiligen Sprache ersten Stufe.

Definition 3.9. Eine *Struktur* $\mathfrak{A} = (A, (P^{\mathfrak{A}})_{P \in \tau}, (F^{\mathfrak{A}})_{f \in \tau}, (c^{\mathfrak{A}})_{c \in \tau})$, für eine gegebene Sprache erster Stufe ist eine Funktion, die als Definitionsbereich $\{\forall\}$ vereinigt mit der Menge der nichtlogischen Symbole der Sprache hat und für die folgendes gilt:

0. $\mathfrak{A}$ weist dem Quantor $\forall$ eine nicht leere Menge $|\mathfrak{A}| = A$ zu, die das Universum von $\mathfrak{A}$ oder der Träger von $\mathfrak{A}$ (domain, universe, support) genannt wird. Wir schreiben oft A für $|\mathfrak{A}|$.
1. $\mathfrak{A}$ weist jedem n-stelligen Prädikatssymbol P ein n-stelliges Prädikat $P^{\mathfrak{A}} \subseteq |\mathfrak{A}|^n$ zu, d.h. $P^{\mathfrak{A}}$ ist eine Menge von n-Tupeln von Elementen des Universums $|\mathfrak{A}|$.
2. $\mathfrak{A}$ weist jedem Konstantensymbol c ein Element $c^{\mathfrak{A}} \in |\mathfrak{A}|$ zu.
3. $\mathfrak{A}$ weist jedem n-stelligen Funktionssymbol eine n-stellige Funktion $f^{\mathfrak{A}} \colon |\mathfrak{A}|^n \to |\mathfrak{A}|$ zu.

Die Idee ist: $\mathfrak{A}$ gibt dem Quantor $\forall$ die Bedeutung für jedes Element von A. Außerdem weist $\mathfrak{A}$ den Prädikats-, Funktions- und Konstantensymbolen der Sprache Bedeutungen zu. Wir verlangen, dass die n-stelligen Funktionssymbole so interpretiert werden wie totale Funktionen auf dem Definitionsbereich A^n.

Nun möchten wie eine Bedeutung für den Satz „die Formel φ ist wahr in der Struktur $\mathfrak{A}$" angeben. Hierfür müssen wir zuerst eine Bedeutung für die Vabiablen unserer Logik definieren. Eine Belegung in $\mathfrak{A}$ oder eine $\mathfrak{A}$-Belegung ist eine Funktion s von der Menge der Variablen in $|\mathfrak{A}|$. Wir wollen

„$\mathfrak{A}$ mit der Belegung s erfüllt φ",

kurz $\mathfrak{A} \models \varphi[s]$, auch $(\mathfrak{A}, s) \models \varphi$, für beliebige $\mathfrak{A}$-Belegungen per Induktion über φ definieren.

Zuerst erhalten alle Terme Bedeutungen, indem wir s zu einer Funktion $\bar{s}$ auf der Menge der Terme erweitern:

Definition 3.10. Sei $s \colon \{v_0, v_1, \dots\} \to A$ eine Belegung für die τ-Struktur $\mathfrak{A}$. Wir definieren die Fortsetzung $\bar{s}$ von s auf die Menge der τ-Terme wie folgt:
1. $\bar{s}(v) = s(v)$
2. $\bar{s}(c) = c^{\mathfrak{A}}$
3. Wenn $t_1, \dots, t_n$ Terme sind und f ein n-stelliges Funktionssymbol ist, dann ist $\bar{s}(ft_1 \dots t_n) = f^{\mathfrak{A}}(\bar{s}(t_1), \dots, \bar{s}(t_n))$.

Um die Quantorenschritte in der Definition von „$\mathfrak{A}$ mit der Belegung s erfüllt φ" richtig zu behandeln, definieren wir zunächst eine Art, Belegungen s in einem ihrer Argumente eventuell zu ändern:

Definition 3.11. Wenn s eine $\mathfrak{A}$-Belegung ist und x eine Variable ist und a ein Element von $|\mathfrak{A}|$ ist, dann ist $s(x|a)$ (sprich „s, x ersetzt durch a") die folgende $\mathfrak{A}$-Belegung:

$$s(x|a)(y) = \begin{cases} s(y), & \text{falls } y \neq x \\ a, & \text{falls } y = x. \end{cases}$$

Warum haben wir „eventuell zu ändern" geschrieben? Wenn $s(x) = a$, dann ist $s(x|a) = s$.

Definition 3.12. *„$\mathfrak{A}$ mit der Belegung s erfüllt φ"* wird induktiv über den Aufbau von φ gleichzeitig für alle s definiert:

Wenn φ atomar ist, dann gilt:

1. $\mathfrak{A} \models t_1 = t_2[s]$ gdw $\bar{s}(t_1) = \bar{s}(t_2)$.
2. Für jedes n-stellige Prädikatssymbol P, $\mathfrak{A} \models Pt_1 \ldots t_n[s]$ gdw $\langle \bar{s}(t_1), \ldots, \bar{s}(t_n) \rangle \in P^{\mathfrak{A}}$.

Für zusammengesetzte φ gilt:

3. $\mathfrak{A} \models \neg\varphi[s]$ gdw nicht $\mathfrak{A} \models \varphi[s]$.
4. $\mathfrak{A} \models (\varphi \rightarrow \psi)[s]$ gdw $\mathfrak{A} \not\models \varphi[s]$ oder $\mathfrak{A} \models \psi[s]$. Aus den boole'schen Regeln leitet man nun $\models$ für die Junktoren $\wedge$, $\vee$, $\leftrightarrow$ her.
5. $\mathfrak{A} \models \forall x\varphi[s]$ gdw für jedes $a \in A$, $\mathfrak{A} \models \varphi[s(x|a)]$. Nach der Definition von $\exists$ in Abschnitt 3.2 ist nun (abgeleitet, muss also zwingend so definiert werden): $\mathfrak{A} \models \exists x\varphi[s]$ gdw für ein $a \in A$, $\mathfrak{A} \models \varphi[s(x|a)]$.

Statt $\mathfrak{A} \models \varphi[s]$ schreibt man auch $(\mathfrak{A}, s) \models \varphi$, und man nennt eine Struktur $\mathfrak{A}$ zusammen mit einer Belegung s eine *Interpretation* $(\mathfrak{A}, s)$.

Aus der „Erfüllt-Relation" $\models$ wird nun, analog wie in der Aussagenlogik, die logische Implikation für die Logik der ersten Stufe definiert. Es folgt nun eine der wichtigsten Definitionen in der gesamten Vorlesung:

Definition 3.13. Sei Γ eine Formelmenge und sei φ eine Formel. *„Γ impliziert φ"*, (oder *„aus Γ folgt φ"* oder *„φ folgt aus Γ"* oder, in Zeichen, $\Gamma \models \varphi$) gdw für jede Struktur $\mathfrak{A}$ der Sprache von $\Gamma \cup \{\varphi\}$ und für jede $\mathfrak{A}$-Belegung s gilt: Wenn $(\mathfrak{A}, s)$ jedes Element von Γ erfüllt, dann erfüllt $(\mathfrak{A}, s)$ auch φ.

Wenn Γ nur ein Element Γ hat, dann schreiben wir $\gamma \models \varphi$ statt $\{\gamma\} \models \varphi$. Wenn Γ leer ist, dann schreiben wir $\models \varphi$ statt $\emptyset \models \varphi$, und sagen dass φ allgemeingültig (valid, gültig) ist. Die gültigen Formeln der Logik der ersten Stufe entsprechen den Tautologien der Aussagenlogik. φ ist gültig gdw wenn jede Struktur $\mathfrak{A}$ und jede $\mathfrak{A}$ Belegung zusammen φ erfüllen.

Vorsicht: Die logische Implikation $\models$ heißt auf deutsch die Folge-Relation oder auch die Folgerungsrelation. Letzters ist leicht zweideutig, wie Sie in den nächsten Vorlesungen sehen werden (es kann auch die Relation $\vdash$ unter „Folgerung" verstanden werden). Wenn wir den Gödel'schen Vollständigkeitssatz bewiesen haben (ab Ende des Kapitels 4), können wir mit der Zweideutigkeit leben, da wir dann wissen, dass $\models$ und $\vdash$ äquivalent sind.

Der folgende Satz beschreibt die Rolle der freien Variablen.

Satz 3.14. Koinzidenzlemma. *Seien s_1 und s_2 $\mathfrak{A}$-Belegungen, die auf den Variablen, die frei in φ auftreten, übereinstimmen. Dann gilt*

$$\mathfrak{A} \models \varphi[s_1] \text{ gdw } \mathfrak{A} \models \varphi[s_2].$$

Beweis durch Induktion über den Aufbau von φ.

Fall 1: φ ist atomar. $\varphi = Pt_1 \ldots t_n$ (P kann auch das Gleichheitszeichen sein). In diesem Fall tritt jede Variable in φ frei auf. Deshalb stimmen s_1 und s_2 auf allen Variablen in t_i für jedes i überein, und $\bar{s_1}(t_i) = \bar{s_2}(t_i)$ für jedes i. Daher ist $\mathfrak{A} \models \varphi[s_1]$ gdw $(\bar{s_1}(t_1), \ldots, \bar{s_1}(t_n)) \in P^{\mathfrak{A}}$ gdw $(\bar{s_2}(t_1), \ldots, \bar{s_2}(t_n)) \in P^{\mathfrak{A}}$ gdw $\mathfrak{A} \models \varphi[s_2]$, wie gewünscht.

Fall 2 und Fall 3: φ ist von der Form $\neg\alpha$ oder $(\alpha \to \beta)$. Diese Fälle folgen direkt aus den Induktionsannahmen.

Fall 4 (der interessante Fall): $\varphi = \forall x\psi$. Dann ist jede Variable frei in φ gdw sie frei in ψ und nicht gleich x ist. Somit stimmen $s_1(x|a)$ und $s_2(x|a)$ für jedes Element a von $|\mathfrak{A}|$ auf den freien Variablen von ψ überein. Nach der Induktionannahme gilt für jedes $a \in A$, $\mathfrak{A} \models \psi[s(x|a)]$ gdw $\mathfrak{A} \models \psi[s(x|a)]$. Da dies für beliebiges a gilt, folgt, dass $\mathfrak{A} \models \forall x\psi[s_1]$ gdw $\mathfrak{A} \models \forall x\psi[s_2]$. $\dashv$

Korollar 3.15. *Wenn σ ein Satz ist, dann gilt $\mathfrak{A} \models \sigma[s]$ für keine Belegung oder aber für alle Belegungen.*

Wir sagen, dass der Satz σ wahr in $\mathfrak{A}$ ist gdw wenn $\mathfrak{A}$ mit mit jeder $\mathfrak{A}$-Belegung erfüllt, geschrieben $\mathfrak{A} \models \sigma$. Sonst ist σ falsch in $\mathfrak{A}$. Wenn σ wahr in $(\mathfrak{A}, s)$ ist, dann ist $(\mathfrak{A}, s)$ ein Modell von σ. Wenn Σ eine Menge von Sätzen ist, dann ist $(\mathfrak{A}, s)$ ein Modell von Σ gdw $(\mathfrak{A}, s)$ ein Modell von jedem Satz Σ ist.

Korollar 3.16. *Seien Σ eine Menge von Sätzen und sei τ ein Satz. Dann gilt $\Sigma \models \tau$ gdw jedes Modell von Σ auch ein Modell von τ ist.*

Beachten Sie, dass die Definition der logischen Implikation für die Logik der ersten Stufe viel komplizierter als für die Aussagenlogik ist. Um zu entscheiden, ob eine Formel gültig ist oder nicht, müssen wir jede Struktur der Sprache und jede Belegung beachten und dann prüfen, ob $(\mathfrak{A}, s)$ φ erfüllt oder nicht. Deshalb ist es nicht klar, ob die Menge der gültigen Formeln der Logik der ersten Stufe entscheidbar ist. Tatsächlich werden wir zeigen, dass die Gültigkeit nicht entscheidbar ist. Überraschenderweise impliziert der Gödelsche Vollständigkeitssatz, dass die Menge der gültigen Formeln der Logik der ersten Stufe effektiv aufzählbar ist. Wir werden diese Aufzählbarkeit im Anschluss an den Gödel'schen Vollständigkeitssatz zeigen und die Unentscheidbarkeit im Kapitel über die Gödel'schen Unvollständigkeitssätze zeigen.

In einer vielleicht etwas willkürlichen, jedoch allgemein akzeptierten Aufteilung gehört $\models$ zu den semantischen Begriffen, die sich unmittelbar mit den Strukturen beschäftigen. Wir werden nun noch zwei weitere semantische Begriffe definieren, bevor wir uns auf die sogenannte synaktische Seite begeben.

Definition 3.17. Sei $\mathfrak{A}$ eine Struktur und sei $\mathrm{fr}(\varphi) \subseteq \{v_0, \ldots, v_{n-1}\}$ und seien $a_0, \ldots, a_{n-1}$ Elemente aus $|\mathfrak{A}|$. Dann schreiben wir

$$\mathfrak{A} \models \varphi[a_0, \ldots, a_{n-1}]$$

gdw $(\mathfrak{A}, s) \models \varphi$ für $s(v_i) = a_i$, $i = 0, \ldots, n-1$.

Definition 3.18. $P \subseteq |\mathfrak{A}|^n$ ist *definierbar in* $\mathfrak{A}$ gdw es eine Formel φ mit den freien Variablen aus $\{v_0, \ldots, v_{n-1}\}$ gibt, so dass $P = \{(a_0, \ldots, a_{n-1}) \mid \mathfrak{A} \models \varphi[a_0, \ldots, a_{n-1}])\}$. Wir sagen hierzu φ *definiert* P *in* $\mathfrak{A}$.

Wir betrachten zum Beispiel Definierbarkeit in der Struktur $\mathfrak{N} = (\mathbb{N}, 0, 1, +, \cdot)$: Einige Relationen sind in $\mathfrak{N}$ definierbar, andere nicht. Da es nur abzählbar viele Formeln gibt, sind nur abzählbar viele Relationen definierbar. Es gibt jedoch überabzählbar viele Relationen auf $\mathbb{N}$. Zum Beispiel ist die Ordungsrelation $\{(n,m) \mid n < m\}$ definierbar und die Menge der Primzahlen und die Exponentiationsrelation $\{(n,m,s) \mid s = n^m\}$. Die erstgenannte wird durch die Formel $\exists v_3 v_1 + Sv_3 = v_2$ definiert, die zweite durch die Formel $1 < v_1 \wedge \forall v_2 \forall v_3 (v_1 = v_2 \cdots v_3 \to v_2 = 1 \vee v_3 = 1)$ und die dritte durch eine kompliziertere Formel, die wir im sogenannten Gödel'schen β-Lemma 5.23 sehen werden. Wir werden im Kapitel über die Unvollständigkeitssätze auch zeigen, dass jede entscheidbare Relation und jede effektiv aufzählbare Relation und viele weitere Relationen auf $\mathbb{N}$ in $\mathfrak{N}$ definierbar sind.

Es ist oft viel schwieriger zu zeigen, dass eine gegebene Relation in einer gegebenen Struktur $\mathfrak{A}$ nicht definierbar ist. Ein hinreichendes Kriterium für Nicht-Definierbarkeit in $\mathfrak{A}$ ist zum Beispiel die Existenz eines Automorphismus i von $\mathfrak{A}$, der P bewegt, d.h. $i''P := \{i(\bar{a}) \mid \bar{a} \in P\} \neq P$. Hier schreiben wir $i(\bar{a}) = (i(a_0), \dots, i(a_{n-1}))$.

Definition 3.19. Seien $\mathfrak{A}$ und $\mathfrak{B}$ τ-Strukturen. i ist ein *Isomorphismus von* $\mathfrak{A}$ *auf* $\mathfrak{B}$ gdw $i \colon A \to B$ eine Bijektion ist, die alle Symbole in τ erhält. In Formeln heißt dies:

(a) $i(c^{\mathfrak{A}}) = c^{\mathfrak{B}}$ für jedes Konstantensymbol $c \in \tau$,

(b) $P^{\mathfrak{A}}(a_0, \dots, a_{n-1})$ gdw $P^{\mathfrak{B}}(i(a_0), \dots, i(a_{n-1}))$ für jedes n und jedes Symbol für ein n-stelliges Prädikat $P \in \tau$ und

(c) $i(f^{\mathfrak{A}}(a_0, \dots a_{n-1})) = f^{\mathfrak{B}}(i(a_0), \dots i(a_{n-1}))$ für jedes n und jedes Symbol für eine n-stellige Funktion in τ.

$\mathfrak{A}$ und $\mathfrak{B}$ heißen isomorph, in Zeichen $\mathfrak{A} \cong \mathfrak{B}$, wenn es einen Isomorphismus von $\mathfrak{A}$ auf $\mathfrak{B}$ gibt.

Isomorphismen von $\mathfrak{A}$ auf $\mathfrak{A}$ heißen Automorphismen.

Lemma 3.20. *Jeder Isomorphismus von* $\mathfrak{A}$ *auf* $\mathfrak{B}$ *ist treu für jede definierbare Relation.*

Beweis: Dies zeigt man induktiv über den Aufbau der definierenden Formel. Übung!

Kapitel 4

Der Gödel'sche Vollständigkeitssatz

In diesem Kapitel beweisen wir Schritt für Schritt den Gödel'schen Vollständigkeitssatz.

4.1 Beweistheorie

Der Begriff der Gültigkeit ist in der Logik der ersten Stufe ($\Gamma \models \varphi$) recht kompliziert. Gibt es eine einfachere Definition? Können wir zum Beispiel für eine Sprache mit nur endlich vielen nichtlogischen Symbolen die Menge der gültigen Formeln effektiv aufzählen? Gödel gab eine positive Antwort. Dieses Ergebnis ist eines der Korollare aus dem Gödel'schen Vollständigkeitssatz.

Die Idee des Vollständigkeitssatzes ist einfach. Wir zeigen, dass jede gültige Formel mit einfachen Mitteln aus der leeren Voraussetzungsmenge beweisbar ist. Hierzu präzisieren wir den Beweisbegriff wie folgt: Wir geben eine entscheidbare Menge Λ gültiger logischer Axiome an. Dann nennen wir eine endliche Folge von Formeln einen *formalen Beweis*, wenn jede dieser Formeln ein logisches Axiom ist oder unter Verwendung der Schlussregel Modus Ponens aus früheren Formeln folgt. Die Modus Ponens Regel (MP) lautet:

$$\text{Aus } \varphi \text{ und } (\varphi \to \psi) \text{ folgt } \psi.$$

Die Menge der formalen Beweise ist entscheidbar. (Überlegen Sie sich das, nachdem Sie die Definition gesehen haben.) Nun ist eine Formel ein formaler Satz gdw sie die letzte Formel eines Beweises ist. Da die Menge der Beweise entscheidbar ist, folgt, dass die Menge der formalen Sätze effektiv aufzählbar ist. Da die Menge der formalen Beweise mit der Menge der gültigen Formeln übereinstimmt, ist daher such die letztere effektiv aufzählbar.

Formale Beweise

Wir geben nun ein kurzes vollständiges Regelwerk für formale Beweise an: Sei eine Sprache τ der ersten Stufe gegeben. Wir folgen hier dem dem sogenannten Hilbert'schen Beweiskalkül, der in den Büchern von Joseph Shoenfield [7], Herbert Enderton [2] und Ziegler [9] beschrieben wird. Eine äquivalenter

Beweiskalkül, der sogenannte Sequenzenkalkül, wird in den Lehrbüchern von Hermes [4], Ebbinghaus, Flum und Thomas [1] und auch Ziegler [9] beschrieben.

Wir definieren zunächst eine besondere Menge von Formeln Λ, die Menge der logischen Axiome.

Einschub: Wie ist der Begriff „Axiom" im Duden definiert?

Definition 4.1. Der sogenannte *Hilbertkalkül* für Beweise. Die Menge Λ der *logischen Axiome* besteht aus sechs Teilmengen. φ heißt eine Verallgemeinerung von ψ gdw für ein $n \geq 0$ und Variablen $x_0 \dots, x_{n-1}$ gilt

$$\varphi = \forall x_0 \dots \forall x_{n-1} \psi.$$

Wir gestatten auch $n = 0$. Die logischen Axiome sind Verallgemeinerungen von Formeln aus den Gruppen 1 bis 6. Seien x, y Variablen und α, β Formeln. Wir beschreiben nun die Gruppen 1 bis 6:

1. Alle Tautologien der Aussagenlogik.
2. Ersetzungsaxiome: $\forall x \alpha \to \alpha^x_t$, wenn t für x in α eingesetzt werden kann. Wann t überhaupt für x eingesetzt werden kann, wird in Definition 4.2 beschrieben. Die Formel α^x_t wird in Definition 4.4 definiert.
3. $\forall x(\alpha \to \beta) \to (\forall x \alpha \to \forall x \beta)$.
4. $\alpha \to \forall x \alpha$, wenn x nicht frei in α auftritt.
5. $x = x$.
6. $x = y \to (\alpha \to \alpha')$, wenn α atomar ist und α' aus α dadurch entsteht, dass x durch y an einigen Stellen ersetzt wird.

Sei nun Γ eine Menge von Formeln und sei φ eine Formel. Ein Beweis von φ aus Γ ist eine Folge $\langle \alpha_0, \dots, \alpha_n \rangle$ von Formeln, so dass $\alpha_n = \varphi$ und für jedes $i \leq n$ gilt:

1. $\alpha_i \in \Gamma \cup \Lambda$ oder
2. es gibt $j, k < i$ so dass $\alpha_k = (\alpha_j \to \alpha_i)$ ist. In diesem Falle ergibt sich α_i aus vorangehenden α_j und α_k aus dem modus ponens.

Wenn es einen Beweis von φ aus Γ gibt, dann sagen wir, dass φ aus Γ *beweisbar* ist, und schreiben $\Gamma \vdash \varphi$.

Gruppe 1: Tautologien.

Die Tautologien der Logik der ersten Stufe ergeben sich aus den Tautologien der Aussagenlogik, die nur $\neg$ und $\to$ (nicht aber $\wedge$, $\vee$, $\leftrightarrow$ enthalten), indem wir jedes Satzsymbol durch eine Formel der Logik der ersten Stufe ersetzen. In Gruppe 1 nehmen wie auch alle Verallgemeinerungen solcher Formeln auf. Zum Beispiel gehört die Formel $\forall x[(\forall y \neg Py \to \neg Px) \to (Px \to \neg \forall y \neg Py)]$ zu Gruppe 1, weil sie eine Verallgemeinerung der Formel in eckigen Klammern ist und sich die Formel in eckigen Klammern aus der folgenden Tautologie der Aussagenlogik ergibt:

$$(A \to \neg B) \to (B \to \neg A).$$

Eine andere Art, die Menge der Tautologien der ersten Stufe zu erklären, ist folgende: Nennen wir eine Formel *Satzsymbolformel* gdw sie entweder atomar ist oder von der Form $\forall x\alpha$ ist. Bei manchen Autoren werden die Satzsymbolformeln auch Primformeln genannt, aber prim war bei uns schon ein Synonym für atomar. Jede Formel der Logik der ersten Stufe wird aus Satzsymbolformeln mithilfe der Junktoren $\neg$ und $\rightarrow$ aufgebaut. Nun definieren wir die Tautologien der Aussagenlogik um, indem wir die Primformeln wie Satzsymbole behandeln und nur die Junktoren $\neg$ und $\rightarrow$ verwenden. Dann darf „außen" noch verallgemeinert werden. Das Ergebnis ist die Menge der Tautologien der Logik der ersten Stufe.

Gruppe 2: Ersetzung. Von der Technik her ist dies die komplexeste Axiomengruppe.

Definition 4.2. Wir definieren induktiv über den Aufbau von α, wann *t für x in α eingesetzt werden kann.* Man sagt hierfür auch *„x ist frei für t in α".*

1. t kann in atomaren α immer für x eingesetzt werden.
2. t kann in $\neg\alpha$ für x eingesetzt werden gdw dies für α möglich ist. t kann in $(\alpha \rightarrow \beta)$ für x eingesetzt werden, gdw dies für α und für β der Fall ist.
3. t kann in $\forall y\alpha$ für x eingesetzt werden, wenn
   (a) x in $\forall y\alpha$ nicht frei auftritt, oder
   (b) $x \neq y$ und y in t nicht auftritt und t in α für x eingesetzt werden kann.

Lemma 4.3. *x ist frei für t in α gdw kein freies Vorkommen von x im Wirkungsbereich eines Quantors von α ist, der eine Variable von t bindet.*

Definition 4.4. Nun definieren wir für Formeln α, Variablen x und Terme t die Formel α^x_t, unter der Voraussetzung, dass t für x in α eingesetzt werden kann. α^x_t heißt α, x ersetzt durch t. Die Formel α^x_t wird indukiv über den Aufbau von α definiert, und — Vorsicht — ist nicht immer definiert.

1. Für atomares α ergibt sich α^x_t, indem wir in α alle x durch t ersetzen.
2. $(\neg\alpha)^x_t = \neg\alpha^x_t$, wenn t für x in α eingesetzt werden kann, undefiniert sonst, Ebenso definiert man $(\alpha \rightarrow \beta)^x_t = (\alpha^x_t \rightarrow \beta^x_t)$, wenn t für x in α eingesetzt werden kann und in β eingesetzt werden kann, und undefiniert sonst,
3. $(\forall y\alpha)^x_t$ ist
   (a) $\forall y\alpha$, falls $y = x$ oder falls $y \neq x$ und x in $\forall y\alpha$ nicht frei auftritt,
   (b) $\forall y\alpha^x_t$, falls $x \neq y$ und y in t nicht auftritt und t in α für x eingesetzt werden kann
   (c) in allen anderen Fällen ist die Ersetzung undefiniert.

Hier wird es interessant. Es scheint, dass $(\forall x\alpha \to \alpha^x_t)$ ein vernünftiges (d.h. korrektes, s.u.) Axiom ist und dass wir die ungewöhnliche Einschränkung der Definition nicht brauchen. Ohne die Einschränkung bei der Definition kann das Axiom aber falsch sein, wenn nämlich x für ein freies y eingesetzt wird. Zum **Beispiel** schauen wir die Formel $\alpha = \exists y(y \neq x)$ und $t = y$ an. Dann wird $\forall x\alpha \to \alpha^x_t$ zu

$$\forall x \exists y(x \neq y) \to \exists y(y \neq y),$$

was offensichtlich für jede Struktur falsch ist, deren Universum mehr als ein Element hat. Wir lösen dieses Problem, indem wir α, x und t einschränken wie in Definition 4.2. Erhalten wir nun tatsächlich eine gültige Formel $\forall x\alpha \to \alpha^x_t$, wenn t für x in α eingesetzt werden kann?

Lemma 4.5 (Ersetzungslemma oder Substitutionslemma)**.** *Wenn der Term t in der Formel α für die Variable x eingesetzt werden kann, dann*

$$\mathfrak{A} \models \alpha^x_t[s] \text{ gdw } \mathfrak{A} \models \alpha[s(x|\bar{s}(t))].$$

Beweis: Wir führen den interessantesten Schritt vor: Sei $\varphi = \forall y\alpha$, und sei $x \neq y$ und trete y nicht in t auf und möge t für x in α eingesetzt werden können. Dann
$\mathfrak{A} \models \varphi^x_t[s]$
gdw $\mathfrak{A} \models \forall y\alpha^x_t[s]$
gdw f.a. $a \in A$, $\mathfrak{A} \models \alpha^x_t[s(y|a)]$
gdw (IV) f.a. $a \in A$, $\mathfrak{A} \models \alpha[s(y|a)(x|\bar{s}(y|a)(t))]$
gdw (da $x \neq y$ und da y in t nicht auftritt) f.a. $a \in A$, $\mathfrak{A} \models \alpha[s(x|\bar{s}(t))(y|a)]$
gdw $\mathfrak{A} \models \forall y\alpha[s(x|\bar{s}(t))]$
gdw $\mathfrak{A} \models \varphi[s(x|\bar{s}(t))]$.

Wir führen einen weiteren interessanten Schritt vor, den von Definition 4.2 Punkt 4(b): Sei $\varphi = \forall y\alpha$, und sei $x \neq y$ und trete x nicht frei in $\forall y\alpha$ auf. Dann
$\mathfrak{A} \models \varphi^x_t[s]$
gdw $\mathfrak{A} \models \forall y\alpha^x_t[s]$
gdw f.a. $a \in A$, $\mathfrak{A} \models \alpha^x_t[s(y|a)]$
gdw (IV) f.a. $a \in A$, $\mathfrak{A} \models \alpha[s(y|a)(x|\bar{s}(y|a)(t))]$
gdw (da $x \neq y$ und da x in α nicht frei auftritt, Koinzidenzlemma, Satz 3.14) f.a. $a \in A$, $\mathfrak{A} \models \alpha[s(y|a)]$
gdw $\mathfrak{A} \models \forall y\alpha[s]$
gdw (wieder Koinzidenzlemma, Satz 4.14) $\mathfrak{A} \models \forall y\alpha[s(x|\bar{s}(t))]$
gdw $\mathfrak{A} \models \varphi[s(x|\bar{s}(t))]$. $\dashv$

Korollar 4.6. *Die Ersetzungsaxiome sind gültig.*

Beweis: Nehmen wir an, dass t in α für x eingesetzt werden kann und $(\mathfrak{A}, s)$ die Formel $\forall x\alpha$ erfüllt . Wir müssen zeigen, dass $(\mathfrak{A}, s)$ auch α^x_t erfüllt. Wir wissen, dass $\mathfrak{A} \models \alpha[s(x|a)]$ für jedes $a \in |\mathfrak{A}|$. Dann gilt dies insbesondere für $a = \bar{s}(t)$. Damit ist die Formel $(\forall x\alpha \to \alpha^x_t)$ gültig, wie gewünscht. $\dashv$

Satz 4.7 (Gültigkeitssatz)**.** *Jedes Axiom ist gültig, d.h., wenn φ ein logisches Axiom ist und $\mathfrak{A}$ eine Struktur ist und s eine $\mathfrak{A}$-Belegung ist, dann gilt* $(\mathfrak{A}, s) \models \varphi$.

Beweis: Beachten wir zuerst, dass jede Verallgemeinerung einer gültigen Formel auch gültig ist. Wenn zum Beispiel φ die Formel $\forall x\psi$ mit einem gültigen ψ ist, dann haben wir $\mathfrak{A} \models \psi[s]$ für alle $\mathfrak{A}$, s, gdw $\mathfrak{A} \models \psi[s(x|a)]$ für alle $\mathfrak{A}$, s, a, gdw $\mathfrak{A} \models \varphi[s]$.

Derselbe Beweis funktioniert, wenn φ von der Form $\forall x_1 \dots \forall x_n\psi$ ist.

Somit ist zu zeigen, dass alle Formeln in allen sechs Axiomengruppen gültig sind. Das ist klar für die Tautologien, da sie sich aus Tautologien der Aussagenlogik ergeben. Wir haben für die 2. Gruppe schon gezeigt, dass die Ersetzungsaxiome gültig sind.

Nun kommen wir zur Gruppe 3:
Betrachten wir das Axiom

$$\forall x(\alpha \to \beta) \to (\forall x\alpha \to \forall x\beta)$$

Um die Gültigkeit dieses Axioms zu sehen, genügt es zu zeigen, dass $\mathfrak{A} \models \forall x\beta[s]$, wenn $\mathfrak{A} \models \forall x(\alpha \to \beta)[s]$ und $\mathfrak{A} \models \forall x\alpha[s]$. Die beiden Voraussetzungen implizieren $\mathfrak{A} \models (\alpha \to \beta)[s]$ für alle $a \in |\mathfrak{A}|$ und $\mathfrak{A} \models \alpha[s(x|a)]$ für alle $a \in |\mathfrak{A}|$. Zusammen ergibt dies und $\mathfrak{A} \models \beta[s(x|a)]$ für alle $a \in |\mathfrak{A}|$, also $\mathfrak{A} \models \forall x\beta[s]$.

Gruppe 4. Betrachten wir das Axiom

$$\alpha \to \forall x\alpha,$$

für α, in dem x nicht frei auftritt. Für die Korrektheit haben wir zu zeigen: Für alle $(\mathfrak{A}, s)$: Wenn $(\mathfrak{A}, s) \models \alpha$, dann auch $(\mathfrak{A}, s) \models \forall x\alpha$. Letzteres heißt, dass für jedes $a \in A$, $\mathfrak{A} \models \alpha[s(x|a)]$. Nun stimmen aber s und $s(x|a)$ für alle $a \in |\mathfrak{A}|$ auf den freien Variablen von α überein, da x nicht frei in α auftritt. Nach dem Koinzidenzlemma für Belegungen gilt, dass $\mathfrak{A} \models \varphi[s(x|a)$ für jedes $a \in \mathfrak{A}$. Deshalb ist $\mathfrak{A} \models \forall x\varphi[s]$.

Gruppe 5: Klar.

Gruppe 6: Wir nehmen ein atomares α und betrachten das Axiom. $x = y \to (\alpha \to \alpha')$. Hierbei sei α' die Abwandlungvon α, in der x an manchen Stellen durch y ersetzt wurde. Wir zeigen für alle $(\mathfrak{A}, s)$: Wenn $\mathfrak{A} \models \alpha[s]$ und $s(x) = s(y)$, dann $\mathfrak{A} \models \alpha'[s]$. Nehmen wir an, dass $\alpha = Pt_1 \dots t_n$ und $\alpha' = Pt'_1 \dots t'_n$ und dass t'_i aus t_i durch Ersetzung einiger x durch y entsteht. Die Relation P kann auch das Gleichheitszeichen sein. Da $s(x) = s(y)$ ist, ist auch $\bar{s}(t_i) = \bar{s}(t'_i)$, wie man induktiv über den Aufbau der Terme zeigt. Daher ist $(\bar{s}(t_1), \dots, \bar{s}(t_n)) \in P^{\mathfrak{A}}$ gdw $(\bar{s}(t'_1), \dots, \bar{s}(t'_n)) \in P^{\mathfrak{A}}$ und daher $\mathfrak{A} \models Pt_1 \dots t_n[s]$ gdw $\mathfrak{A} \models Pt'_1 \dots t'_n[s]$. Es folgt, dass $\mathfrak{A} \models \alpha'[s]$. $\dashv$

So, nun haben wir also zwei Begriffe logischer Implikation: $\Gamma \vdash \varphi$ bedeutet, dass es einen Beweis von φ aus Γ (und den logischen Axiomen Λ) gibt. Und $\Gamma \models \varphi$ (Γ impliziert φ logisch) bedeutet: Für jede Struktur $\mathfrak{A}$ und jede

Belegung s gilt: Wenn $(\mathfrak{A}, s)$ alle γ aus Γ erfüllt, dann erfüllt $(\mathfrak{A}, s)$ auch φ. Der Gültigkeitssatz oder Korrektheitssatz stellt eine Verbindung zwischen den beiden Begriffen her:

Satz 4.8 (Korrektheitssatz). *Wenn* $\Gamma \vdash \varphi$, *dann* $\Gamma \models \varphi$.

Beweis: Per Induktion über die kürzeste Länge eines formalen Beweises von φ aus Γ. Sei $\varphi_1 \ldots \varphi_n = \varphi$ ein Beweis von φ aus Γ minimaler Länge.

Fall 1: φ ist ein logisches Axiom. Dann folgt aus dem Gültigkeitssatz, dass $\emptyset$ die Formel φ logisch impliziert. Daher impliziert Γ die Formel φ logisch.

Fall 2: $\varphi \in \Gamma$. Dann impliziert Γ φ.

Fall 3: Es gibt $i, j < n$, so dass $\varphi_i = (\varphi_j \rightarrow \varphi)$. Nach der Induktionsannahme impliziert Γ die beiden Formeln φ_i und φ_j logisch, weil es kürzerere Beweise aus Γ für φ_i und für φ_j gibt. Es folgt, dass $\Gamma\varphi$ logisch impliziert. $\dashv$

Der Korrektheitssatz liefert auch ein Korollar, das die Begriffe Konsistenz und Erfüllbarkeit in eine Richtung verbindet:

Definition 4.9.

1. Eine Formelmenge Γ ist *widerspruchsfrei* oder *konsistent* (consistent) gdw es keine Formel φ gibt, so dass Γ sowohl φ als auch $\neg\varphi$ beweist.
2. Γ ist erfüllbar (satisfiable) gdw es eine Struktur $\mathfrak{A}$ und eine Belegung s gibt, so dass $(\mathfrak{A}, s) \models \varphi$ für jedes $\varphi \in \Gamma$.

Die Konsistenz wird als syntaktischer Begriff aufgefasst, da sie sich mit dem Konzept „Beweis" befasst, die Erfüllbarkeit hingegen wird als semantischer Begriff aufgefasst, da sie sich mit dem Konzept „Struktur" befasst. Der Korrektheitsatz liefert uns nun folgendes

Korollar 4.10. *Wenn* Γ *erfüllbar ist, dann ist* Γ *auch konsistent.*

Beweis: Wenn Γ nicht konsistent wäre, dann würde Γ sowohl φ als auch $\neg\varphi$ für ein φ beweisen. Wegen des Korrektheitssatzes bedeutet dies, dass Γ die Formel φ und $\neg\varphi$ logisch impliziert. Da $\varphi \wedge \neg\varphi$ in jeder Struktur $\mathfrak{A}$ mit jeder $\mathfrak{A}$-Belegung s falsch ist, kann Γ nicht erfüllbar sein. $\dashv$

Der Vollständigkeitssatz liefert Umkehrungen zum Korrektheitssatz und zu dessen Korollar.

Satz. Der Gödel'sche Vollständigkeitssatz.

(a) *Jede konsistente Formelmenge ist erfüllbar.*

(b) *Wenn* $\Gamma \models \varphi$, *dann* $\Gamma \vdash \varphi$.

Auf den folgenden sechs Seiten werden wir den Satz beweisen.

4.2 Metasätze

Dem Beweis des Vollständigkeitssatzes stellen wir zunächst einige Sätze über formale Beweise voran. Diese werden Metasätze genannt, da sie Sätze über das Konzept von Sätzen sind. Diese Ergebnisse erklären auch unsere Wahl der logischen Axiome.

Wir sagen, dass $\{\alpha_1, \ldots, \alpha_n\}$ die Formel β tautologisch impliziert gdw die Formel $(\bigwedge_{1 \le i \le n} \alpha_i) \to \beta$ ein Axiom aus der ersten Gruppe ist.

Lemma 4.11. *(Metasatz über die Tautologische Implikation) Wenn* Γ *die Formeln* $\alpha_0, \alpha_1, \ldots, \alpha_{n-1}$ *beweist, und* $\{\alpha_0, \ldots, \alpha_{n-1}\}$ *die Formel* β *tautologisch impliziert, dann beweist* Γ *auch* β.

Beweis: Da die Formel $\bigwedge_{i<n} \alpha_i \to \beta$ eine Tautologie ist, ist sie ein logisches Axiom und somit aus Γ beweisbar. Wir können nun den modus ponens n Mal anwenden, um zu zeigen, dass Γ die Formel β beweist:

$$\Big(\bigwedge_{i<n} \alpha_i \to \beta\Big) \leftrightarrow (\alpha_0 \to (\alpha_1 \to \cdots \to (\alpha_{n-1} \to \beta) \ldots))$$

$\dashv$

Lemma 4.12 (Deduktionsmetasatz). *Wenn* $\Gamma \cup \{\gamma\}$ *die Formel* φ *beweist, dann beweist* Γ *auch die Formel* $(\gamma \to \varphi)$.

Beweis: Wir zeigen durch Induktion über die kürzeste Länge eines Beweises von φ aus $\Gamma \cup \{\gamma\}$, dass $(\gamma \to \varphi)$ aus Γ beweisbar ist.

Fall 1: $\varphi = \gamma$. Dann beweist Γ offensichtlich $(\gamma \to \varphi)$, da $\gamma \to \gamma$ eine Tautologie ist.

Fall 2: φ ist ein logisches Axiom oder ein Element von Γ. Dann beweist schon Γ alleine die Formel φ. Und $\{\varphi\}$ impliziert tautologisch $(\gamma \to \varphi)$. Es folgt aus der tautologischen Implikation, dass Γ die Formel $(\gamma \to \varphi)$ beweist.

Fall 3: φ entsteht durch modus ponens aus ψ und $\psi \to \varphi$. Nach Induktionsannahme beweist Γ dann die Formeln $(\gamma \to \psi)$ und $(\gamma \to (\psi \to \varphi))$. Aus dem Lemma über die tautologischen Implikation, angewandt mit der Tautologie $(\gamma \to \varphi) \leftrightarrow (\gamma \to \psi) \wedge (\gamma \to (\psi \to \varphi))$, folgt nun, dass Γ die Formel $(\gamma \to \varphi)$ beweist. $\dashv$

Lemma 4.13 (Reductio ad absurdum, Widerspruchsbeweis, RAA). *Wenn* $\Gamma \cup \{\varphi\}$ *nicht konsistent ist, dann beweist* Γ *die Formel* $\neg\varphi$.

Beweis: Nach Annahme gibt es eine Formel β, so dass $\Gamma \cup \{\varphi\}$ sowohl β als auch $\neg\beta$ beweist. Mit dem Deduktionsmetasatz erhalten wir, dass Γ sowohl $(\varphi \to \beta)$ als auch $(\varphi \to \neg\beta)$ beweist. Es folgt aus der tautologischen Implikation, dass Γ die Formel $\neg\varphi$ beweist.

Viele einzelne Beweisschritte verstecken sich hinter dem vorigen deutschen Satz: Aus $(\varphi \to \beta)$ beweist man mit einer Tautologie und MP $(\neg\beta \to \neg\varphi)$. Aus $(\varphi \to \neg\beta)$ beweist man mit einer Tautologie und MP $(\neg\neg\beta \to \neg\varphi)$. Hieraus erhält man mit MP $(\neg\beta \to \neg\varphi) \wedge (\neg\neg\beta \to \neg\varphi)$. Außerdem ist $(\neg\beta \vee \neg\neg\beta)$ eine Tautologie. MP gibt nun $\neg\varphi$. ⊣

Lemma 4.14 (Erste Regel über die Verallgemeinerung)). *Wenn Γ die Formel φ beweist, und x in keiner Formel von Γ frei auftritt, dann beweist Γ die Formel $\forall x\varphi$.*

Beweis. Wir zeigen durch Induktion über die kürzeste Länge eines Beweises von φ aus Γ, dass Γ die Formel $\forall x\varphi$ beweist.

Fall 1: φ ist ein logisches Axiom. Dann ist $\forall x\varphi$ auch ein logisches Axiom und somit beweist Γ die Formel $\forall x\varphi$. Wir haben vereinbart, dass alle Verallgemeinerungen logischer Axiome wieder logische Axiome sind.

Fall 2: $\varphi \in \Gamma$. Dann tritt x nicht frei in φ auf. Deshalb ist $(\varphi \to \forall x\varphi)$ ein Axiom der Gruppe 4. Somit beweist Γ sowohl φ als auch $(\varphi \to \forall x\varphi)$. Also beweist Γ auch $\forall x\varphi$.

Fall 3: φ entsteht via modus ponens aus ψ und $\psi \to \varphi$. Jedoch gehört folgende Formel zu Axiomengruppe 3:

$$\forall x(\psi \to \varphi) \to (\forall x\psi \to \forall x\varphi).$$

Aus der Induktionsvoraussetzung erhalten wir $\Gamma \vdash \forall x\psi$ und $\Gamma \vdash \forall x(\psi \to \varphi)$. Aus der Anwendung des modus ponens auf das Axiom erhalten wir einen Beweis von $\forall x\varphi$ aus Γ. ⊣

Lemma 4.15. *Einfache Tatsachen über die Ersetzung.*

(a) *x kann in jeder Formel für sich selbst eingesetzt werden.*

(b) *t kann in φ für x eingesetzt werden, wenn keine Variable von φ in t auftritt.*

(c) *Wenn x, y Variablen sind und y nicht in φ auftritt, dann kann x in φ^x_y für y eingesetzt werden, und es gilt $(\varphi^x_y)^y_x = \varphi$ (x wird zu y und dann wieder zu x).*

(d) *Wenn x, y, z Variablen sind und $x \neq z$ ist und t für x in φ eingesetzt werden kann, dann kann t für x in φ^y_z eingesetzt werden.*

(e) *Wir nehmen an, dass t für x in φ eingesetzt werden könne, y ein Variable sei, die nicht in φ auftrete, und c ein Konstantensymbol sei. Der Term t^c_y und die Formel φ^c_y entstehen dadurch, dass wir in t und φ c durch y ersetzen. Dann kann in φ^c_y t^c_y für x eingesetzt werden.*

Beweisskizze: Induktiv über den Aufbau von φ werden die gebundenen Variablen in geeignete neue Variablen umbenannt. ⊣

Lemma 4.16 (Zweite Regel über die Verallgemeinerung). *Wenn Γ die Formel φ beweist und c ein Konstantensymbol ist, das nicht in Γ auftritt, dann gibt es eine Variable y, die in φ nicht auftritt, so dass Γ die Formel $\forall y\varphi^c_y$ beweist.*

Beweis: Induktiv über den Aufbau eines Beweises von φ aus Γ. Beim Induktionsschritt: modus ponens.

Lemma 4.17 (Dritte Regel über die Verallgemeinerung)**.** *Wenn Γ die Formel φ^x_c beweist und c ein Konstantensymbol ist, das weder in Γ noch in φ auftritt, dann beweist Γ die Formel $\forall x\varphi$, und es gibt eine Ableitung hierfür, in der c nicht auftritt.*

Beweis: Vom vorigen Lemma haben wir eine Ableitung von $\forall y((\varphi^x_c)^c_y)$ aus Γ, in der c nicht auftritt, wenn y neu ist. Aber da c nicht in φ auftritt, haben wir auch

$$((\varphi^x_c)^c_y) = \varphi^x_y.$$

Es bleibt zu zeigen, dass $\forall y \varphi^x_y \vdash \forall x\varphi$. Dies folgt, wenn man weiß, dass $\forall y\varphi^x_y \to \varphi$ ein Axiom ist. Dies folgt aus $(\varphi^x_y)^y_x = \varphi$ und den Ersetzungsaxiomen in Gruppe 2. ⊣

Lemma 4.18 (Metasatz über die Umbenennung von Variablen)**.** *Wenn φ eine Formel, t ein Term und x eine Variable ist, dann gibt es eine Formel φ' so dass*

(a) t für x in φ' eingesetzt werden kann, und

(b) $\varphi \to \varphi'$ und $\varphi' \to \varphi$ beweisbar sind.

Beweis: **Übung**

Wir wiederholen noch einmal:

Satz 4.19. Der Gödel'sche Vollständigkeitssatz

(a) Jede konsistente Formelmenge ist erfüllbar.

(b) Wenn $\Gamma \models \varphi$, dann $\Gamma \vdash \varphi$

Beweis: Es genügt, (a) zu beweisen, denn aus (b) erhält man (a) wie folgt: Impliziere Γ die Formel φ logisch. Wir möchten zeigen, dass Γ φ auch beweist. Wenn $\Gamma \cup \{\neg\varphi\}$ inkonsistent ist, dann beweist Γ die Formel φ durch reductio ad absurdum, und deshalb beweist Γ φ durch tautologische Implikation, wie gewünscht. Daher können wir annehmen, dass $\Gamma \cup \{\neg\varphi\}$ konsistent ist. Dann ist nach (a) dieses auch erfüllbar. Dies widerspricht unserer Annahme, dass Γ die Formel φ logisch impliziert.

Umgekehrt gilt auch, dass aus (b) (a) folgt: Sei Γ nicht erfüllbar. Dann gilt für alle φ, dass $\Gamma \models \varphi \wedge \neg\varphi$. Dann gilt nach (b), $\Gamma \vdash \varphi \wedge \neg\varphi$, also ist Γ inkonsistent.

Nun beweisen wir (a). Sei Γ konsistent. Dann definieren wir eine neue Formelmenge Δ in einer um die Konstantensymbole $c_0, c_1 \dots$ erweiterten Sprache, so dass folgendes gilt:

(i) $\Gamma \subseteq \Delta$,

(ii) Δ ist in der erweiterten Sprache *maximalkonsistent*, d.h. für alle $\varphi \in \mathscr{L}(\tau \cup \{c_i \mid i \in \mathbb{N}\})$ ist $\varphi \in \Delta$ oder ist $\neg\varphi \in \Delta$.

(iii) Δ ist eine *Henkin-Menge*, d.h. für jede Formel φ und jede Variable x gibt es eine Konstante c, so dass die Formel $(\neg\forall x\varphi \to \neg\varphi^x_c)$ ein Element von Δ ist.

Danach konstruieren wir aus Δ eine Struktur $\mathfrak{A}$ und eine Belegung s, so dass $(\mathfrak{A}, s)$ φ erfüllt.

Wir beschreiben nun die Definition von Δ, die sich in abzählbar viele Erweiterungen der ersten Art und eine Erweiterung der zweiten Art zergliedert. Für jede Formel der Sprache $\mathscr{L} = \mathscr{L}_0$ und jede Variable x wählen wir ein neues Konstantensymbol c^x_φ, das wir zu $\mathscr{L}$ hinzufügen, und erhalten so die Sprache $\mathscr{L}_1$. Dann erweitern wir Γ für alle $\varphi \in \mathscr{L}$ und alle Variablen x um die Formel $(\neg\forall x\varphi \to \neg\varphi^x_{c^x_\varphi})$ und erhalten so Γ_1.

Behauptung 4.20. Γ_1 *ist konsistent.*

Beweis: Annahme: Γ_1 wäre widerspruchsvoll. Wir denken uns Γ_1 schrittweise aus Γ aufgebaut, und nehmen die minimale Schrittzahl, so dass $\Gamma \cup \{\psi_0, \dots, \psi_n, \psi_{n+1}\}$ widerspruchsvoll ist. Dann folgt nach RAA: $\Gamma\cup\{\psi_0, \dots, \psi_n\} \vdash \neg\psi_{n+1}$. ψ_{n+1} ist von der Form $(\neg\forall x\varphi \to \neg\varphi^x_{c^x_\varphi})$. Also ist $\neg\psi_{n+1} = \neg\forall x\varphi \wedge \varphi^x_{c^x_\varphi}$. Wir haben daher $\Gamma \cup \{\psi_0, \dots, \psi_n\} \vdash \neg\forall x\varphi$ und $\Gamma \cup \{\psi_0, \dots, \psi_n\} \vdash \varphi^x_{c^x_\varphi}$.

Aus letzterem erhalten wir mit dem Lemma 4.7 (der dritten Regel über die Verallgemeinerung) $\Gamma\cup\{\psi_0, \dots, \psi_n\} \vdash \forall x\varphi$. Nun ist also schon $\Gamma\cup\{\psi_0, \dots, \psi_n\}$ widerspruchsvoll, im Widerspruch zur Minimalität. $\dashv$

Jetzt wiederholen wir diese Konstruktion, und erhalten aus der Sprache $\mathscr{L}_1$ die Sprache $\mathscr{L}_2$ und die Formelmenge Γ_2, so dass Γ_2 alle Formeln der Form $(\neg\forall\varphi \to \neg\varphi^x_{c^x_\varphi})$ enthält, für jede Formel φ, die in $\mathscr{L}_1$, aber nicht in $\mathscr{L}_0$ ausgedrückt werden kann. Dann folgt wieder aus der Behauptung 4.11, dass die Konsistenz von Γ_1 jene von Γ_2 impliziert. Durch wiederholte Erweiterungen dieser Art erhalten wir eine Kette $\Gamma \subseteq \Gamma_1 \subseteq \Gamma_2 \dots$ in den Spachen $\mathscr{L} \subseteq \mathscr{L}_1 \subseteq \mathscr{L}_2 \dots$. Schließlich definieren wir $\Gamma^* = \bigcup\{\Gamma_n \,|\, n \in \mathbb{N}\}$ und $\mathscr{L}^* = \bigcup\{\mathscr{L}_n \,|\, n \in \mathbb{N}\}$. Eine aufsteigende Vereinigung von widerspruchsfreien Mengen ist widerspruchsfrei, da man für einen Widerspruch nur endlich viel Elemente braucht.

Behauptung 4.21. Γ^* *ist Teilmenge einer maximalkonsistenten Formelmenge* Δ *in der Sprache* $\mathscr{L}^*$. Δ *ist ebenfalls eine Henkin-Menge, da Erweiterungen von Henkin-Mengen in derselben Sprache Henkin-Mengen bleiben.*

Beweis: Wir beweisen die Behauptung hier nur für den Fall eine abzählbaren Formelmenge Γ. Die Behauptung für überabzählbare Γ ist auch richtig, doch hierzu braucht man überabzählbar lange Auflistungen, und das heißt überabzählbare Ordinalzahlen.

Behauptung 4.22. *Jede konsistente Formelmenge* Γ^* *in einer abzählbaren Sprache* $\mathscr{L}$ *kann zu einer maximalkonsistenten Formelmenge* Δ *erweitert werden.*

Beweis: Sei $\varphi_0, \varphi_1 \dots$ eine Liste der Formel der Sprache $\mathscr{L}^*$. Wir definieren ψ_n per Induktion über n. Wenn $\Gamma \cup \{\psi_0, \dots, \psi_{n-1}\} \cup \{\varphi_n\}$ konsistent ist, dann setzen wir $\psi_n = \varphi_n$. Sonst beweist $\Gamma \cup \{\psi_0, \ldots, \psi_{n-1}\}$ die Formel $\neg\varphi_n$ und wir definieren $\psi_n = \neg\varphi_n$. Induktiv über n zeigt man nun, dass $\Gamma \cup \{\psi_0, \ldots, \psi_n\}$ konsistent ist. Daher ist $\Gamma \cup \{\psi_i \mid i < \omega\}$ konsistent. Da für jede Formel φ oder $\neg\varphi$ zu Δ gehört, ist Δ maximalkonsistent, wie gewünscht. $\dashv$

Somit sind nun (i), (ii) und (iii) unseres Beweisplanes erreicht. Wir definieren nun die Struktur $\mathfrak{A}$ wie folgt: Sei $\mathcal{T}$ die Menge der $\mathscr{L}^*$-Terme. Wir schreiben $t \approx t_2$ und sagen, dass t_1 und t_2 äquivalent sind, wenn die Formel $t_1 = t_2$ in Δ ist.

Behauptung 4.23. *$\approx$ ist eine Äquivalenzrelation.*

Beweis: Dies folgt aus den Fakten über die Gleichheit (a), (b), (c) und Einsetzen der Terme für x, y, z. Falls dies verboten sein sollte, nimmt man zuerst andere Variablennamen anstelle von x, y, z, die nicht in den fraglichen Termen vorkommen. $\dashv$

Nun schreiben wir $[t]$ für die Äquivalenzklasse von t. Das Universum von $\mathfrak{A}$ ist die Menge $\mathcal{T}/\approx$ all dieser Äquivalenzklassen. Nun sei

(a) Für jedes n-stelliges Prädikatssymbol P
$$P^{\mathfrak{A}} = \{\langle [t_1], \ldots, [t_n]\rangle \mid Pt_1 \ldots t_n \in \Delta\}.$$

(b) Für jedes n-stelliges Funktionssymbol P
$$f^{\mathfrak{A}}([t_1], \ldots, [t_n]) = [ft_1 \ldots t_n].$$

(c) Für jedes Konstantensymbol c ist $c^{\mathfrak{A}} = [c]$.

Behauptung 4.24. *$P^{\mathfrak{A}}$ und $f^{\mathfrak{A}}$ sind wohldefiniert. Wenn für $i < n$, $[t_i] = [t'_i]$, dann gehört $Pt_0 \ldots t_{n-1}$ zu Δ gdw $Pt'_0 \ldots t'_{n-1}$ zu Δ gehört. $[ft_0 \ldots t_{n-1}] = [ft'_0 \ldots t'_{n-1}] \in \Delta$.*

Beweis: Die Wohldefiniertheit folgt aus den Teilen (d) und (e) des Lemmas über die Fakten über die Gleichheit. Wenn $[t_i] = [t'_i]$, dann ist $t_i = t'_i \in \Delta$. Daher ist wieder nach den Fakten über die Gleichheit und der Abgeschlossenheit von Δ unter $\vdash$ (Δ ist ja maximal widerspruchsfrei) auch $(Pt_0 \ldots t_{n-1} \leftrightarrow Pt'_0 \ldots t'_{n-1})$ in Δ gehört. Hieraus folgt dann, dass $Pt_0 \ldots t_{n-1}$ zu Δ gehört gdw $Pt'_0 \ldots t'_{n-1}$ zu Δ gehört. Wenn für alle i $t_i = t'_i \in \Delta$, dann ist nach selben Argumenten auch $ft_0 \ldots t_{n-1} = ft'_0 \ldots t'_{n-1} \in \Delta$. $\dashv$

Die A-Belegung s ist definiert durch $s(x) = [x]$ für jede Variable x.

Behauptung 4.25. *$(\mathfrak{A}, s) \models \varphi$ gdw $\varphi \in \Delta$.*

Beweis: Wir verwenden die erste Regel über die Verallgemeinerung, den Metasatz über die tautologische Implikation, das Ersetzungslemma, die Eigenschaften einer maximalkonsistenten Henkin-Menge und den Metasatz über die Umbenennung von Variablen.

Zuerst beachten wir, dass für jeden Term t die Gleichheit $\bar{s}(t) = [t]$ gilt. Dies beweisen wir durch Induktion über den Aufbau von t: Wenn t eine Variable oder ein Konstantensymbol ist, dann folgt dies aus der Definition von s. Wenn t von der Form $ft_1, \ldots t_n$ ist dann haben wir $\bar{s}(t) = f^{\mathfrak{A}}(\bar{s}(t_1), \ldots, \bar{s}(t_n))$, das nach Induktionsannahme gleich $f^{\mathfrak{A}}([t_1], \ldots [t_n])$ ist. Letzteres ist nach Definition von $f^{\mathfrak{A}}$ gleich $[t]$.

Wir beweisen nun die zentrale Behauptung der Induktion über den Aufbau von φ. φ sei atomar. Wenn φ die Formel $t_1 = t_2$ ist, dann haben wir $\mathfrak{A} \models \varphi[s]$ gdw $\bar{s}(t_1) = \bar{s}(t_2)$ gdw $[t_1] = [t_2]$ gdw $t_1 = t_2 \in \Delta$. Wenn φ die Formel $Pt_1 \ldots t_n$ ist, dann $\mathfrak{A} \models \varphi[s]$ gdw $\langle \bar{s}(t_1), \ldots, \bar{s}(t_n) \rangle) \in P^{\mathfrak{A}}$ gdw $\langle [t_1], \ldots, [t_n] \rangle \in P^{\mathfrak{A}}$ gdw $Pt_1 \ldots t_n \in \Delta$. Der letzte Schritt verwendete die Definition von $P^{\mathfrak{A}}$.

$\varphi = \neg\psi$. Wir haben $(\mathfrak{A}, s) \models \varphi$ gdw $(\mathfrak{A}, s) \not\models \psi$ gdw $\psi \notin \Delta$ gdw $\varphi \in \Delta$. Der letzte Schritt verwendete die Maximalkonsistenz von Δ.

$\varphi = (\psi \to \gamma)$. Wir haben $\mathfrak{A} \models (\psi \to \gamma)[s]$ gdw ($\mathfrak{A} \not\models \psi[s]$ oder $\mathfrak{A} \models \gamma[s]$) gdw ($\psi \notin \Delta$ oder $\gamma \in \Delta$) gdw ($\neg\psi \in \Delta$ oder $\gamma \in \Delta$). Letzteres impliziert wegen der Maximalität von Δ, dass $(\psi \to \gamma) \in \Delta$. Sei umgekehrt $(\psi \to \gamma) \in \Delta$. Dann können wegen der Konsistenz von Δ nicht sowohl ψ als auch $\neg\gamma$ zu Δ gehören. Wegen der Maximalkonsistenz von Δ erhalten wie daher $\neg\psi \in \Delta$ oder $\gamma \in \Delta$.

$\varphi = \forall x\psi$. Zu zeigen ist $\mathfrak{A} \models \forall x\psi[s]$ gdw $\forall x\psi \in \Delta$. Erfülle $\mathfrak{A}$ die Formel $\forall x\psi$ mit s. Wir wählen ein Konstantensymbol c, so dass das Axiom $(\neg\forall x\psi \to \neg\psi^x_c)$ zu Δ gehört. Dies ist möglich, da Δ eine Henkin-Menge ist. Nach Annahme gilt $\mathfrak{A} \models \psi[s(x|[c])]$ und $[c] = \bar{s}(c)$. Nach Ersetzungslemma ist $\mathfrak{A} \models \psi^x_c[s]$. Nach Induktionsannahme ist $\psi^x_c \in \Delta$ und deshalb $\neg\forall x\psi \notin \Delta$ wegen des obigen Henkinaxioms. Weil Δ maximal konsistent ist, ist daher $\forall x\psi \in \Delta$.

Nun nehmen wir an, dass $\mathfrak{A} \not\models \forall x\psi[s]$. Dann gibt es einen Term t, so dass $\mathfrak{A} \not\models \psi[s(x|[t])]$ und $[t] = \bar{s}(t)$. Wir möchten nun mithilfe des Ersetzungslemmas folgern, dass $\mathfrak{A} \not\models \psi^x_t[s]$. Das Problem ist aber, dass wir nicht wissen, ob das Ersetzungslemma anwendbar ist, da nicht klar ist, ob t für x in ψ eingesetzt werden kann. Mit dem Metasatz zu Umbenennung von Variablen können wir eine Formel ψ' wählen, so dass $\psi \vdash \psi' \vdash \psi$ und so dass in ψ' t für x eingesetzt werden kann. Nach der ersten Verallgemeinerungsregel haben wir $\forall x\psi \vdash \forall x\psi'$, da $\forall x\psi$ die Formel ψ und ψ die Formel ψ' beweist. Wir haben:

$\mathfrak{A} \not\models \psi[s(x|[t])]$ nach Voraussetzung. Da ψ und ψ' logisch äquivalent sind, ist $\mathfrak{A} \not\models \psi'[s(x|[t])]$. Nach dem Ersetzungslemma folgt hieraus $\mathfrak{A} \not\models (\psi')^x_t[s]$. Dann ist nach Induktionsannahme $(\psi')^x_t \notin \Delta$. Da $\forall x\psi' \to (\psi')^x_t$ ein logisches Axiom und Δ abgeschlossen unter beweisbarer Implikation ist, haben wir daher $\forall x\psi' \notin \Delta$. Da $\forall x\psi \vdash \forall x\psi'$, ist $\forall x\psi \notin \Delta$. $\dashv$

4.3 Korollare aus dem Vollständigkeitssatz

Der Vollständigkeitssatz hat wichtige Folgen für die Relation $\models$ der logischen Implikation und für die Größe von Modellen von Theorien in der Sprache erster Stufe.

Satz 4.26. *Der Kompaktheitssatz.*

(a) *Wenn jede endliche Teilmenge von Γ erfüllbar ist, dann ist auch Γ erfüllbar.*

(b) *Wenn Γ die Formel φ logisch impliziert, dann gibt es ein endliches $\Gamma_0 \subseteq \Gamma$, so dass Γ_0 die Formel φ impliziert.*

Beweis. (a) Wegen des Vollständigkeitssatzes sind Erfüllbarkeit und Konsistenz gleichwertig. Deshalb genügt es, (a) für „konsistent” statt „erfüllbar” zu beweisen. Wenn jede endliche Teilmenge von Γ konsistent ist, dann ist auch Γ konsistent, da ein Widerspruchsbeweis aus Γ nur eine endliche Teilmenge von Γ verwenden würde.

(b) Der Vollständigkeitssatz impliziert, dass die Relationen $\vdash$ (beweist) und $\models$ (impliziert logisch) gleichwertig sind. Deshalb genügt es, (b) für die Relation $\vdash$ zu beweisen. Wenn Γ die Formel φ beweist, dann gibt ein einen endlichen Beweis von φ aus Γ. Sei Γ_0 die endliche Menge von Formeln aus Γ, die im Beweis verwendet werden. Dann beweist Γ_0 φ. $\dashv$

Ein Beispiel für eine Anwendung des Kompaktheitssatzes: Ein Modell einer Satzmenge Γ ist eine Struktur $\mathfrak{A}$, in der jeder Satz aus Γ wahr ist. Da Γ nur Sätze enthält, brauchen wir keine Belegung. Der Kompaktheitssatz impliziert, dass es keine Satzmenge gibt, deren Modelle genau die Strukturen mit endlichen Universum sind: Wenn Γ eine solche Satzmenge wäre und für n der Satz φ_n sagt, dass es mindestens n Elemente im Universum gibt, dann wäre die Menge $\Gamma^* = \Gamma \cup \{\varphi_n \mid n < \omega\}$ endlich erfüllbar. („Γ ist endlich erfüllbar” ist ein Jargon-Ausdruck, und heißt korrekt: „jede endliche Teilmenge von Γ ist erfüllbar”.) Nach dem Kompaktheitssatz hat Γ^* ein Modell. Diese ist auch ein Modell von Γ und hat natürlich ein unendliches Universum.

Unsere nächste Anwendung des Vollständigkeitssatzes verwendet die Begriffe „Entscheidbarkeit” und „effektive Aufzählbarkeit”, die wir im Kapitel über Komplexitätstheorie definierten. Eine Sprache erster Stufe heißt „effektiv” gdw wenn die Menge ihrer nichtlogischen Symbole abzählbar ist und darüber hinaus die drei Relationen

$$\{(P,n) \mid P \text{ ist ein } n\text{-stelliges Prädikatssymbol}\}$$
$$\{(f,n) \mid f \text{ ist ein } n\text{-stelliges Funktionssymbol}\}$$
$$\{c \mid c \text{ ist ein Konstantensymbol}\}.$$

entscheidbar sind. Zum Beispiel ist jede Sprache mit nur endlich vielen nichtlogischen Symbolen effektiv. Das Teilgebiet der mathematischen Logik, das Rekursiontheorie oder auch Berechenbarkeitstheorie genannt wird, befasst sich mit den Begriffen „Entscheidbarkeit” und „effektive Aufzählbarkeit”.

Satz 4.27. *Aufzählbarkeitssatz. Sei Γ eine entscheidbare Formelmenge in einer effektiven Sprache. Dann ist die Menge der logischen Implikationen aus Γ $\{\varphi \mid \Gamma \models \varphi\}$ effektiv aufzählbar.*

Beweis. Weil die Sprache effektiv ist, sind die Menge der Formel die Menge A der logischen Axiome und die Menge der endlichen Beweise aus Γ entscheidbar.

Somit können wir eine effektive Liste der logischen Implikationen aus Γ herstellen, indem wir die Menge der Beweise aus Γ aufzählen und die letzte Formel jedes Beweises in unsere effektive Aufzählung aufnehmen. $\dashv$

Definition 4.28. Eine Theorie ist eine Satzmenge Γ. Genauer: Eine τ-Theorie oder auch $\mathscr{L}(\tau)$-Theorie ist eine Satzmenge in $\mathscr{L}(\tau)$.

Dies könnte man natürlich auch für andere Logiken als die Logik $\mathscr{L}$ erster Stufe betrachten, und dieses Spezialgebiet heißt allgemeine Modelltheorie (abstract model theory).

Definition 4.29. $\Gamma \subseteq \mathscr{L}(\tau)$ heißt *vollständig* in $\mathscr{L}(\tau)$ gdw für jeden Satz $\varphi \in \mathscr{L}(\tau)$ gilt: $\Gamma \models \varphi$ oder $\Gamma \models \neg\varphi$.

Die meisten konsistenten Theorien sind nicht vollständig! Ohne Beweis geben wir hier Beispiele für vollständige Theorien:

1. In the Sprache $\mathscr{L}(\emptyset)$: die Theorie der unendlichen Mengen.
2. In der Sprache $\mathscr{L}(<)$: die Theorie der dichten offenen linearen Ordnungen.
3. In der Sprache $\mathscr{L}(+,\cdot)$: die Theorie der algebraisch abgeschlossenen Körper einer festen Charakteristik.

Definition 4.30. Sei $\mathfrak{M}$ eine τ-Struktur. $Th(\mathfrak{M}) = \{\varphi \in \mathscr{L}(\tau) \mid \varphi \text{ Satz und } \mathfrak{M} \models \varphi\}$ heißt *die Theorie von* $\mathfrak{M}$ (in der Sprache erster Stufe).

4. Die Theorie jeder τ-Struktur $\mathfrak{M}$ ist vollständig. (Genau dann, wenn $\mathfrak{M}$ endlich ist, beschreibt die Theorie von $\mathfrak{M}$ bis auf Isomorphie eindeutig, d.h., alle Elemente von $\mathrm{Mod}(Th(\mathfrak{M})) = \{\mathfrak{N} \mid \mathfrak{N} \models Th(\mathfrak{M})\}$ sind zu $\mathfrak{M}$ isomorph. Falls zusätzlich τ endlich ist, ist diese Theorie sogar endlich axiomatisierbar.)

Korollar 4.31. *Sei Γ eine entscheidbare Satzmenge einer effektiven Sprache. Sei Γ vollständig. Dann ist die Menge der Folgerungen aus Γ entscheidbar.*

Beweis: Nach dem Aufzählbarkeitssatz sind sowohl diese Menge als auch ihr Komplement (die Menge der Formeln φ, dass Γ die Formel $\neg\varphi$ beweist) effektiv aufzählbar. Jede effektiv aufzählbare Menge, die ein effektiv auszählbares Komplement hat, ist aber entscheidbar mit folgendem Verfahren: Wir können schrittweise abwechselnd die Menge und auch ihr Komplement aufzählen und darauf warten, dass ein gegebener Satz in einer der zwei Aufzählungen auftritt. $\dashv$

Unsere dritte Anwendung des Vollständigkeitssatzes betrifft die Größe (Mächtigkeit) von Modellen. Eine Struktur heißt abzählbar gdw ihr Universum abzählbar ist.

Satz 4.32. *Satz von Löwenheim und Skolem, Spezialfall für abzählbare Sprachen. Sei Γ eine konsistente Menge von Sätzen in einer abzählbaren Sprache. Dann hat Γ ein abzählbares Modell. Wenn Γ eine unendliches Modell hat, dann hat Γ auch ein überabzählbares Modell.*

Beweis. Des Beweis des Vollständigkeitssatzes zeigt, dass jede konsistente Satzmenge in einer abzählbaren Sprache ein abzählbares Modell hat.

Nun nehmen wir an, dass $\mathfrak{A}$ ein unendliches Modell von Γ ist. Wir nehmen eine überabzählbare Menge neuer Konstantensymbole $\{c_i \mid i < \aleph_1\}$ und erweitern Γ, indem wir für je zwei neue ungleiche Konstantensymbole c_i und c_j den Satz $c_i \neq c_j$ zu Γ hinzufügen.

Wir nennen diese erweiterte Satzmenge Γ^*. Jede endliche Teilmenge Γ_0^* von Γ^* ist erfüllbar, denn wir können in $\mathfrak{A}$ die endlich vielen in Γ_0 vorkommenden c_i durch endlich viele verschiedene Punkte interpretieren und so ein Modell Γ_0^* erhalten. Nach der Version des Vollständigkeitssatzes für überabzählbare Formelmengen — wenn Sie misstrauisch sind, da in der Vorlesung nur die abzählbare Version bewiesen wurde, lesen Sie in einem Mengenlehrebuch über transfinite Induktion nach — hat Γ^* ein Modell $\mathfrak{B}^* = (\mathfrak{B}, (c_i^{\mathfrak{B}^*})_{i<\aleph_1})$. Wenn wir die nun $\mathfrak{B}^*$ auf die Sprache τ beschränken, dann haben wir ein überabzählbares Modell $\mathfrak{B}$ von Γ. $\dashv$

Der Satz von Löwenheim und Skolem hat überraschende Konsequenzen. Zum Beispiel haben die natürlichen Axiome für die Struktur $\mathfrak{N} = (\mathbb{N}, 0, 1, +, \cdot)$ der Arithmetik auch ein überabzählbares Modell. Wenn wir hingegen in einer abzählbaren Sprache Axiome für eine große Struktur wie das Mengenuniversum hinschreiben, dann haben unsere Axiome, wenn sie konsistent sind, notwendig ein abzählbares Modell (dieser Sachverhalt heißt das Skolem'sche Paradoxon).

Diese Sachverhalte zeigen eine Schwäche der Sprache der ersten Stufe: Keine Axiomenmenge (auch nicht in einer überabzählbaren Symbolmenge) kann eine gegebene unendliche Struktur bis auf Isomorphie eindeutig charakterisieren.

Kapitel 5

Die Gödel'schen Unvollständigkeitssätze

In diesem Kapitel werden wir einige konkrete nicht rekursive Mengen sehen. Unser erstes Beispiel ist das Halteproblem. Im ersten Gödel'schen Unvollständigkeitssatz zeigen wir, dass für genügend reiche Symbolmengen τ die Menge der allgemeingültigen Sätze in $\mathscr{L}(\tau)$ nicht rekursiv ist. Dann betrachten wir die Zahlentheorie und einige rekursiv axiomatisierbare Teiltheorien. Zum Schluss geben wir einen recht vollständigen Beweis des zweiten Gödel'schen Unvollständigkeitssatzes.

5.1 Die Unentscheidbarkeit des Halteproblems für Turingmaschinen

Wir kehren zu Turingmaschinen zurück, um auf schnelle Weise eine unentscheidbare Menge herzustellen. Wir lassen nun auch Turingmaschinen zu, die auf manchen Inputs womöglich niemals stoppen. Diese Erweiterung hat starke Auswirkungen.

Definition 5.1. *M akzeptiert im weiteren Sinne das Wort w* gdw M angesetzt auf w stoppt. D.h., nach endlich vielen Anwendungen von δ auf die Anfangskonfiguration (q_0, w) wird eine Konfiguration mit Zustand in F, dies ist die Menge der akzeptierenden Zustände, erreicht.

In diesem weiteren Sinn haben wir also acuch: M akzeptiert w nicht, wenn (M, w) nicht stoppt, oder in einem Nicht-Akzeptierungszustand stoppt.

Statt im weiteren Sinn sagt man auch " im r.a. oder im r.e. Sinn" r.a. [r.e.] steht für rekursiv aufzählbar, recursively enumerable, wie in Definition 5.3.

Definition 5.2. Die Menge der von M im weiteren Sinne akzeptierten Wörter, traditionell auch Sprache von M genannt, ist die folgende Menge

$$L(M) = \{w \in \Sigma^* \mid M \text{ akzeptiert } w\}.$$

Nun folgt eine der wichtigsten Definitionen der Mathematik. Gödel 1931, Church, Turing, 1936.

Definition 5.3. $L \subseteq \Sigma^*$ heißt *rekursiv aufzählbar*, wenn es eine nicht notwendigerweise stoppende Turingmaschine M gibt, die genau $L = L(M)$ im weiteren Sinne akzeptiert.

Satz 5.4. *$L \subseteq \Sigma^*$ ist entscheidbar, gdw es eine Turingmaschine die L im weiteren Sinne akzeptiert und eine Turingmaschine gibt, die $\Sigma^* \setminus L$ im weiteren Sinne akzeptiert.*

Beweis: Die Vorwärtsrichtung folgt daraus, dass Komplemente entscheidbarer Mengen entscheidbar sind. Für die Rückrichtung sei $L(M_1) = L$ und $L(M_2) = \Sigma^* \setminus L$. Wir lassen M_1 und M_2, beide angesetzt auf w, abwechselnd einen Schritt ausführen. Genau eine der Maschinen akzeptiert w im weiteren Sinne. Es ist nun leicht, M_1 und M_2 zu einer Turingmaschine zusammenzusetzen, die immer stoppt und genau L akzeptiert. ⊣

Bemerkung 5.5. Es gibt eine *Aufzählungsmaschine* A_M von $L(M)$: A_M gibt im Akzeptierungszustand von M ein Wort aus, das in $L(M)$ aufgenommen wird, löscht die Eingabe und untersucht das nächste Wort mit M. Außerdem mischt A_M die Untersuchungen, da A_M bei einer nicht haltenden Untersuchung ja nicht abstürzen soll.

Definition 5.6. Gödelnummern in einem kleinen festen Alphabet für alle Turingmaschinen mit beliebigem endlichen Γ, Q.

(1) Es sei $Q = \{q_i \,|\, i \leq n\}$, q_0 Anfangszustand. Es sei $\Sigma = \{x_i \,|\, 1 \leq i \leq m'\}$ das Eingabealphabet. Es sei $\Gamma = \{b = x_0, x_1, x_2, \dots, x_m\} \supseteq \Sigma$ das Arbeitsalphabet. Wir nehmen an, dass keines des q_i oder x_j ein $*$ ist. Dies können wir durch Umbenennung immer erreichen.

    Hier steht b für ein unbeschriftetes Feld auf einem Turingband.

    Wir setzen
$$\Sigma_U = \{, , *, (,), L, R\}.$$

(2) Nun definieren wir ein Codewort für M, auch *Gödelnummer von M* genannt, in Σ_U: Wir schreiben $*^{(i)}$ für das Wort $\underbrace{* \cdots *}_{i}$. Sei $B = x_0$. Für eine Zeile der Tafel δ, $\delta(q_i, x_j) = (q_{i'}, x_{j'}, R)$, schreiben wir
$$(*^{(i+1)}, *^{(j+1)}, *^{(i'+1)}, *^{(j'+1)}, R)$$
    und wir schreiben die Tupel in δ in irgendeiner Anordnung einfach hintereinander und nennen dieses Σ_U-Wort $\#\delta$.

    Es sei $M = (Q, \Sigma, \Gamma, \delta, q_0, q_{\text{ak}}, q_{\text{ab}})$ eine Turingmaschine (siehe Definition 2.1) mit obigem Q, Σ, Γ, q_0. Wir nehmen an, dass $q_k = q_{\text{ak}}$ und $q_\ell = q_{\text{ab}}$.

    Wie definieren
$$\#M = (*^{(n+1)}, *^{(m')}, *^{(m+1)}, \#\delta, *^{(k+1)}, *^{(\ell+1)}).$$
    und erhalten so $\#M \in \Sigma_U^*$, die *„Gödelnummer"* von M.

Wir führen die Idee einer einheitlichen Kodierung noch weiter, um alle Turingmaschinen (in jeglichen endlichen Alphabeten) und alle endlichen Bandinschriften zu erfassen:

Definition 5.7. Codewert für Wörter über $\Gamma = \Sigma_U \cup \{x_i \mid i \leq m\} \cup \{q_i \mid i \leq n\}$:

$$\begin{aligned} \text{lettercode}(x_i) &= (*^{(i+1)}) \text{ für } x_i \in \Gamma \cup Q \setminus \Sigma_U \\ \text{lettercode}(X) &= X \text{ für } X \in \Sigma_U \\ \text{wordcode}(X_0 \dots X_{k-1}) &= \text{lettercode}(X_0) \dots \text{lettercode}(X_{k-1}) \text{ für } X_i \in \Gamma \cup Q \setminus \Sigma_U. \end{aligned}$$

Diese Codewerte sind also in der Menge $(\Sigma_U)^*$.

Definition 5.8. Das *Halteproblem* ist die folgende Menge:

$$H = \{(\#M, \text{wordcode}(w)) \in \Sigma_U^* \mid M \text{ ist eine TM und } M \text{ akzeptiert } w\}.$$

Definition 5.9. Sei für $u \in \Sigma_U^*$, M_u die Turingmaschine, deren Tafel durch das Wort u gegeben ist. Eine Turingmaschine M heißt *universell*, falls sie für alle u, w, auf Eingabe von u, w hin wie M_u angesetzt auf w arbeitet.

Lemma 5.10. *Es gibt eine universelle Turingmaschine.*

Satz 5.11. *Das Halteproblem ist Turing-aufzählbar.*

Satz 5.12. *Das Halteproblem H ist nicht Turing-entscheidbar. Genauer: Das Komplement $\Sigma_U^* \setminus H = H^c$ des Halteproblems ist nicht rekursiv aufzählbar.*

Beweis: Annahme es gibt eine TM M, die das Komplement des Halteproblems, $H^c = \Sigma_U^* \setminus H$, aufzählt. Es sei M' eine Abwandlung von M, die eine Eingabe w genau dann akzeptiert, wenn M die Eingabe (w, w) akzeptiert. Dann gilt:

$$M' \text{ akzeptiert } \#M' \Leftrightarrow M \text{ akzeptiert } (\#M', \#M'),$$

$$M \text{ akzeptiert } (\#M', \#M') \Leftrightarrow (\#M', \#M') \in H^c,$$

$$(\#M', \#M') \in H^c \Leftrightarrow M' \text{ akzeptiert } \#M' \text{ nicht.}$$

Dabei gilt das erste $\Leftrightarrow$ nach der Definition von M', die zweite Äquivalenz gilt nach der Annahme von M als Aufzähler von H^c. Das dritte $\Leftrightarrow$ folgt aus der Definition von H^c als die Menge der M, die auf Input $\#M$ hin nicht halten und daher auch nicht akzeptieren. $\dashv$

5.2 Der erste Gödel'sche Unvollständigkeitssatz

Sei T eine entscheidbare Satzmenge in einer effektiven Sprache ersten Stufe. Wir haben gesehen, dass die Menge der Sätze, die aus T logisch folgen, effektiv aufzählbar ist. Ist sie auch entscheidbar?

Wir beginnen auf der spielerischen Seite der Kunst mit einem Gedicht von Hans Magnus Enzensberger:

Hommage à Gödel

Münchhausens Theorem, Pferd, Sumpf und Schopf,
ist bezaubernd, aber vergiß nicht:
Münchhausen war ein Lügner.

Gödels Theorem wirkt auf den ersten Blick
etwas unscheinbar, doch bedenk:
Gödel hat Recht.

„In jedem genügend reichhaltigen System,
lassen sich Sätze formulieren,
die innerhalb des Systems
weder beweisbar noch widerlegbar sind,
es sei denn, das System
wäre selber inkonsistent."

Du kannst deine eigene Sprache
in deiner eigenen Sprache beschreiben:
aber nicht ganz.
Du kannst dein eigenes Gehirn
mit deinem eigenen Gehirn erforschen,
aber nicht ganz.
Usw.

Um sich zu rechtfertigen
muß jedes denkbare System
sich transzendieren,
d.h. zerstören.

„Genügend reichhaltig" oder nicht:
Widerspruchsfreiheit
ist eine Mangelerscheinung
oder ein Widerspruch.

(Gewißheit = Inkonsistenz.)

Jeder denkbare Reiter,
also auch Münchhausen,
also auch du bist ein Subsystem
eines genügend reichhaltigen Sumpfes.

Und ein Subsystem dieses Subsystems
ist der eigene Schopf, dieses Hebezeug,
für Reformisten und Lügner.

In jedem genügend reichhaltigen System,
also auch in diesen Sumpf hier
lassen sich Sätze formulieren
die innerhalb des Systems
weder beweis- noch widerlegbar sind.

Diese Sätze nimm in die Hand
und zieh.

In diesem Teil der Vorlesung beschreiben wir eine einfache endliche Menge A_E von Sätzen in der Sprache $\{+, \cdot, E, 0, 1\}$ der Arithmetik mit einem zweistelligen Funktionssymbol E für die Exponentiation und zeigen, dass die Menge der logischen Folgerungen aus jeder konsistenten Obermenge von A_E unentscheidbar ist. Dann zeigen wir, dass genügend Eigenschaften der Exponentiation schon aus der Theorie Q folgen. Zusammen mit dem Korollar 4.31 über vollständige entscheidbare Satzmengen erhalten wir dann, dass keine entscheidbare konsistente Obermenge von Q vollständig ist. Dieses Korollar ist sehr wichtig und wird *Erster Gödel'scher Unvollständigkeitssatz* genannt.

Ein mathematisch exakter Beriff von Entscheidbarkeit heißt „Rekursivität" und wurde im Kapitel über Turingmaschinen definiert.

Eine *Theorie* ist eine Menge von Sätzen in einer Sprache erster Stufe. Sei T eine Theorie in einer Sprache, die das Konstantensymbal 0 und das Funktionssymbol S für eine einstellige Funktion enthält. Wir verwenden den Term S^n0 für die natürliche Zahl n und schreiben einfach n. Außerdem schreiben wir $\varphi(n_1, \ldots, n_k)$ für $(\cdots((\varphi(v_1, \ldots, v_k))^{v_1}_{n_1})^{v_2}_{n_2}) \cdots)^{v_k}_{n_k}$.

Definition 5.13. Es sei $R \subseteq \mathbb{N}^k$. Eine Formel φ *stellt R in T dar* gdw für alle $_1, \ldots, n_k \in \mathbb{N}$ folgendes gilt:

$$\begin{aligned} &\langle n_1, \ldots, n_k\rangle \in R \text{ gdw } T \vdash \varphi(n_1, \ldots, n_k), \text{ und} \\ &\langle n_1, \ldots, n_k\rangle \notin R \text{ gdw } T \vdash \neg\varphi(n_1, \ldots, n_k). \end{aligned} \tag{5.1}$$

R heißt darstellbar in T gdw es eine Formel φ gibt, die R in T darstellt.

Satz 5.14. *Eine k-stellige Relation $R \subseteq \mathbb{N}^k$ ist rekursiv gdw R in einer endlichen Theorie T darstellbar ist.*

Beweis: Durch abwechselndes Berechnen eines Schrittes einer Herleitung $T \vdash \varphi$ und eines Schrittes einer Herleitung von $T \vdash \neg\varphi$ erhält man die Richtung $\Leftarrow$ im folgenden Satz. Durch Übersetzung der Turingtafel, die die Rekursivität bezeugt, in die Sprache erster Stufe erhält man eine darstellende Formel, die die Richtung $\Rightarrow$ beweist. $\dashv$

Wir definieren nun eine endliche Theorie A_E, die einige der Eigenschaften des Standardmodells

$$\mathfrak{N}_E = (\mathbb{N}, 0, 1, +, \cdot, E, S, \leq)$$

(mit der Nachfolgerfunktion S, der Kleinergleichrelation $\leq$ und der Exponentiation E) der Arithmetik beschreibt. Wir zeigen, dass jede rekursive Relation in A_E darstellbar ist.

Danach gehen wir zu dem bekannteren Standardmodell

$$\mathfrak{N} = (\mathbb{N}, 0, 1, +, \cdot, S, \leq)$$

zeigen wir analoge Resultate ohne die Exponentiation, für die endlich axiomatisierbare Theorie $Q \subseteq Th(\mathfrak{N})$ und die nicht endlich axiomatisierbare, dafür aber in Σ_1-Sätzen (Def. 5.28) axiomatisierbare Theorie Q^*.

Aus diesen Tatsachen und aus einigen Lemmata erhalten wir danach recht einfach die gewünschten Ergebnisse über Unentscheidbarkeit und Unvollständigkeit.

Definition 5.15. Wir schreiben für $n \in \mathbb{N}$, $S^{(n)}$ für die n-fach iterierte Anwendung von S, also $S^{(0)}x = x$ und $S^{(n+1)}x = S^{(n)}Sx$.

(1) Die Axiome von A_E:

S1. $\forall x Sx \neq 0$.
S2. $\forall x \forall y (Sx = Sy \to x = y)$.

L1. $\forall x \forall y (x < Sy \leftrightarrow x \leq y)$.
L2. $\forall x (x \not< 0)$.
L3. $\forall x \forall y (x < y \vee x = y \vee y < x)$.

A1. $\forall x (x + 0 = x)$.
A2. $\forall x \forall y (x + Sy = S(x + y))$.

M1. $\forall x (x \cdot 0 = 0)$.
M2. $\forall x \forall y (x \cdot Sy = x \cdot y + x)$.

E1. $\forall x (xE0 = S0)$.
E2. $\forall x \forall y (xESy = (xEy) \cdot x)$.

(2) Die Axiome der *Robinson'schen Theorie* Q sind A_E ohne E1 und E2, also in der Sprache $\{0, 1, +, \cdot, S, \leq\}$, abgefasst.

(3) Die Axiome der *Cobham'schen Theorie* Q^* sind
Q^*_1. Für alle $n, m \in \mathbb{N}$: $S^{(n)}0 + S^{(m)}0 = S^{(n+m)}0$.
Q^*_2. Für alle $n, m \in \mathbb{N}$, $S^{(n)}0 \cdot S^{(m)}0 = S^{(n \cdot m)}0$.
Q^*_3. Für alle $n \in \mathbb{N}$: $(\forall x)(x < S^{(n)}0 \leftrightarrow (x = S^{(0)}0 \vee \cdots \vee x = S^{(n-1)}0))$.

Jeder Satz aus A_E und somit jede Folgerung aus A_E ist im Standardmodell $\mathfrak{N}_E$ von A_E wahr. Jedoch ist nicht jeder Satz, der in $\mathfrak{N}_E$ wahr ist, eine Folgerung aus A_E. Denn jede Theorie einer festen Struktur ist vollständig. Wir werden sehen, dass A_E in einem starken Sinne unvollständig ist.

Man zeigt induktiv über den Aufbau der quantorenfreien Sätze: Quantorenfreie Sätze, die in $\mathfrak{N}_E/\mathfrak{N}$ wahr sind, können aus A_E/Q (und sogar aus Q^*) gefolgert (=formal bewiesen) werden.

5.3 Gödelnummern

Wir zeigen, dass jede rekursive Relation nicht nur in einer geeigneten endlichen Theorie, sondern auch in A_E darstellbar ist. Wie können wir zeigen, dass jede Relation auf den natürlichen Zahlen, die unter Verwendung der Beweisbarkeit aus einer endlichen Theorie definiert wird, in einer Theorie der Arithmetik, wie zum Beispiel in A_E, dargestellt werden kann?

Die Lösung dieses Problems wird durch die Verwendung von Gödelnummern geliefert. Diese sind Kodenummern in $\mathbb{N}$ für die zentralen syntaktischen Objekte der Logik: Symbole, Terme, Formeln und Beweise aus einer endlichen VOraussetzungsmenge T (im Sinne der Definition von $\vdash$). Wir werden zeigen, dass die Menge der natürlichen Zahlen, die Beweise der Logik ersten Stufe kodieren, in A_E darstellbar ist. Aus diesem Resultat folgt die Darstellbarkeit beliebiger rekursiver Relationen in A_E.

Zuerst beschreiben wir eine Funktion h, die jedem Symbol aus einer gegebenen Sprache erster Stufe eine Kodenummer zuweist: Für die logischen Symbole wird h wie folgt definiert Die Zahlen 1, 3, 5, 7, 9 sind die Werte von h für die Symbole (,), $\neg$, $\rightarrow$, $=$. Für die Variablen sei $h(v_n) = 9 + 2n + 2$. Wir setzen $h(\forall) = 0$ und verwenden die anderen geraden Zahlen, um die anderen nichtlogischen Symbole in der effektiven Sprache (=Symbolmenge) zu kodieren. Obwohl unser zentrales Interesse den Sprachen mit endlich vielen nichtlogischen Symbolen gilt, nehmen wie lediglich an, dass unsere Symbolmenge rekursiv ist. Dies bedeutet, dass die folgende Mengen in A_E darstellbar sind:

$$\{k \mid k \text{ ist } h(c) \text{ für ein Konstantensymbol } c\},$$
$$\{\langle k, n\rangle \mid k \text{ ist } h(P) \text{ für ein } n\text{-stelliges Prädikatensymbol } P\},$$
$$\{\langle k, n\rangle \mid k \text{ ist } h(f) \text{ für ein } n\text{-stelliges Funktionssymbol } f\}.$$

Dies gilt offensichtlich, wenn diese Mengen endlich sind.

Für eine endliche Folge $s_0, \ldots, s_n$ von Symbolen unserer Sprache $\mathscr{L}(\tau)$ (diese Folge kann ein sinnvoller Ausdruck der Sprache sein oder einfach nur eine Zeichenreihe) definieren wir die Gödelnummer $\ulcorner s_0, \ldots, s_n \urcorner$ wie folgt. Sei $p_0, p_1 \ldots$ die streng monotone Aufzählung der Primzahlen.

$$\begin{aligned} \ulcorner s_0, \ldots, s_n \urcorner &= \langle\langle h(s_0), \ldots, h(s_n)\rangle\rangle \text{ mit} \\ \langle\langle k_0, \ldots, k_n\rangle\rangle &= 2^{k_0+1} \cdot \cdots \cdot p_n^{k_n+1}. \end{aligned}$$

Wenn Φ eine Menge von endlichen Folgen von Symbolen ist, dann schreiben wir $\ulcorner \Phi \urcorner$ für die Menge $\{\ulcorner \varepsilon \urcorner \mid \varepsilon \in \Phi\}$. Wir brauchen auch Kodenummern für Beweise: Wenn $\langle \alpha_0, \ldots, \alpha_n\rangle$ eine endliche Folge von Formeln ist, dann setzen wir

$$\ulcorner \langle \alpha_0, \ldots, \alpha_n\rangle \urcorner = \langle\langle \ulcorner \alpha_0 \urcorner, \ldots, \ulcorner \alpha_n \urcorner \rangle\rangle.$$

Nun muss man zeigen, dass jeder syntaktische Begriff unserer Sprache erste Stufe in A_E darstellbar ist, wenn er in Gödelnummern übersetzt wird. Im folgenden sagen wir kurz „darstellbar” anstelle von „darstellbar in A_E”.

Warnung: Ab hier ist der Rest des Kapitels nur eine Skizze. Die Durchführung der ausgelassenen Beweise ist nicht schwer, würde jedoch etliche Sitzungen dauern.

Lemma 5.16. 1. *Die Menge der Gödelnummern für Variablen ist darstellbar.*

2. *Die Menge der Gödelnummern für Terme ist darstellbar.*
3. *Die Menge der Gödelnummern für atomare Formeln ist darstellbar.*
4. *Die Menge der Gödelnummern für Formeln ist darstellbar.*
5. *Sei* ers *die folgende dreistellige Funktion:* $\mathrm{ers}(\ulcorner \alpha \urcorner, \ulcorner x \urcorner, \ulcorner t \urcorner) = \ulcorner \alpha^x_t \urcorner$ *für Formeln* α*, Terme* t *und Variablen* x*, und sonst undefiniert. Dann ist der Graph von* ers *eine darstellbare vierstellige Relation.*
6. *Die Funktion, die* n *auf die Gödelnummer* $\ulcorner n \urcorner$ *abbildet, ist darstellbar.*

7. *Sei* frei *die folgende zweistellige Relation:* $\langle \ulcorner \alpha \urcorner, \ulcorner x \urcorner \rangle \in$ frei *gdw* x *in* α *frei auftritt. Dann ist* frei *darstellbar.*
8. *Die Menge der Gödelnummern für Sätze ist darstellbar.*
9. *Sei* ein *die folgende dreistellige Relation:* $\langle \ulcorner \alpha \urcorner, \ulcorner x \urcorner, \ulcorner t \urcorner \rangle \in ein$ *gdw in* α t *für* x *eingesetzt werden kann. Dann ist* ein *darstellbar.*
10. *Sei* all *die folgende zweistellige Relation:* $\langle \ulcorner \alpha \urcorner, \ulcorner \beta \urcorner \rangle \in$ all *gdw* β *eine Verallgemeinerung von* α *ist. Dann ist* all *darstellbar.*
11. *Die Menge der Gödelnummern für logische Axiome ist darstellbar.*
12. *Sie* A *eine Menge von Formeln, so dass* $\ulcorner A \urcorner$ *darstellbar ist. Dann ist* $\{\ulcorner D \urcorner \mid D$ *ist ein Beweis aus* $A\}$ *darstellbar.*
13. *Jede rekursive Relation ist darstellbar in* A_E*.*

Beweis: 1. Dies ist die Menge $\{a \mid \exists b < a(a = \langle\langle 11 + 2b \rangle\rangle)\}$.

13. Sei R eine einstellige rekursive Relation. Dann ist R durch eine Formel φ in einer endlichen Theorie T darstellbar. Wir definieren $f(n) =$ das kleinste d, so das $d = \ulcorner D \urcorner$ für einen Beweis D von $\varphi(S^n 0)$ oder $\neg\varphi(S^n 0)$ aus T ist. Dann ist f darstellbar in A_E: Für $\ulcorner \varphi(S^n 0) \urcorner = k_m$ ist $f(n) = \langle\langle k_0, \dots, k_m \rangle\rangle$, falls $T \vdash \varphi(S^n 0)$, und sonst gilt Analoges mit $\neg\varphi(S^n 0)$. Daher ist R in A_E darstellbar. Dasselbe Argument gilt für n-stellige Relationen.

Satz 5.17 (Der starke Unentscheidbarkeitssatz)**.** *Sie T eine Theorie, so dass $T \cup A_E$ konsistent ist. Dann ist die Menge*

$$\{\ulcorner \varphi \urcorner \mid \varphi \text{ ist ein Satz und } T \vdash \varphi\}$$

nicht rekursiv.

[1]

Beweis: Für jedes n definieren wir die Formel φ_n wie folgt: Wenn n die Gödelnummer einer Formel φ ist, die nur eine freie Variable v_1 hat, dann setzen wir $\varphi_n = \varphi$. Im anderen Fall nehmen wir für φ_n die Formel $v_1 = v_1$.

Schließlich setzen wir

$$R_n = \{k \mid T \cup A_E \vdash \varphi_n(S^k 0)\}.$$

Sei R eine rekursive Teilmenge von $\mathbb{N}$. Dann gibt es nach unserer Definition von Rekursivität und nach dem Punkt 13 des obigen Lemmas eine Formel $\varphi_R(v_1)$ so dass:

$$\begin{aligned} k \in R &\text{ gdw } A_E \vdash \varphi_R(S^k 0), \\ k \notin R &\text{ gdw } A_E \vdash \neg\varphi_R(S^k 0). \end{aligned}$$

Dann gilt natürlich erst recht:

$$\begin{aligned} k \in R &\text{ gdw } T \cup A_E \vdash \varphi_R(S^k 0), \\ k \notin R &\text{ gdw } T \cup A_E \vdash \neg\varphi_R(S^k 0). \end{aligned}$$

[1] Ab hier bis Ende des Kapitels im Wintersemerster 2022/23 nicht vorgetragen.

Nun setzen wir $\ulcorner\varphi_R(v_1)\urcorner = n$ und erhalten $R = R_n$. R_n, $n \in \omega$, ist also eine Auzählung aller rekursiven einstelligen Relationen.

Wir nehmen nun indirekt an, dass $\{\ulcorner\varphi\urcorner \mid \varphi \text{ ist ein Satz und } T \vdash \varphi\}$ rekursiv ist. Dann ist auch $\{\ulcorner\varphi\urcorner \mid \varphi \text{ ist ein Satz und } T \vdash \bigwedge A_E \to \varphi\}$ rekursiv. Somit ist $\{\ulcorner\varphi\urcorner \mid \varphi \text{ ist ein Satz und } T \cup A_E \vdash \varphi\}$ rekursiv. Nun schränken wir dies ein auf bestimmte Sätze und erhalten:

$$\{\langle k, n\rangle \mid T \cup A_E \vdash \varphi_n(S^k 0)\}$$

ist rekursiv. Nach unserer Definition der Relationen R_n ist also $\{\langle k, n\rangle \mid k \in R_n\}$ rekursiv. Es gibt also ein $\psi(v_1, v_2)$, das $\{\langle k, n\rangle \mid k \in R_n\}$ in A_E darstellt. Dann gibt es $\psi'(v_1) = \psi(v_1, v_1)$, so dass für alle n, $A_E \vdash \psi'(S^n 0) \leftrightarrow \varphi_n(S^n 0)$.

Doch nun definieren wir eine weitere Menge:

$$n \in R \text{ gdw } n \notin R_n.$$

Auch diese Menge R ist rekursiv, denn $n \in R$ gdw $T \cup A_E \nvdash \psi'(S^n 0)$.

Dieses ist aber ein Widerspruch, da sich im Punkte n die beiden Relationen R_n und R unterscheiden. $\dashv$

Korollar 5.18 (Starke Version des Gödel'schen Unvollständigkeitssatzes)**.** *Sei T eine rekursive Menge von Axiomen, und sei $T \cup A_E$ konsistent. Dann ist T unvollständig: Es gibt einen Satz φ, so dass $T \nvdash \varphi$ und $T \nvdash \neg\varphi$.*

Beweis: Nach dem Satz über die starke Unentscheidbarkeit ist die Menge $X = \{\ulcorner\varphi\urcorner \mid \varphi \text{ ist ein Satz und } T \vdash \varphi\}$ nicht rekursiv. Wenn T vollständig wäre, dann wäre aber X rekursiv.

Dies zeigt man wie folgt: Wenn T vollständig ist, dann definieren wir eine totale Funktion f wie folgt: $f(n)$ das kleinste d, so dass entweder n keine Gödelnummer eines Satzes ist oder aber $n = \ulcorner\varphi\urcorner$ für einen Satz φ und d von der Form $\ulcorner D\urcorner$ für einen Beweis D aus T von φ oder von $\neg\varphi$ ist. Die Funktion f ist rekursiv, da T eine rekursive Axiomenmenge hat. Nun haben wir $n \in X$ gdw n die Gödelnummer eines Satzes ist und $f(n)$ ein Beweis von φ ist und $\ulcorner\varphi\urcorner = n$ ist. Dies widerspricht der Nicht-Rekursivität von X. $\dashv$

Einige bekannte Spezialfälle des starken Unentscheidbarkeitssatzes sind: Sei $\mathfrak{N}_E = (\mathbb{N}, +, \cdot, E, 0, 1)$.

Korollar 5.19. *Sei $T = Th(\mathfrak{N}_E)$ die Zahlentheorie. Dann ist $\ulcorner T\urcorner$ nicht rekursiv.*

Beweis $T \cup A_E$ ist konsistent, da $A_E \subseteq T$. Und da es sich um eine unter $\models$ und unter $\vdash$ abgeschlossenen Menge handelt, ist $T = \{\varphi \mid \varphi \text{ ist ein Satz und } T \vdash \varphi\}$. Deshalb folgt das Ergebnis aus dem starken Unentscheidbarkeitssatz. $\dashv$

Korollar 5.20. *Sei S die Menge der allgemeingültigen Sätze in der Sprache der Arithmetik. Dann ist $\ulcorner S\urcorner$ nicht rekursiv.*

Beweis: Dies folgt aus dem starken Unentscheinbarkeitssatz, weil $S \cup A_E$ offensichtlich konsistent ist. $\dashv$

Nun können wir diese Ergebnisse verstärken. Erinnern Sie sich daran, dass $A \subseteq \mathbb{N}$ definierbar in $\mathfrak{N}_E$ ist, gdw es eine Formel φ gibt, so dass $k \in A$ gdw $\mathfrak{N}_E \models \varphi(k)$.

Ein rekursives A ist definierbar in $\mathfrak{N}_E$, da jede Formel φ die A in A_E darstellt, A auch in $\mathfrak{N}_E$ definiert. Dei Umkehrung ist falsch: als Beispiel ist die Menge $\ulcorner\{\varphi \mid \varphi$ ist ein Satz und $A_E \vdash \varphi\}\urcorner$ nicht rekursiv aber definierbar in $\mathfrak{N}_E$.

Wie haben gesehen, dass $\ulcorner Th(\mathfrak{N}_E)\urcorner$ nicht rekursiv ist. Da dies vollständig ist, haben wir sogar

Satz 5.21. *Undefinierbarkeitssatz.* $\ulcorner Th(\mathfrak{N}_E)\urcorner$ *ist nicht definierbar in* $\mathfrak{N}_E$.

Beweis: Dieser ist ähnlich wie der Beweis des starken Unentscheidbarkeitssatzes. Für jedes n sei φ_n die Formel mit der Gödelnummer n, wenn eine solche solche existiert und die freie Variable v_1 hat, sonst sei φ_n die Formel $v_1 = v_1$.

Sei $A_n = \{k \mid \mathfrak{N} \models \varphi_n(k)\}$. Dann ist $\{A_n \mid n \in \mathbb{N}\}$ eine Aufzählung aller in $\mathfrak{N}$ definierbaren Mengen. Nehmen wir nun an, dass $\ulcorner Th(\mathfrak{N})\urcorner$ in $\mathfrak{N}$ definierbar wäre. Dann wäre die Menge

$$A = \{n \mid n \notin A_n\}$$

auch definierbar in $\mathfrak{N}$. Da $n \in A$ gdw $\ulcorner\varphi_n(n)\urcorner \notin \ulcorner Th(\mathfrak{N})\urcorner$. Aber da A sich von allen A_n's unterscheidet, ist dies ein Widerspruch. $\dashv$

Die vorstehenden Ergebnisse haben wir in der Sprache der Arithmetik mit Exponentiation ausgedrückt. Sie sind aber auch gültig für die Arithmetik ohne Exponentiation.

Satz 5.22. *E ist definierbar in Q, g.h. es gibt eine Formel φ_E so dass für alle $n, m, u \in \mathbb{N}$ gilt $n^m = u$ gdw $Q \vdash \varphi_E(S^{(n)}0, S^{(m)}0, S^{(u)}0$.*

Beweis: Wir zeigen, dass Folgen kodierbar sind. a^b wird durch die Folge $(1, a, a^2, a^3, \ldots, a^b)$ gegeben. Deren Glieder erfüllen die Formel $a_0 = 1$ und $a_{n+1} = a_n \cdot a$, und a_b ist das gesuchte a^b. Wenn also die gesamte Folge in Q definierbar ist, dann ist auch die Exponentiationsfunktion definierbar.

Lemma 5.23. *Das Gödel'sche β-Lemma. Es gibt eine Funktion $\beta\colon \mathbb{N}^3 \to \mathbb{N}$ mit den folgenden Eigenschaften*

(a) *Für jede Folge $(a_0, \ldots, a_r)$ über $\mathbb{N}$ gibt es t, $p \in \mathbb{N}$, so dass für $i \leq r$, $\beta(t, p, i) = a_i$.*

(b) *Es gibt eine $\{0, 1, +, \cdot\}$-Formel $\chi(v_0, \ldots, v_3)$, die β definiert im folgenden Sinn:*

$$Q \vdash \chi(S^{(t)}0, S^{(p)}0, S^{(i)}0, S^{(a)}0) \text{ gdw } \beta(t, p, i) = a.$$

Beweis des Lemmas: Wir nehmen eine Primzahl $p \geq a_i$ für $i \leq r$ und $p \geq r + 1$ und setzen

$$t = 1 \cdot p + a_0 p^1 + 2p^2 + \cdots + (r + 1)p^{2r} + a_r p^{2r+1}.$$

t kodiert den Graphen von t in p-adischer Weise.

$$\beta(t,p,i) = a \text{ gdw } Q \vdash \chi(S^t(0), S^p(0), S^i(0), S^a(0)),$$

mit folgenden Formeln

$$\begin{aligned}\psi(v_0,\dots,v_3) = \exists v_4 v_5 v_6 (v_0 = v_4 + v_5((v_2+1) + v_3 v_1 + v_5 v_1^2) \\ \wedge\, v_3 < v_1 \wedge v_4 < v_5 \wedge \\ \exists v_7 (v_5 = v_7^2 \wedge \exists v_8 (\forall v_9 (v_9 \mid v_5 \to v_7 \mid v_9))))\end{aligned}$$

und $\chi(v_0, \dots v_3)$ sagt, dass v_3 das $<$-minimale Element ist, so dass $\psi(v_0,\dots,v_3)$ wahr ist. $\dashv$

Gödel zeigte das β-Lemma durch Verwendung des chinesischen Restsatzes. Eine Struktur $\mathfrak{A}$ heißt *entscheidbar* gdw $\ulcorner Th(\mathfrak{A}) \urcorner$ rekursiv ist. Wir wissen also nun, dass nicht nur die Struktur $\mathfrak{N}_E$ sondern auch die Struktur $\mathfrak{N}_M$ unentscheidbar ist.

Korollar 5.24. *Alle Sätze dieses Kapitels, auch diejenigen im nächsten Abschnitt, über A_E gelten auch für Q.*

Man kann zeigen, dass die Struktur

$$\mathfrak{N}_P = (N, 0, S, +, <)$$

entscheidbar ist, die *Presburger-Arithmetik*. Hieraus folgt insbesondere, dass die Multiplikation nicht in $\mathfrak{N}_P$ definierbar ist.

Wir können auch die verwandten Strukturen $(\mathbb{Z}, +, \cdot)$, $(\mathbb{Q}, +, \cdot)$, $(\mathbb{R}; +, \cdot)$ und $(\mathbb{C}, +, \cdot)$ der Mengen der ganzen, rationalen, reellen und komplexen Zahlen betrachten. Die Menge der natürlichen Zahlen ist in den ersten beiden Strukturen definierbar, daher sind diese unentscheidbar, Die letzten beiden Strukturen sind entscheidbar aufgrund tiefer liegender Ergebnisse aus der Algebra.

Die Kodierung der Exponentialfunktion E durch die Addition und die Multiplikation wie im β-Lemma hat kein Analogon über den reellen Zahlen. Bis heute ist die Entscheidbarkeit der Theorie von

$$(\mathbb{R}, +, \cdot, E, 0, <)$$

offen. Man kennt hinreichende Kriterien für eine bejahende Antwort, jedoch in der gegenteiligen Richtung ist nichts bekannt.

5.4 Der zweite Gödel'sche Unvollständigkeitssatz

Sei T eine rekursive Theorie, die A_E enthält. Wegen userer obigen Ergebnisse wissen wir, dass

$$\{ \ulcorner D \urcorner, \ulcorner \varphi \urcorner) \mid D \text{ ist ein Beweis von } \varphi \text{ aus } T\}$$

rekursiv ist und somit durch eine Formel $\mathrm{Bew}_T(v_1, v_2)$ darstellbar in A_E ist. Nun betrachten wir den Satz $\neg\exists d(\mathrm{Bew}_T(d, \ulcorner 0 = 1\urcorner))$. Dieser Satz der Arithmetik drückt die Eigenschaft „T ist konsistent" in der ersten Stufe aus, und wir schreiben ihn als Kon_T. Wir skizzieren nun die Beweisidee für den Beweis der Tatsache, dass für alle genügend starken Theorien T der Satz Kon_T nicht aus T beweisbar ist. Zunächst betrachten wir die folgende allgemeine Tatsache

Satz 5.25. *Fixpunktsatz. In der Sprache der Arithmetik gibt es für jede Formel β mit nur einer freien Variablen einen Satz σ, so dass A_E den Satz $\sigma \leftrightarrow \beta(\ulcorner\sigma\urcorner)$ beweist.*

Beweis. Sei f die rekursive Funktion, die jedem Paar $(\ulcorner\alpha\urcorner, n)$ den Wert $\ulcorner\alpha(n)\urcorner$ zuweist. Stelle $\theta(v_1, v_2, v_3)$ f in A_E dar. Wir betrachten nun die Formel

$$\gamma(v_1) = \forall v_3(\theta(v_1, v_1, v_3) \to \beta(v_3)).$$

Sei $q = \ulcorner\gamma\urcorner$. Dann erfüllt der Satz

$$\sigma = \gamma(q) = \forall v_3(\theta(q, q, v_3) \to \beta(v_3))$$

das Gewünschte: Weil $\theta(v_1, v_2, v_3)$ die obige Funktion f darstellt und $\ulcorner\sigma\urcorner$ der Wert dieser Funktion an der Stelle (q, q) ist, gilt

$$A_E \vdash \forall v_3(\theta(q, q, v_3) \leftrightarrow v_3 = \ulcorner\sigma\urcorner).$$

Wir setzen

$$\sigma = \forall v_3(\theta(q, q, v_3) \to \beta(v_3)). \tag{5.2}$$

Sei $A_E \vdash \sigma$. Dann $A_E \vdash \forall v_3(\theta(q, q, v_3) \to \beta(v_3))$. Wir setzen $v_3 = \ulcorner\sigma\urcorner$ ein. $A_E \vdash \theta(q, q, \ulcorner\sigma\urcorner) \to \beta(\ulcorner\sigma\urcorner)$. Also erhalten wir $A_E \cup \{\sigma\} \vdash \beta(\ulcorner\sigma\urcorner)$. $A_E \vdash \sigma \to \beta(\ulcorner\sigma\urcorner)$.

Umgekehrt
$A_E \vdash \beta(\ulcorner\sigma\urcorner) \to (\forall v_3 \theta(q, q, v_3) \to \beta(v_3))$, da

$$A_E \vdash \forall v_3(\theta(q, q, v_3) \leftrightarrow v_3 = \ulcorner\sigma\urcorner).$$

Nun setzen wir dies zusaätzlich zu $A_E \Vdash \beta(\ulcorner\sigma\urcorner)$ als aus A_E beweisbar ein und erhalten $A_E \vdash \beta(\ulcorner\sigma\urcorner) \to (\forall v_3 \theta(q, q, v_3) \to v_3 = \ulcorner\sigma\urcorner \wedge \beta(v_3))$ und lassen dann die Gleichung $v_3 = \ulcorner\sigma\urcorner$ weg. Nach der Definition von σ in (5.2),
$A_E \vdash \beta(\ulcorner\sigma\urcorner) \to \sigma$.

Somit ist $A_E \vdash \sigma \leftrightarrow \beta(\ulcorner\sigma\urcorner)$, wie gewünscht. $\dashv$

Nehmen wir nun an, dass T eine Theorie und $\ulcorner T\urcorner$ rekursiv ist.

Definition 5.26. Der Lügnersatz σ. Sei $\beta(v_2)$ die Formel $\neg\exists v_1 \mathrm{Bew}_T(v_1, v_2)$. Wir wenden den Fixpunktsatz auf β an und erhalten dadurch einen Satz σ, so dass $A_E \models \sigma \leftrightarrow \beta(\ulcorner\sigma\urcorner)$.

„Ich bin wahr genau dann, wenn es keinen Beweis für mich gibt."

Dann sagt σ, dass σ aus T unbeweisbar ist:

Lemma 5.27. *Wenn T rekursiv und konsistent ist und $T \supseteq A_E$, dann $T \not\vdash \sigma$.*

Beweis: Wir nehmen an, dass T σ beweist. Dann sei D ein solcher Beweis von σ aus T und sei d seine Gödelnummer. Dann ist $A_E \vdash \mathrm{Bew}_T(d, \ulcorner\sigma\urcorner)$. Deshalb gilt $A_E \vdash \neg\beta(\ulcorner\sigma\urcorner)$, daher $A_E \vdash \neg\sigma$. Weil T die Theorie A_E enthält, also $T \vdash \neg\sigma$. Somit beweist T sowohl σ als auch $\neg\sigma$, und deshalb ist T inkonsistent. $\dashv$

Definition 5.28. Eine $S_{\mathrm{ar}} = \{0, 1, +, \cdot, S, \leq\}$-Formel φ heißt Σ_1-*Formel*, wenn sie nur Existenzquantoren und $\forall$-Quantoren der Form $\forall x < S^n(0)$ für geeignete $n \in \mathbb{N}$ (sogenannte *beschränkte* $\forall$-*Quantoren*) enthält.

Definition 5.29. Sei T eine S_{ar}-Theorie. $\mathrm{Bew}_T(x)$ ist eine *gute T-Beweisbarkeitsformel* gdw die folgenden Bedingungen erfüllt sind:

(1) $\mathrm{Bew}_T(x)$ ist Σ_1, und $\mathrm{Bew}_T(n)$ ist wahr gdw es einen Satz ψ gibt, so dass $n = \ulcorner\psi\urcorner$ und $T \vdash \psi$.

(2) Für Σ_1-Sätze ψ: $T \vdash (\psi \to \mathrm{Bew}_T(\ulcorner\psi\urcorner))$.

(3) Für alle Sätze ψ und γ: $T \vdash (\mathrm{Bew}_T(\ulcorner\psi\urcorner) \wedge \mathrm{Bew}_T(\ulcorner(\psi \to \gamma))\urcorner) \to \mathrm{Bew}_T(\ulcorner\gamma\urcorner)$.

Definition 5.30. Sei T eine S_{ar}-Theorie. $(T, \mathrm{Bew}_T(x))$ erfüllt die *Löb-Axiome* gdw die folgenden Bedingungen erfüllt sind:

(L1) $T \vdash \varphi$ impliziert $T \vdash \mathrm{Bew}_T(\ulcorner\varphi\urcorner))$.

(L2) $T \vdash (\mathrm{Bew}_T(\ulcorner\psi\urcorner) \to \mathrm{Bew}_T(\ulcorner\mathrm{Bew}_T(\ulcorner\varphi\urcorner)\urcorner))$.

(L3) Für alle Sätze ψ und γ: $T \vdash (\mathrm{Bew}_T(\ulcorner\psi\urcorner) \wedge \mathrm{Bew}_T(\ulcorner(\psi \to \gamma)\urcorner)) \to \mathrm{Bew}_T(\ulcorner\gamma\urcorner))$.

Lemma 5.31. *Sei* Bew_T *eine gute Σ_1-Beweisformel für T. Dann erfüllt* (T, Bew_T) *die Löb-Axiome.*

Ein wichtiges Beispiel einer Theorie T, die A_E enthält, für die es eine gute T-Beweisbarkeitsformel gibt, ist die Peano-Arithmetik:

Definition 5.32. Die *Peano-Arithmetik*, kurz PA. ergibt sich, indem wir zu Q für jede Formel φ, die möglicherweise zusätzlich zu x weitere freie Variablen enthält, das folgende Induktionsaxiom für die natürlichen Zahlen hinzufügen:

$$(\varphi(0) \wedge \forall x(\varphi(x) \to \varphi(Sx))) \to \forall x\varphi(x).$$

Satz 5.33. *Satz von Löb. Die Peano-Arithmetik* PA *und* ZFC *(siehe nächstes Kapitel) haben jeweils ein gutes Σ_1-Beweisprädikat.*

Beweis: Siehe Hinman [5] oder Ziegler [9, Kapitel 18 und 20].

Definition 5.34. Wir schreiben Kon_T für $\neg\mathrm{Bew}_T(\ulcorner 0 = 1\urcorner)$.

Satz 5.35. *Zweiter Gödel'scher Unvollständigkeitssatz. Sei T eine konsistente, rekursive (oder auch nur rekursiv aufzählbare) Axiomenmenge ist, so dass es eine Beweisbarkeitsformel für T gibt, derart dass $(T, \mathrm{Bew}_T(x))$ die Löb-Axiome erfüllt. Dann $T \nvdash \mathrm{Kon}_T$.*

Beweis. Sei σ wie oben in Definition 5.26 gewählt. Zuerst zeigen wir, dass $T \vdash \mathrm{Kon}_T \to \sigma$. Da $T \nvdash \sigma$, folgt, dass $T \nvdash \mathrm{Kon}_T$.

Es genügt zu zeigen, dass $T \vdash \neg\sigma \to \neg\mathrm{Kon}_T$. Wir haben:

$T \vdash \mathrm{Bew}_T(\ulcorner\sigma\urcorner) \to \mathrm{Bew}_T(\ulcorner(\mathrm{Bew}_T(\ulcorner\sigma\urcorner))\urcorner)$ nach Axiom (L2) von $\mathrm{Bew}_T(x)$
$T \vdash \mathrm{Bew}_T(\ulcorner\sigma\urcorner) \to (\sigma \to 0 = 1)$ nach Wahl von σ
$T \vdash \mathrm{Bew}_T(\ulcorner(\mathrm{Bew}_T(\ulcorner\sigma\urcorner) \to (\sigma \to 0 = 1))\urcorner)$, denn $A_E \subseteq T$ beweist alle wahren Σ_1-Sätze
$T \vdash (\mathrm{Bew}_T(\ulcorner(\mathrm{Bew}_T(\ulcorner\sigma\urcorner))\urcorner) \wedge \mathrm{Bew}_T(\ulcorner(\mathrm{Bew}_T(\ulcorner\sigma\urcorner) \to (\sigma \to 0 = 1))\urcorner)) \to \mathrm{Bew}_T(\ulcorner(\sigma \to 0 = 1)\urcorner)$ nach Axiom (L3) für $\mathrm{Bew}_T(x)$.
Wir wissen aber, dass T sowohl die zweite Hypothese der letzten Implikation als auch den Satz $\mathrm{Bew}_T(\ulcorner\sigma\urcorner) \to \mathrm{Bew}_T(\ulcorner(\mathrm{Bew}_T(\ulcorner\sigma\urcorner))\urcorner)$ beweist. Somit haben wir:
$T \vdash \mathrm{Bew}_T(\ulcorner\sigma\urcorner) \to \mathrm{Bew}_T(\ulcorner(\sigma \to 0 = 1)\urcorner)$.
Nach Axiom (L3) für die Formel $\mathrm{Bew}_T(x)$ erhalten wir:

$$T \vdash (\mathrm{Bew}_T(\ulcorner\sigma\urcorner) \wedge \mathrm{Bew}_T(\ulcorner(\sigma \to 0 = 1)\urcorner)) \to \mathrm{Bew}_T(\ulcorner(0 = 1)\urcorner).$$

Also $T \vdash \mathrm{Bew}_T(\ulcorner\sigma\urcorner) \to \mathrm{Bew}_T(\ulcorner(0 = 1)\urcorner)$. $T \vdash \neg\sigma \to \mathrm{Bew}_T(\ulcorner\sigma\urcorner)$ nach Wahl von σ. Es folgt, dass $T \vdash \neg\sigma \to \neg\mathrm{Kon}_T$, wie gewünscht. $\dashv$

Die Bedingung, dass T eine gute Beweisformel hat, ist allerdings nicht so einfach zu verifizieren und führt tiefer in die Beweistheorie (als Gebiet gemeint, in der heutzutage gängigen Einteilung der mathematischen Logik in Beweistheorie, Rekursionstheorie, Mengenlehre und Modelltheorie).

Wir formulieren den zweiten Gödelschen Unvollständigkeitssatz noch einmal für die gängigen Axiomensysteme PA und ZFC:

Korollar 5.36. $\mathrm{PA} \nvdash \mathrm{Kon}_{\mathrm{PA}}$ *und ZFC* $\nvdash \mathrm{Kon}_{\mathrm{ZFC}}$.

Kapitel 6

Logisches Programmieren

6.1 Die Resolutionsmethode

In diesem Abschnitt kehren wir zurück zur Aussagenlogik. Wir sahen, dass man mit disjunktiven Normalformen einfacher rechnen kann als mit allgemeinen Formeln. Nun sehen wir eine andere Methode, die auf konjunktive Normalformen zugeschnitten ist, die Resolutionsmethode.

Definition 6.1. (1) Eine Klausel C ist eine endliche Menge von Literalen L.

(2) Eine Belegung v der Variablen erfüllt die Klausel C, wenn $\bar{v}(L) = W$ für ein $L \in C$. Anders ausgedrückt, wenn $\bar{v}(\bigvee_{L\in C} L) = W$.

(3) Sei $\mathcal{C}$ eine endliche Menge von Klauseln. Eine Belegung v erfüllt $\mathcal{C}$, wenn $\bar{v}$ alle $C \in \mathcal{C}$ erfüllt. Das kann man auch schreiben als $\bar{v}(\bigwedge_{C\in\mathcal{C}} \bigvee_{L\in C} L) = W$.

Definition 6.2. Sei A eine Variable, $C = \{A\} \cup P$ und $D = \{\neg A\} \cup Q$ zwei Klauseln. Dann ist $P \cup Q$ eine *Resultante* von C und D. Wir schreiben $P \cup Q = \mathrm{Res}_A(C, D)$.

Satz 6.3. *(Die Resolutionsmethode). Eine endliche Menge $\mathcal{C}$ von Klauseln ist genau dann erfüllbar, wenn sich aus $\mathcal{C}$ durch sukzessives Bilden von Resultanten niemals die leere Klausel ergibt.*

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

Lemma 6.4. *Sei v eine Belegung der Variablen von $\mathcal{C}$ und $\bar{v}(A) = W$. Dann ist $\bar{v}(\bigwedge \mathcal{C}) = W$ genau dann, wenn $\bar{v}(\mathcal{C}|A = W) = W$ ist. Analoges gilt für F statt W. Sei v eine Belegung der Variablen von $\mathcal{C}$ und $v(A) = F$. Dann ist $\bar{v}(\bigwedge \mathcal{C}) = W$ genau dann, wenn $\bar{v}(\mathcal{C}|A = F) = W$ ist.*

Beweis: Sei $v(A) = W$. Aus der Definition der Fortsetzung von v folgt: $\bar{v}(\bigvee_i L_i) = W$ wenn A unter den L_i ist, und $\bar{v}(\bigvee_i L_i) = \bar{v}(\bigvee_{i\neq j} L_i)$ wenn $L_j = \neg A$. Gespiegeltes gilt für $v(A) = F$. $\dashv$

Lemma 6.5. *Sei $\varphi \equiv \psi$. Dann ist φ erfüllbar genau dann, wenn ψ erfüllbar ist.*

Beweis: Definition von $\varphi \equiv \psi$. $\dashv$

Die Umkehrung gilt nicht. A_0 und A_1 sind beide erfüllbar, also ist A_0 erfüllbar genau dann wenn A_1 erfüllbar ist, aber A_0 ist nicht äquivalent mit A_1.

Lemma 6.6. *(1) Seien P, Q Klauseln in den Variablen $A_0, \ldots, A_{n-1}$. $(\bigvee P) \wedge (\bigvee Q)$ ist äquivalent zu $(\bigvee P) \wedge (\bigvee Q) \wedge \bigwedge_{i<n}(\bigvee \mathrm{Res}_{A_i}(P,Q))$.*

(2) Seien P, Q Klauseln in den Variablen $A_0, \ldots, A_{n-1}$. $(\bigvee P) \wedge (\bigvee Q)$ ist erfüllbar gdw $(\bigvee P) \wedge (\bigvee Q) \wedge \bigwedge_{i<n}(\bigvee \mathrm{Res}_{A_i}(P,Q))$ erfüllbar ist.

(3) $\bigwedge \mathcal{C}$ ist äquivalent zu $\bigwedge(\mathcal{C} \cup \{\mathrm{Res}_{A_i}(C_1,C_2) \mid C_1, C_2 \in \mathcal{C}, i < n\})$.

(4) $\mathcal{C}$ ist erfüllbar gdw $\mathcal{C} \cup \{\mathrm{Res}_{A_i}(C_1,C_2) \mid C_1, C_2 \in \mathcal{C}, i < n\}$ erfüllbar ist.

Beweis: (1) Sei $\bar{v}(\bigvee P \wedge \bigvee Q) = W$ und sei A_i eine Variable. Sei $v(A_i) = F$ (der andere Fall geht gespiegelt). Dann gilt nach Lemma 6.4 $\bar{v}(P|A = F) = W$ und $\bar{v}(Q|A = F) = W$. Sei $P = \{A, P_1, \ldots, P_{k-1}\}$ und sei $Q = \{\neg A, Q_1, \ldots, Q_{\ell-1}\}$. Da $\bar{v}(P) = W$, ist $\bar{v}(\{P_1, \ldots, P_{k-1}\}) = W$. Daher ist auch $\bar{v}(\mathrm{Res}_A(P,Q)) = W$. Da A und F beliebig waren, gilt dies für alle Variablen und alle Wahrheitswerte.

Teil (3) folgt daraus, dass man (1) iterativ auf alle Paare von Klauseln in $\mathcal{C}$ anwenden kann. $\dashv$

Ende des Beweises von Satz 6.3 Sei $\mathcal{C}$ eine endliche Menge von Klauseln in den Variablen $A_0, \ldots A_{n-1}$. Das zweite Lemma, so oft angewandt, wie es verschiedene Klauseln über $\{A_0, \ldots, A_{n-1}\}$ gibt, nämlich höchstens 4^n, da jede Variable in jeder Klausel entweder gar nicht, positiv oder negiert oder (positiv und negiert) vorkommen kann, beweist den Satz. $\square_{6.3}$

Sowohl die semantische Methode, Erfüllbarkeit zu prüfen (rate eine Belegung v der Variablen und rechne dann $\bar{v}(\bigvee_{C\in\mathcal{C}} \bigwedge C)$ aus) als auch die Resolutionsmethode haben exponentielle Zeitkomplexität. Wir sahen im Satz von Cook, dass das Erfüllbarkeitsproblem für Formeln in KNF NP-vollständig ist.

Es ist unbekannt, ob es bessere Algorithmen gibt.

6.2 Der Satz von Herbrand und automatisches Beweisen

Nun betrachten wir das Erfüllbarkeitsproblem (oder dual dazu das Allgemeingültigkeitsproblem) in der Sprache der ersten Stufe. Beide Probleme sind von der gleichen Komplexität, da φ nicht erfüllbar ist gdw $\neg\varphi$ allgemeingültig ist.

Wir kennen diese Komplexität schon: Das Erfüllbarkeitsproblem ist nicht entscheidbar. Jedoch ist sein Komplement rekursiv aufzählbar: Nach dem Gödelschen Vollständigkeitssatz ist die Menge der allgemeingültigen Formeln rekursiv aufzählbar. Somit ist auch die Menge der unerfüllbaren Formel rekursiv aufzählbar. Wir werden nun einen weiteren Algorithmus zur Auszählung aller unerfüllbaren Formeln vorstellen. Im Gegensatz zu dem im Kapitel „Unvollständigkeitssätze" skizzierten Algorithmus arbeitet der jetzige Algorithmus mehr mit der semantischen Seite als mit den formalen Beweisen des Beweiskalküls für die Logik der ersten Stufe. Der Algorithmus wurde 1930 von Herbrand vorgestellt und basiert auf einer Kombination folgender Gedankenschritte und Methoden:

(1) Resolutionsmethode für den quantorenfreien Kern
(2) Skolemisierung der zu untersuchenden Formel, so dass nur noch eine Allformel dasteht.
(3) Reduktion auf Sprachen ohne die Gleichheit.
(4) Arbeit mit Termodellen (nicht modulo =) und Unifizierung

Für den ersten Schritt können wir die Ergebnisse des vorigen Abschnitts verwenden, denn es gilt folgendes:

Lemma 6.7. *Sei $F(A_0, \dots, A_{k-1})$ eine aussagenlogische Formel, in den paarweise verschiedenen Variablen A_i. Für jedes $i < k$ sei α_i ein Literalformel (d.h., eine atomare Formel oder negierte atomare Formel) der Form $R(t_0, \dots, t_{n-1})$ oder $\neg R(t_0, \dots, t_{n-1})$. Die α_i seien paarweise verschieden. Dann ist $F(\alpha_0, \dots, \alpha_{k-1})$ allgemeingültig, wenn $F(A_0, \dots, A_{k-1})$ allgemeingültig ist.*

Der Beweis ist offensichtlich.

Bemerkung 6.8. Beachten Sie: $R(t_0, \dots, t_{n-1})$ und $R(t'_0, \dots, t'_{n-1})$ sind vielleicht nicht äquivalent, wenn auch nur ein t_i nicht buchstäblich mit t'_i übereinstimmt. Die Formel

$$R(t_0, \dots, t_{n-1}) \vee \neg R(t'_0, \dots, t'_{n-1})$$

is genau dann allgemeingültig, wenn die beiden Zeichenreihen $R(t_0, \dots, t_{n-1})$ und $R(t'_0, \dots, t'_{n-1})$ übereinstimmen. Sie ahnen vielleicht jetzt schon, dass die Unifikation (das ist ein Überführungsverfahren eines Tupels von Termen in ein anderes Tupel, das wir später kennenlernen) von Termen für die Entscheidung der Allgemeingültigkeit aussagenlogischer Kombinationen von Literalformeln obiger Form relevant ist.

Nun kommen wir zur Skolemisierung, dem zweiten Bestandteil des Herbrand'schen Verfahrens zur Aufzählung der allgemeingültigen Formeln.

Konvention. Wir schreiben in diesem Abschnitt alle Formeln in *bereinigter* Form, d.h. keine gebundene Variable ist gleich benannt wie ein freie Variable, und alle gebundenen Variablen sind paarweise verschieden benannt.

Da $Qx\varphi(x)$ äquivalent zu $Qy\varphi(y)$ ist, hat jede Formel eine äquivalente bereinigte Formel.

Definition 6.9. Eine Formel ist in *pränexer Normalform*, wenn alle Quantoren am Anfang der Formel stehen, wenn also die Formel die Gestalt

$$Q_0 x_0 Q_1 x_1 \dots Q_{n-1} x_{n-1} \psi$$

hat hat ψ quantorenfrei ist und $Q \in \{\exists, \forall\}$.

Lemma 6.10. *(Pränexe Normalform). Jede Formel lässt sich in eine äquivalente pränexe Formel umwandeln.*

Beweis, induktiv über den Aufbau der Formeln. Sei $\varphi = \varphi_0 \wedge \varphi_1$, und seien $\varphi_i = Q^i_0 x^i_0 Q^i_1 x^i_1 \dots Q^i_{n_i-1} x^i_{n_i-1} \psi_i$ für $i = 0, 1$ schon in pränexer Normalform. Wir nennen die x^0_j und die x^1_k so um, dass

$$(\forall j < n_0)(\forall k < n_1)(x^0_j \neq x^1_k).$$

Außerdem dürfen die $\bar{x}^0$ and $\bar{x}^1$ mit keiner anderen freien Variablen $\bar{z}$, die in ψ_0 oder in ψ_1 vorkommt, übereinstimmen.

Dann ist φ äquivalent zu

$$Q^0_0 x^0_0 Q^0_1 x^0_1 \dots Q^0_{n_0-1} x^0_{n_0-1} Q^1_0 x^1_0 Q^1_1 x^1_1 \dots Q^1_{n_1-1} x^1_{n_1-1} (\psi_0 \wedge \psi_1).$$

Für den Induktionsschritt $\varphi = \neg\psi$ nehmen wir an, dass ψ in pränexer Normalform ist und ziehen dann die Quantoren mithilfe der Umwandlungen von $\neg\forall$ in $\exists\neg$ und $\neg\exists$ in $\forall\neg$ die Quantoren nach vorne. Die Induktionsschritte der Form $\varphi = Qx\psi$ bereiten keine Arbeit. $\dashv$

Definition 6.11. Eine *universelle Formel* hat die Form $\forall x_0 ... \forall x_{n-1} \psi$, wobei ψ eine quantorenfreie Formel ist. *Existentielle Formeln* haben die Form $\exists x_0 ... \exists x_{n-1} \psi$, wobei ψ eine quantorenfreie Formel ist.

Satz 6.12. *(Skolem-Normalform). Zu jeder $\mathscr{L}(\tau)$-Aussage φ kann man eine Spracherweiterung $\mathscr{L}(\tau')$ und einen universellen $\mathscr{L}(\tau')$-Satz φ' angeben, derart, dass φ in einer τ-Struktur $\mathfrak{A}$ genau dann gilt, wenn sich $\mathfrak{A}$ zu einem τ'-Modell von φ' expandieren lässt. φ ist also genau dann erfüllbar, wenn φ' erfüllbar ist.*

Beweis: Wir nummerieren nun die $\forall$-$\exists$-Blöcke von rechts an, da im n-ten Induktionsschritt der mit n indizierte Block umgewandelt wird.

Wir nehmen an, dass die Formel schon in bereinigter pränexer Normalform ist, und führen Induktion über die Anzahl der Quantorenblöcke. Nachdem wir einen Quantor $\forall \bar{x}_{n-1}$ eventuell davor schreiben (beachten Sie: $\mathrm{lh}(\bar{x}_n) = 0$ ist erlaubt), können wir o.B.d.A. annehmen, dass

$$\varphi = \forall \bar{x}_{n-1} \exists \bar{y}_{n-1} \dots \forall \bar{x}_0 \exists \bar{y}_0 \psi(\bar{x}_0, \bar{y}_0, \dots, \bar{x}_{n-1}, \bar{y}_{n-1}, \bar{z})$$

mit einem quantorenfreien ψ, das auch der quantorenfreie Kern von φ genannt wird. Hierbei stehen $\forall \bar{x}_i$ für $\forall x_{i,0} \dots \forall x_{i,m_i-1}$ mit der Tupellänge $\mathrm{lh}(\bar{x}_i) = m_i$ und $\exists \bar{y}_i$ für $\exists y_{i,0} \dots \exists y_{i,\ell_i-1}$ mit der Tupellänge ℓ_i. Die Tupellänge m_i der x-Tupel geht in die Stelligkeiten der Skolemfunktionen ein (als Summand, von rechts beginnend). Die Tupellänge ℓ_i der y-Tupel gibt die Stelligkeit der Tupel in der Bildmenge der i-ten Skolemfunktion an.

Induktionsschritt: Sei $\varphi = \forall \bar{x}_{n-1} \exists \bar{y}_{n-1} \chi$, so dass $\chi = \chi(\bar{x}_0, \dots \bar{x}_{n-2}, \bar{x}_{n-1}, \bar{y}_{n-1}, \bar{z})$ nach Induktionsannahme schon eine Allformel ist, in der die Skolemfuntionen F_0, ..., F_{n-2} vorkommen. Wir nehmen ein neues Funktionssymbol F_{n-1} für eine $\sum_{j<n} m_j + \mathrm{lh}(\bar{z})$-stellige Funktion mit ℓ_{n-1}-stelligen Wertebereich (oder ℓ_{n-1} Funktionen mit jeweils einstelligem Wertebereich).

Die Funktion F_{n-1} hängt also in der Regel von *allen* freien Variablen in der Formel χ ab, dieses sind alle $\bar{x}_i$, zusammen von Länge $\sum_{j<n}(m_j + \ell_j)$, und dann noch die freien $\bar{z}$. Dieser letzte Schritt ist nicht repräsentativ. Beachten Sie, dass beim Schritt 0 die freien Variablen in $\psi = \chi_0$ alle $\bar{x}_i$, $i \geq 0$, $\bar{y}_i$, $i > 0$ und alle außer den x und y in der Formel vorkommenden Variablen $\bar{z}$ sind. Allegemein gilt: Beim Schritt j des Verfahrens sind die freien Variablen in χ_j alle $\bar{x}_i$, $i > 0$, $\bar{y}_i$, $i > j$ und alle noch in der Formel vorkommenden Variablen $\bar{z}$.

Wir haben also

$$\begin{aligned}
F_0(\bar{x}_0, \bar{x}_1, \bar{y}_1, \bar{x}_2, \dots, \bar{x}_{n-1}, \bar{y}_{n-1}, \bar{z}),&\ \text{berechnet } \bar{y}_0,\\
F_1(\bar{x}_0, \bar{x}_1, \bar{x}_2, \bar{y}_2, \dots, \bar{x}_{n-1}, \bar{y}_{n-1}, \bar{z})&\ \text{berechnet } \bar{y}_1,\\
&\ \vdots\\
F_{n-1}(\bar{x}_0, \bar{x}_1, \bar{x}_2, \dots, \bar{x}_{n-1}, \bar{z}),&\ \text{berechnet } \bar{y}_{n-1}.
\end{aligned}$$

Nun werden durch das induktive Arbeiten von rechts nach links dann die späteren F_i wiederum in die weiter oben stehenden F_j eingesetzt. Am Ende sind kommen keine $\bar{y}_i$ mehr vor.

Wir zeigen:

$$\begin{aligned}
&\varphi \text{ ist erfüllbar, gdw}\\
&\forall \bar{x}_{n-1} \chi(\bar{x}_0, \dots \bar{x}_{n-2}, \bar{x}_{n-1}, F_{n-1}(\bar{x}_0, \dots, \bar{x}_{n-1}, \bar{z}), \bar{z}) \text{ ist erfüllbar.}
\end{aligned} \tag{$*$}$$

Hier ist es wiederum so, dass die $\bar{x}_{n-1}$ die $\forall$-Variablen im gerade behandelten Quantorenabbau sind, und zusätzlich als freie Parameter in der Formel schon $\bar{z}$ (von ganz vom Anfang) und die $\bar{x}_0$, ..., $\bar{x}_{n-1}$ (von den schon übersetzten $\forall \bar{x}_i \exists \bar{y}_i$, $i < n-1$) da sind. Von all diesen darf F_{n-1} abhängen.

Wir schreiben $\bar{x} = (\bar{x}_0, \dots, \bar{x}_{n-1})$. Die Implikation „ $\Rightarrow$ in $(*)$ folgt direkt aus Auswahlaxiom, das garantiert, dass es zu $\forall \bar{x} \exists y \chi(\bar{x}, \bar{y}, \bar{z})$ eine *Auswahlfunktion* F gibt, so dass $\forall x \chi(\bar{x}, F(\bar{x}, \bar{z}))$. Eine Funktion heißt Auswahlfunktion zu einem Mengensystem

$$\{((\bar{x}, \bar{z}), \{\bar{y} \mid \chi(\bar{x}, \bar{y}, \bar{z})\}) \mid (\bar{x}, \bar{z}) \in A^{\sum_{j<n} m_j + \mathrm{lh}(\bar{z})}\},$$

wenn für alle $(\bar{x}, \bar{z}) \in A^{\sum_{j<n} m_j + \mathrm{lh}(\bar{z})}$, $F(\bar{x}, \bar{z}) \in \{\bar{y} \mid \chi(\bar{x}, \bar{y}, \bar{z}))\}$.

Die Implikation „$\Leftarrow$" in $(*)$ beweist man durch „Vergessen" von F. $\dashv$

Proposition 6.13. *Falls φ in einer Sprache ohne das Gleichheitszeichen formuliert ist, finden wir im Satz über die Skolem'sche Normalform immer ein φ' ohne Gleichheitszeichen.*

Man nennt die neu eingeführten Konstanten und Funktionszeichen in $\tau' \setminus \tau$ *Skolemfunktionen.* Streng genommen haben wir keine Funktionssymbole für Funktionen mit ℓ_i-stelliger Bildmenge in einer zur ersten Stufe gehörenden Symbolmenge, doch man kann leicht jede solche Funktion als ℓ_i Funktionen (nämlich die Komponentenfunktionen) mit einstelliger Bildmenge schreiben.

Durch Bilden der negierten Formeln erhält man:

Korollar 6.14. *(Herbrand-Normalform). Zu jeder $\mathscr{L}(\tau)$-Aussage φ kann man eine Spracherweiterung τ' und einen existentiellen $\mathscr{L}(\tau')$-Satz φ' angeben, der genau dann allgemeingültig ist, wenn φ allgemeingültig ist.*

[1]

Wir betrachten nun nur Sätze φ, denn jede Formel mit freien Variablen ist allgemeingültig, genau dann wenn die All-Abquantifizierung aller freien Variablen allgemeingültig ist.

Definition 6.15. Sei φ ein Allsatz in Skolemform. Falls es kein Konstantensymbol in φ (oder in der Sprache) gibt, fügen wir ein Konstantensymbol a hinzu. Das *Herbrand-Universum $D(\varphi)$ von φ* ist die kleinste Menge, so dass folgendes gilt

(a) Jede Konstante in φ und a sind Elemente von $D(\varphi)$,

(b) wenn $t_0, \dots, t_n \in D(\varphi)$ und f ein Funktionssymbol ist (entweder eine Skolemfunkion oder eine alte Funktion), dann ist $ft_0 \dots t_{n-1} \in D(\varphi)$.

Definition 6.16. Sei φ ein Allsatz in Skolemform. Eine Struktur $\mathfrak{A}$ mit Träger A heißt *Herbrand-Struktur von φ*, wenn folgendes gilt

(a) $A = D(\varphi)$,

(b) wenn f ein Funktionssymbol ist (entweder eine Skolemfunkion oder eine alte Funktion), dann ist für alle $t_0, \dots, t_n \in D(\varphi)$, $f^{\mathfrak{A}}(t_0 \dots t_{n-1}) = ft_0 \dots t_{n-1} \in D(\varphi)$.

Definition 6.17. $\mathfrak{A}$ heißt *Herbrandmodell von φ* gdw $\mathfrak{A}$ eine Herbrandstruktur für φ ist und $\mathfrak{A} \models \varphi$.

Hat jeder erfüllbare Allsatz in Skolemform ein Herbrand-Modell? Wenn zwei Terme t_0, t_1 im Herbrand-Universum als unterschiedlich interpretiert sind, gibt es Schwierigkeiten, wenn die Sprache $=$ enthält und die Formel φ zum Beispiel $t_0 = t_1$ fordert. Wir betrachten daher in diesem Kapitel nicht die ganze erste Stufe, sondern lassen das Gleichheitszeichen weg. Man verliert dadurch keine Ausdrucksstärke, wenn wir uns nur für die Erfüllbarkeit interessieren. Dies zeigen wir im Folgenden:

Definition 6.18. Eine zweistellige Relation E heißt *Kongruenzrelation auf einer Struktur $\mathfrak{A}$*, gdw wenn E eine Äquivalenzrelation (d.h. reflexiv, symmetrisch und transitiv) ist, so dass für alle $P \in \tau$ und $f \in \tau$ und $c \in \tau$ gilt:

[1]Ende des Vorgelesenen im WS 22/23.

(1) Wenn P n-stellig ist, dann gilt für alle $\bar{a}, \bar{b} \in A^n$,

$$((\bigwedge_{i<n} a_i E b_i) \rightarrow (\bar{a} \in P^{\mathfrak{A}} \leftrightarrow \bar{b} \in P^{\mathfrak{A}})).$$

(2) Wenn f n-stellig ist, dann gilt für alle $\bar{a}\bar{b} \in A^n$, $a_n, b_n \in A$,

$$((\bigwedge_{i<n+1} a_i E b_i) \rightarrow (f^{\mathfrak{A}}(\bar{a}) = a_n \leftrightarrow f^{\mathfrak{A}}(\bar{b}) = b_b)).$$

(3) Wenn $c \in \tau$ eine Konstante ist, dann gilt für alle $a, b \in A$,

$$(aEb \rightarrow (c^{\mathfrak{A}} = a \leftrightarrow c^{\mathfrak{A}} = b)).$$

Sei $E \notin \tau$. Beachten Sie, dass man „E ist in allen τ-Strukturen eine Kongruenzrelation" für endliches τ in einem Satz $\mathrm{Kongr}(E, \tau)$ der ersten Stufe ausdrücken kann. $\varphi(\frac{=}{E})$ entstehe aus φ, indem man jedes Gleichheitszeichen durch ein E ersetzt.

Satz 6.19. *Sei $\varphi \in \mathscr{L}(\tau)$ und $E \notin \tau$. Dann ist φ erfüllbar gdw $\varphi(\frac{=}{E}) \wedge$ $\mathrm{Kongr}(E, \tau)$ erfüllbar ist.*

Beweis: Die Vorwärtsrichtung folgt daraus, dass die Gleichheit eine Kongruenzrelation ist. Für die Rückwärtstichtung nehmen wir ein Modell von $\varphi(\frac{=}{E}) \wedge$ $\mathrm{Kongr}(E, \tau)$ und interpretieren jede E-Klasse als ein Element und übertragen alle Relationen und Funktionen treu auf den Quotienten. Die so erhaltene Quotienten-Struktur ist ein Modell von φ. $\dashv$

Satz 6.20. *Sei Φ eine Menge von Allsätzen in einer Sprache ohne das Gleichheitszeichen. Dann hat Φ ein Modell gdw Φ ein Herbrandmodell hat.*

Beweis: Wir wiederholen die Henkin-Konstruktion. Sie liefert bei minimaler Wahl der Konstantenmenge automatisch ein Herbrandmodell. $\dashv$

Proposition 6.21. *Falls φ in einer Sprache ohne das Gleichheitszeichen formuliert ist, finden wir im Satz über die Skolem'sche Normalform immer ein φ' ohne Gleichheitszeichen.*

Nachdem wir uns nach den vorigen effektiven Übersetzungen auf Existenzsätze ohne Gleichheitszeichen beschränken können, kehren wir zurück zum Erfüllbarkeitsproblem für Formeln, oder dual dazu, zum Allgeingültigkeitsproblem:

Satz 6.22. *(Satz von Herbrand). Sei $\psi(x_0, \dots, x_{n-1})$ eine quantorenfreie Formel in einer Sprache $\mathscr{L}(\tau)$, die mindestens eine Konstante enthält. Dann ist*

$$\exists x_0 \exists x_1 \dots \exists x_{n-1} \psi(x_0, x_1, \dots, x_{n-1})$$

genau dann allgemeingültig, wenn es $m \in \mathbb{N}$ und Terme ohne Variablen

$$t_0^0, t_1^0, \dots, t_{m-1}^0, t_0^1, t_1^1, \dots, t_{n-1}^1, \dots, t_0^{m-1}, t_1^{m-1}, \dots, t_{n-1}^{m-1}$$

gibt, für die die quantorenfreie Aussage

$$\bigvee_{i=0}^{m-1} \psi(\bar{t^i}) = \psi(t_0^0, t_1^0, \ldots, t_{m-1}^0) \vee \psi(t_0^1, t_1^1, \ldots, t_{n-1}^1) \vee \cdots \vee \psi(t_0^{m-1}, t_1^{m-1}, \ldots, t_{n-1}^{m-1})$$

allgemeingültig ist.

Beweis: $t_0^0, t_1^0, \ldots, t_{m-1}^0, t_0^1, t_1^1, \ldots, t_{n-1}^1, \ldots, t_0^{m-1}, t_1^{m-1}, \ldots, t_{n-1}^{m-1}$ impliziert natürlich φ, denn die Kontraposition des entsprechenden Ersetzungsaxioms (erinnern Sie sich an den Hilbertkalkül?) ist korrekt.

Für die umgekehrte Implikation nehmen wir an, dass für jede beliebige Wahl der konstanten Terme t_i^j die Formel

$$\bigwedge_{i=0}^{m-1} \neg\psi(\bar{t^i}) = \neg\psi(t_0^0, t_1^0, \ldots, t_{m-1}^0) \wedge \neg\psi(t_0^1, t_1^1, \ldots, t_{n-1}^1) \wedge \cdots \wedge \neg\psi(t_0^{m-1}, t_1^{m-1}, \ldots, t_{n-1}^{m-1})$$

erfüllbar ist. Dann ist die Theorie

$$\{\neg\psi(t_0, \ldots, t_{n-1}) \,|\, t_i \text{ Terme ohne freie Variablen}\}$$

endlich erfüllbar und somit nach dem Kompaktheitssatz erfüllbar. Sei $\mathfrak{A}$ ein Modell. In dieser Theorie sind alle Sätze quantorenfrei und daher sind sie Allsätze. Nach dem Satz 6.20 hat die Theorie dann auch ein Herbrandmodell. In diesem sind alle Punkte konstante Terme, und daher erfüllt das Herbrandmodell $\forall x_0 \ldots \forall x_{n-1} \neg\psi$. Also ist φ nicht allgemeingültig. $\dashv$

Definition 6.23. Sei $\varphi = \forall x_0 \ldots, \forall x_{n-1}\psi$ ein Skolemsatz in einer Sprache ohne Gleichheitszeichen. Die *Herbrand-Expansion $E(\varphi)$ von φ* ist definiert durch

$$(\varphi) = \{\psi(\frac{t_0}{x_0}, \ldots, \frac{t_{n-1}}{x_{n_1}}) \,|\, t_0, \ldots, t_{n-1} \in D(\varphi\}.$$

Als Korollar des Satzes von Herbrand erhalten wir:

Satz 6.24. *(Gödel, Herbrand, Skolem) Sei φ ein Skolemsatz in einer Sprache ohne Gleichheitszeichen. Dann ist φ erfüllbar gew $E(\varphi)$ als Menge aussagenlogischer Formeln aufgefasst, ein Modell hat.*

Nun kommen wir zum letzten Schritt des Herbrand'schen Aufzählverfahrens aller allgemeingültigen Formeln: Wann hat eine quantorenfreie Formel der ersten Stufe kein Modell? Dies ist ein aussagenlogisches Problem (das wir schon kennen) und ein Unifikationsproblem. Die Bestandteile (d.h., die einzelnen Disjunktions- und die Konjunktionsglieder) der Formel sind ja von der Form

$$R(s_0, \ldots, s_{n-1})$$

mit Termen s_i und Relationssymbolen oder negierten Relationssymbolen R.

Definition 6.25.

$$S^0(x_0,\dots,x_{n-1}) = (s^0_0(x_0,\dots,x_{n-1}),\dots,s^0_{k-1}(x_0,\dots,x_{n-1}))$$

und

$$S^1(x_0,\dots,x_{n-1}) = (s^1_0(x_0,\dots,x_{n-1}),\dots,s^1_{k-1}(x_0,\dots,x_{n-1}))$$

seien zwei gleich lange Folgen von Termen. Eine Termfolge $T = (t_0,\dots,t_{n-1})$ unifiziert S^0 und S^1, wenn

$$S^0(t_0,\dots,t_{n-1}) = S^1(t_0,\dots,t_{n-1}).$$

Hier steht $S^j(t_0,\dots,t_{n-1})$ für

$$\langle s^j_{k'}(\frac{t_0}{x_0},\dots,\frac{t_{n-1}}{x_{n-1}}) \mid k' < k\rangle.$$

Bemerkung 6.26. Beachten Sie, dass es in den sukzessiven Ersetzungen ganz entscheidend auf die Reihenfolge ankommt:

$$s_{k'}(\frac{t_0}{x_0},\dots,\frac{t_{n-1}}{x_{n-1}})$$

steht für

$$((s_{k'}(\frac{t_0}{x_0}),\dots)(\frac{t_{n-1}}{x_{n-1}}).$$

Ersetzung ist die Substitution aus dem vierten Kapitel, die es auf der syntaktischen Seite dort in α^t_x gibt. Die Bildung α^t_x bedeutet die Bildung $s(\mathsf{t}x)$ (dies heißt, s, in dem alle Vorkommen von x durch t ersetzt sind) für jeden Term s in α. Schon in Kapitel 3 sahen wir eine entsprechende Substitution auf der semantischen Seite. In den t_i können die Variabeln x_j mit $j \geq i$ vorkommen.

Satz 6.27. *(Unifikationssatz, John Alan Robinson). Wenn S^0 und S^1 unifizierbar sind, gibt es eine universelle unifizierende Termfolge $U(x_0,\dots,x_{n-1})$. Das heißt, dass*

$$S^0(U(x_0,\dots,x_{n-1})) = S^1(U(x_0,\dots,x_{n-1})$$

und dass eine Termfolge T genau dann S^0 und S^1 unifiziert, wenn es Terme $r_0,\dots,r_{n-1}$ gibt, so dass

$$T = U(r_0,\dots,r_{n-1}).$$

Man kann U durch ein einfaches Verfahren finden, das gleichzeitig entscheidet, ob S^0 und S^1 unifizierbar sind.

Beweis: Wir geben einen Algorithmus an und beweisen dessen Korrektheit. Wir fassen S^0, S^1 als die Menge der Gleichungen

$$S = \{s^0_i = s^1_i \mid i < n\}$$

auf. Eine Folge von Ersetzungen T für die x_i unifiziert S, wenn alle Gleichungen in S allgemeingültig sind, d.h., in mit $\forall x_0 \dots \forall x_n$ abquantifizierter Form gelten. Unser Unifikationsverfahren formt S in äquivalente (und auch nicht stärkere) Gleichungssysteme um. Wir wenden, solange es geht, (in irgendeiner möglichen Reihenfolge) die folgenden beiden Umwandlungsschritte an:

(1) Wenn S eine Gleichung $f^0 t^0_0 \dots t^0_{\ell_0-1} = f^1 t^1_0 \dots t^1_{\ell_1-1}$ enthält und $f^0 \neq f^1$, dann ist S nicht unifizierbar und das Verfahren bricht ab. Wenn hingegen $f^0 = f^1$, dann ersetzen wir die Gleichung durch $t^0_j = t^1_j$, $j < \ell_0$. (Es ist dann automatisch $\ell_0 = \ell_1$.) Überlegen Sie sich mithilfe des Termrangs (definiert durch $\mathrm{rk}(x_i) = 0$, $\mathrm{rk}(f s_0 \dots s_{n-1}) = \max\{\mathrm{rk}(s_i) \mid i < n\} + 1$), dass es im Verfahren nur endlich viele Schritte der ersten Art geben kann.

(2) Wenn S eine Gleichung der Form $x_i = s$ enthält, und s ein zusammengesetzter Term ist, in dem die Variable x_i vorkommt, dann bricht das Verfahren ab und S ist nicht unifizierbar. Wenn $s = x_i$, dann streichen wir die Gleichung einfach durch. Wenn x_i nicht in s vorkommt, dann ersetzen wir in allen *anderen* Gleichngen in S die Variable x_i durch den Term s. Beachten Sie, dass von nun an x_i nicht mehr in S vorkommt. Es gibt also höchstens n Schritte der zweiten Art.

Wenn das Verfahren nicht abbricht mit dem Ergebnis, dass es keine Unifizierung gibt, dann stoppt das Verfahren, wenn das umgewandelte S – nach Umnummerierung der Variablen – die Gestalt

$$\{x_m = u_m(x_0, \dots, x_{m-1}), \dots, x_{n-1} = u_{n-1}(x_0, \dots x_{m-1})\}$$

hat für ein $0 \leq m \leq n-1$. Eine gesuchte universelle unifizierende Termfolge ist dann

$$U = (x_0, \dots x_{m-1}, u_m, \dots u_{n-1}).$$

Da das Verfahren erfolgreich stoppt, hat S also im Stoppzusand nur noch Gleichungen der Art $t^0_j = t^1_j$, in denen die beiden t^i_j genau die gleiche Zeichenreihe sind. Die Menge dieser Gleichungen ist natürlich allgemeingültig.

Wir zeigen nun (ohne Umnummerierung), dass U tatsächlich eine universale unifizierende Folge ist.

Sei

$$U = \mathrm{sub}^U_{n^U-1} \circ \dots \circ \mathrm{sub}^U_0 = (\mathrm{sub}^U_0, \dots, \mathrm{sub}^U_{n^U-1})$$

mit

$$\mathrm{sub}^U_k = \frac{t^U_k(x_\ell \mid \ell \in n \setminus \{j^U_0, \dots, j^U_k\})}{x_{j^U_k}}$$

für $k < n^U$. Sei

$$V = (\mathrm{sub}^V_0, \dots, \mathrm{sub}^V_{n^V-1})$$

mit

$$\mathrm{sub}^V_k = \frac{t^V_k(x_\ell \mid \ell \in n \setminus \{j^V_0, \dots j^V_k\})}{x_{j^V_k}}$$

für $k < n^V$ und sei

$$S^0(V) = S^1(V).$$

Wir zeigen, $V = R \circ U = U(R)$ für eine geeignete Ersetzung R.

Werde $x_{j_0^u}$ in der Ersetzung V im $\mathrm{ind}(j_0^U)$-ten Schritt in $\mathrm{sub}^V_{\mathrm{ind}(j_0^U)}$ behandelt Dann ist für $i = 0, 1$,

$$S^i V =$$

$$(S^i \frac{t_0^U}{x_{j_0^U}})(\mathrm{sub}_0^V(\frac{t_0^U}{x_{j_0^U}}), \dots, \mathrm{sub}^V_{\mathrm{ind}^V(j_0^U)-1}(\frac{t_0^U}{x_{j_0^u}}), \mathrm{sub}^V_{\mathrm{ind}^V(j_0^U)+1}, \dots, \mathrm{sub}^V_{n^V-1}).$$

Für alle x_i mit $i \neq j_0^U$ ist dies klar. Für $x_{j_0^U}$ überlegt man sich, was V vor der Ersetzung von $x_{j_0^U}$, also in $(\mathrm{sub}_0^V(\frac{t_0^U}{x_{j_0^U}}), \dots, \mathrm{sub}^V_{\mathrm{ind}^V(j_0^U)-1}(\frac{t_0^U}{x_{j_0^u}}))$, bewirkt, und was $\mathrm{sub}^V_{\mathrm{ind}^V(j_0^U)}(x_{j_0^U}$ getan hätte.

In den $\mathrm{sub}^V_{\mathrm{ind}^V(j_0^U)+1}$, ..., $\mathrm{sub}^V_{n^V-1}$ kommt $x_{j_0^U}$ nicht vor. Sukzessive zieht man mehr Schritte von U nach vorne, bis alle Ersetzungen in U in der Mitte im Term auf der rechten Seite der Gleichung stehen. Dies ist gestattet, da U nur Variablen identifiziert, die durch die Gleichung $S^0 U = S^1 U$ gefordert werden. ⊣

⊣

Korollar 6.28. *Sei $\psi(x_0, \dots, x_{n-1})$ eine quantorenfreie $\mathscr{L}(\tau)$-Formel ohne Gleichheitszeichen und m eine natürliche Zahl. Man kann effektiv entscheiden, ob es konstante Terme*

$$t_0^0, t_1^0, \dots, t_{m-1}^0, t_0^1, t_1^1, \dots, t_{n-1}^1, \dots, t_0^{m-1}, t_1^{m-1}, \dots, t_{n-1}^{m-1}$$

gibt, für die die quantorenfreie Aussage

$$\varphi = \bigvee_{i=0}^{m-1} \psi(\bar{t^i}) = \psi(t_0^0, t_1^0, \dots t_{n-1}^0) \vee \psi(t_0^1, t_1^1, \dots, t_{n-1}^1) \vee \dots \vee \psi(t_0^{m-1}, t_1^{m-1}, \dots, t_{n-1}^{m-1})$$

allgemeingültig ist.

Beweis: Die Formel enthält womöglich freie Variablen. Daher ist sie allgemeingültig, genau dann wenn ihre mit $\forall$ abquantifizierte Form allgemeingültig ist. Man prüft, ob sich für jedes Symbol R, das in φ vorkommt, für alle Literalformeln der Art

$$(\neg)R(s_0^i, \dots, s_{n-1}^i)$$

jeden $\bar{s}^i$ such mit jedem $\bar{s}^{i'}$ in der Formel unifizieren lässt, und wählt sukzessive Unifikatoren. Wenn das Unifikationsverfahren erfolgreich endet, dann prüft man, ob die aussagenlogische Formel ψ allgemeingültig ist. Sie ist allgemeingültig genau dann, wenn die Formel

$$\psi(x_0, \dots, x_{m-1}, u_m, \dots, u_{n-1})$$

allgemeingültig ist (nach geeigneter Umnummerierung). Diese Formel kann man aussagenlogisch behandeln mit unserem ersten Kapitel. Sie ist allgemeingültig, genau dann, wenn ihre Allabquantifizierung allgemeingültig ist. Genau dann, wenn beide Prüfungen positiv enden, ist die Aussage allgemeingültig. $\dashv$

Bemerkung: Jedoch kann man nicht entscheiden, ob es im Satz von Herbrand ein m gibt, so wie dort (dies zeigt man wie im ersten Gödel'schen Unvollständigkeitssatz). Nur bei festem m und gegebenen t_i^j, $j < m$, kann man die Prüfung starten. Der Satz von Herbrand gibt eine positive Antwort, wenn die Formel allgemeingültig ist. Doch bis zu welchem m soll man den Suchalgorithmus laufen lassen? Wir erhalten also wieder wie im Gödel'schen Vollständigkeitssatz und dem ersten Unvollständigkeitssatz zusammen, dass die Menge der allgemeingültigen $\mathscr{L}(\tau)$-Formeln rekursiv aufzählbar und im Allgemeinen nicht rekursiv ist. (Dies ist sie allerhöchstens bei ganz kleinen Symbolmengen τ).

Literaturverzeichnis

[1] Heinz-Dieter Ebbinghaus, Jörg Flum, and Wolfgang Thomas. *Einführung in die Mathematische Logik.* Hochschultaschenbuch, 4 edition, 1996.

[2] Herbert Enderton. *A Mathematical Introduction to Logic.* Academic Press, 3 edition, 2001.

[3] Ryszard Engelking. *General topology.* Heldermann Verlag, second edition edition, 1989.

[4] Hans Hermes. *Einführung in die mathematische Logik. Klassische Prädikatenlogik.* Mathematische Leitfäden. B. G. Teubner, vierte auflage edition, 1976.

[5] Peter G. Hinman. *Fundamentals of Mathematical Logic.* A K Peters, Wellesley, Massachusetts, 2005.

[6] Uwe Schöning. *Logic for computer scientists.* Birkhäuser, 1989.

[7] Joseph Shoenfield. *Mathematical Logic, Reprint of the 1973 second printing.* Association for Symbolic Logic, Urbana, IL; A K Peters, Ltd., Natick, MA, 2001.

[8] Michael Sipser. *Introduction to the Theory of Computation.* Thompson Course Technology, second international edition edition, 2006.

[9] Martin Ziegler. *Mathematische Logik.* Mathematik kompakt. Birkhäuser, 2010.

Index