Skript

Logik und formale Systeme

Sommersemester 2015

Prof. Dr. Heribert Vollmer
Thorsten Kluge

Institut für Theoretische Informatik

Leibniz Universität Hannover

Version vom 12. April 2015

Gesetzt von Sven Karsten Greiner

Inhaltsverzeichnis

0. Die Bedeutung der Logik für die Mathematik und die Informatik **1**

I. Aussagenlogik **3**

1. Syntax und Semantik der Aussagenlogik **5**
1.1. Die Sprache der Aussagenlogik 5
1.2. Die Semantik der Aussagenlogik 9
1.3. Normalformen 12
1.4. Das Erfüllbarkeitsproblem 16
1.5. Der Endlichkeitssatz 17
1.6. Anwendungen 19
1.6.1. Formeln und logische Gatter 19
1.6.2. Entscheidungsdiagramme 20
1.6.3. Minimierung von DNF 21

2. Hornformeln **23**

3. Resolution **27**
3.1. Resolutionsbeweise 27
3.2. Erfüllbarkeitstests 33

4. Folgern und Schließen **37**
4.1. Ein Ableitungskalkül 38
4.2. Regeln des natürlichen Schließens 39
4.3. Der Vollständigkeitssatz 42
4.3.1. Korrektheit 42
4.3.2. Widerspruchsfreiheit (Konsistenz) von Formelmengen 44
4.3.3. Vollständigkeit von Formelmengen 44
4.4. Exkurs: Semantische Tableaus 47

5. Modallogik **49**
5.1. Syntax der Modallogik 49
5.2. Semantik der Modallogik 49
5.3. Beispiele 51
5.4. Charakterisierung von Rahmeneigenschaften 53
5.5. Weitere Themengebiete 55
5.5.1. Bisimulation 55
5.5.2. Multi-Modallogik 56
5.5.3. Temporale Logik 58

6. Quantifizierte Boole'sche Formeln **61**

II. Prädikatenlogik 63

7. Mathematische Strukturen und Abbildungen **65**
7.1. Grundbegriffe . . . 65
7.2. Strukturen . . . 66
7.3. Beispiele . . . 67
7.4. Mehrsortige Strukturen . . . 68
7.5. Homomorphismen und Isomorphismen . . . 69
7.6. Kongruenzrelationen und Quotientenstrukturen . . . 71
7.7. Kongruenzrelationen und und Homomorphismen . . . 72

8. Die Syntax der Prädikatenlogik **75**

9. Die Semantik der erststufigen Prädikatenlogik **79**

10. Prädikatenlogisches Formalisieren **85**
10.1. Graphen . . . 85
10.2. Arithmetik . . . 87
10.3. Zusammenhang zwischen Modallogik und Prädikatenlogik . . . 88

11. Axiomensysteme **91**

12. Wichtige Äquivalenzen und Normalformen **95**

13. Folgern und Schließen **99**
13.1. Der Ableitungsbegriff in der Prädikatenlogik der 1. Stufe . . . 99
13.2. Grenzen der Formalisierbarkeit . . . 101

A. Elementare Begriffe und Schreibweisen **105**
A.1. Elementbeziehung und Enthaltenseinsrelation (Inklusion) . . . 105
A.2. Möglichkeiten der Definition spezieller Mengen . . . 105
A.3. Operationen auf Mengen . . . 106
A.4. Gesetze für Mengenoperationen . . . 106
A.5. Tupel (Vektoren) und Kreuzprodukt . . . 107
A.6. Anzahl . . . 107
A.7. Induktion . . . 107
A.8. Griechisches Alphabet . . . 108

Literaturverzeichnis **109**

Index **111**

Kapitel 0

Die Bedeutung der Logik für die Mathematik und die Informatik

Die Logik hat sowohl in der Mathematik als auch in der Informatik große Bedeutung.

Bedeutung in der Mathematik

Sie dient der Grundlegung der Mathematik und ihrer Theorien und beantwortet Fragen wie

- Was ist ein Beweis? — Beweis
- Was ist eine Theorie? — Theorie
- Was sind Mengen?
- Kann das semantische Folgern durch algorithmisches Schließen ersetzt werden? — Folgern, Schließen
- Welche mathematischen Theorien sind algorithmisch beherrschbar? (Sind Beweise durch Computer möglich?)

Bedeutung in der Informatik

In der Informatik dient die Logik eher als Handwerkszeug. Sie findet insbesondere Anwendung in folgenden Bereichen:

- Exakte Beschreibungssprache für
  - Aussagen
  - Daten
  - Schaltkreise und deren Funktionsweise
- Algorithmisches Schließen
  - Automatisches Beweisen

  - Logikprogrammierung
  - Schließen in Datenbanken
  - Schließen in Expertensystemen
- Programmiersprachen
  - Semantik von Programmiersprachen
  - Programm-Verifikation

Sehr kurzer geschichtlicher Abriss

Griechische Antike: Bereits axiomatisches Vorgehen (Euklidische Geometrie). Ansätze einer Formalisierung und logischen Sprache bei Aristoteles (Syllogismen) und den Stoikern (antike Form der Aussagenlogik und Modallogik).

ca. 1690 Leibniz: Mathematisierung der Aristotelischen Logik. Untersuchung der Modalitäten *möglicherweise* und *notwendigerweise*, der Begriff der „möglichen Welt“

19. Jahrhundert: de Morgan, Boole, Schröder, Peano: Ansätze logischer Kalküle

1879 Gottlob Frege, Die Begriffsschrift: Erster logischer Kalkül (Sprache), der sich aber aufgrund seiner komplizierten zweidimensionalen Notation nicht durchsetzen konnte. Heutige Notation an Peano angelehnt.

1895 Georg Cantor: Begründung der naiven Mengenlehre.
Eine beliebige Zusammenfassung von Objekten zu einem Ganzen heißt Menge.

1902 Bertrand Russell: Widersprüchlichkeit der naiven Mengenlehre.
Russellsche Antinomie: Die Bildung der naiven Menge $X = \{A : A \notin A\}$ führt zum Widerspruch $X \in X \Leftrightarrow X \notin X$.

1908 Zermelo/Fraenkel: Axiomatik der Mengenlehre.
Axiome legen fest, was eine Menge ist.
Hoffnung: Widersprüche werden vermieden (aber noch unklar).

1920 David Hilbert: Programm des Formalismus.
Jede mathematische Theorie soll axiomatisch aufgebaut werden.

1930 Kurt Gödel: Korrektheit und Vollständigkeit des Prädikatenkalküls.

1931 Kurt Gödel: Keine anspruchsvolle Theorie ist axiomatisierbar!

„The confluence of ideas in 1936“ (R. Gandy): Arbeiten von Turing, Church, Kleene und anderen entwickeln den Begriff der *Berechenbarkeit*.
Anwendung auf die Logik: Das Hilbert'sche Entscheidungsproblem (der Gültigkeit prädikatenlogischer Formeln) ist nicht lösbar.

ca. 1965: Grundlagen der Komplexitätstheorie in Arbeiten von Hartmanis, Lewis and Stearns

1971: Das Erfüllbarkeitsproblem für aussagenlogische Formeln ist NP-vollständig (Satz von Cook).

Teil I

Aussagenlogik

Kapitel 1

Syntax und Semantik der Aussagenlogik

Wir fassen die Menge aller aussagenlogischen Formeln als Sprache Form auf. Die syntaktisch korrekt gebildeten aussagenlogischen Formeln sind dabei die Wörter unserer Sprache Form.

1.1. Die Sprache der Aussagenlogik

Der Zeichenvorrat der Aussagenlogik besteht aus

(i) der Menge der *aussagenlogischen Variablen* $\mathsf{Var} = \{p_1, p_2, p_3 \dots\}$, — aussagenlogische Variablen

(ii) den Konstanten 0, 1,

(iii) den Konnektoren — Konnektor

- $\wedge$ (Konjunktion)
- $\vee$ (Disjunktion)
- $\neg$ (Negation)
- $\rightarrow$ (Implikation)
- $\leftrightarrow$ (Äquivalenz oder Biimplikation)

(iv) und den Klammersymbolen $(,)$.

Über diesem Zeichenvorrat definieren wir jetzt induktiv die Menge der aussagenlogischen Formeln $\mathsf{Form}_{\mathrm{AL}}$.

Definition 1. Die Menge der *aussagenlogischen Formeln* $\mathsf{Form}_{\mathrm{AL}}$ ist induktiv definiert als: — Syntax der Aussagenlogik

(i) Jede Variable $p \in \mathsf{Var}$ und Konstante $0, 1$ ist in $\mathsf{Form}_{\mathrm{AL}}$. Diese Formeln heißen auch *atomare Formeln* oder kurz *Atom.* oder *elementare Aussagen.* — Atom, elementare Aussagen

(ii) Seien $\varphi, \psi \in \mathsf{Form}_{\mathrm{AL}}$. Dann ist auch

- $\neg\varphi \in \mathsf{Form}_{\mathrm{AL}}$
- $(\varphi \vee \psi) \in \mathsf{Form}_{\mathrm{AL}}$
- $(\varphi \wedge \psi) \in \mathsf{Form}_{\mathrm{AL}}$
- $(\varphi \to \psi) \in \mathsf{Form}_{\mathrm{AL}}$
- $(\varphi \leftrightarrow \psi) \in \mathsf{Form}_{\mathrm{AL}}$.

Im Sinne der Theorie der formalen Sprachen kann $\mathsf{Form}_{\mathrm{AL}}$ als Sprache aufgefasst werden. Dazu zunächst einige Definitionen:

Alphabet Ein *Alphabet* Σ ist eine endliche Menge von Zeichen, die wir *Buchstaben* nennen. Ein *Wort* über Σ ist eine endliche Folge $a_1 a_2 \dots a_n$ von Buchstaben aus Σ, d.h. $n \in \mathbb{N}$ und $a_1, \dots, a_n \in \Sigma$. Die Menge der Wörter über Σ wird mit Σ^* bezeichnet.

Sprache Eine *Sprache* L über dem Alphabet Σ ist eine Menge von Wörtern über Σ, d.h. $L \subseteq \Sigma^*$.

Σ_{AL} Das Alphabet der Aussagenlogik ist

$$\Sigma_{\mathrm{AL}} = \{p, I, 1, \wedge, \vee, \neg, \to, \leftrightarrow, (,)\}.$$

$\mathsf{Form}_{\mathrm{AL}}$ Die Formelmenge $\mathsf{Form}_{\mathrm{AL}}$ ist eine Sprache über Σ_{AL}, also $\mathsf{Form}_{\mathrm{AL}} \subseteq \Sigma_{\mathrm{AL}}^*$, die durch folgende Grammatik erzeugt wird:

$$G := (\Sigma_{\mathrm{AL}}, \{S, V, C\}, P, S)$$
$$P := \left\{ \begin{array}{l} S \to V \mid C \mid \neg S \mid (S \wedge S) \mid (S \vee S) \mid (S \to S) \mid (S \leftrightarrow S) \\ V \to p \mid VI \\ C \to 0 \mid 1 \end{array} \right\}$$

Für pI^i schreiben wir auch p_i.

Die syntaktisch korrekten Wörter kann man nun z. B. wie folgt erzeugen:

$$S \Rightarrow \neg S \Rightarrow \neg(S \wedge S) \Rightarrow \neg(VI \wedge VI) \Rightarrow \neg(VI \wedge VII) \Rightarrow \neg(pI \wedge pII) \Rightarrow \neg(p_1 \wedge p_2)$$

Solche Grammatiken notiert man (im Kontext der Logik) auch wie folgt:

$$\varphi ::= 0 \mid 1 \mid p \mid \neg p \mid \varphi \wedge \varphi \mid \varphi \vee \varphi \mid \varphi \to \varphi \mid \varphi \leftrightarrow \varphi$$

In der Informatik ist diese Form von Grammatiken als *erweiterte Bachus-Naur-Form* (EBNF) bekannt.

Klammerregeln Durch folgende Regeln sparen wir Klammern ein:

- Äußere Klammern werden weggelassen.
- $\neg$ bindet am stärksten,
- $\wedge$ bindet stärker als $\vee$ und
- $\vee$ wiederum bindet stärker als $\to$ und $\leftrightarrow$.

– Für mehrfache Konnektoren desselben Typs werden die Klammern von rechts nach links gesetzt, z.B. $p \to q \to r$ entspricht $(p \to (q \to r))$.

Zum Beispiel kann die Formel

$$((p_1 \vee (p_2 \vee p_3)) \leftrightarrow (p_1 \wedge \neg p_2))$$

verkürzt als

$$p_1 \vee p_2 \vee p_3 \leftrightarrow p_1 \wedge \neg p_2$$

geschrieben werden. Für $((p_1 \vee p_2) \wedge p_3)$ dürfen wir aber nicht $p_1 \vee p_2 \wedge p_3$ schreiben.

Zusätzlich verwenden wir noch die folgenden Abkürzungen:

$$\bigwedge_{i=1}^{n} \varphi_i \quad \text{für} \quad \varphi_1 \wedge \cdots \wedge \varphi_n \quad \text{und}$$
$$\bigvee_{i=1}^{n} \varphi_i \quad \text{für} \quad \varphi_1 \vee \cdots \vee \varphi_n.$$

Definition 2. Die Menge $\mathsf{sub}(\varphi)$ aller *Teilformeln* einer aussagenlogischen Formel φ ist induktiv wie folgt definiert: Teilformel $\mathsf{sub}(\varphi)$

(i) $\mathsf{sub}(0) = \{0\}$ und $\mathsf{sub}(1) = \{1\}$.

(ii) Für $p \in \mathsf{Var}$ ist $\mathsf{sub}(p) = \{p\}$.

(iii) Für $\varphi \in \mathsf{Form}_{\mathrm{AL}}$ ist $\mathsf{sub}(\neg\varphi) = \{\neg\varphi\} \cup \mathsf{sub}(\varphi)$.

(iv) Für $\varphi, \psi \in \mathsf{Form}_{\mathrm{AL}}$ und $\circ \in \{\wedge, \vee, \to, \leftrightarrow\}$ ist

$$\mathsf{sub}(\varphi \circ \psi) = \{\varphi \circ \psi\} \cup \mathsf{sub}(\varphi) \cup \mathsf{sub}(\psi) \ .$$

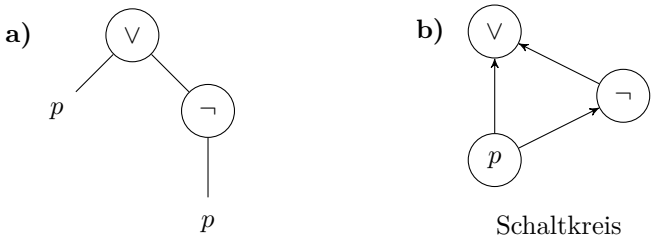
Beispiel 3. $\varphi = p \vee \neg p$

$$\begin{aligned} \mathsf{sub}(\varphi) &= \{p \vee \neg p\} \cup \mathsf{sub}(p) \cup \mathsf{sub}(\neg p) \\ &= \{p \vee \neg p, p\} \cup \{\neg p\} \cup \mathsf{sub}(p) \\ &= \{p \vee \neg p, p, \neg p\} \end{aligned}$$

Für eine aussagenlogische Formel φ definieren wir $\mathsf{Var}(\varphi)$ als die Menge ihrer aussagenlogischen Variablen, d.h. $\mathsf{Var}(\varphi) = \mathsf{Var} \cap \mathsf{sub}(\varphi)$. $\mathsf{Var}(\varphi)$

$$\mathsf{Var}(\varphi) = \{p\} \cap \{p \vee \neg p, p, \neg p\} = \{p\}$$

Diese Formel φ kann auch grafisch dargestellt werden:



Schaltkreis

Eine Darstellung wie in a) nennen wir *Ableitungsbaum*.

Ableitungsbaum **Definition 4.** Der *Ableitungsbaum* einer Formel φ ist wie folgt definiert:

$$T(\varphi) = \bullet\varphi \qquad \text{für } \varphi = p,\ p \text{ Variable.}$$

$$T\big((\varphi \circ \psi)\big) = \begin{array}{ccc} & & (\varphi \circ \psi) \\ & \diagup\ \diagdown & \\ T(\varphi) & & T(\psi) \end{array}$$

$$T\big((\neg\varphi)\big) = \begin{array}{cc} & (\neg\varphi) \\ \big| & \\ T(\varphi) & \end{array}$$

Übungsaufgabe 5.

(i) Zeichnen Sie die Ableitungsbäume zu folgenden Formeln:

a) $(\neg p_2 \to (p_3 \vee (p_1 \leftrightarrow p_2))) \wedge \neg p_3$

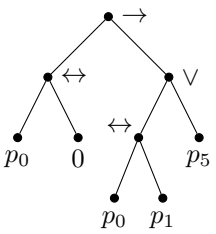
b) $(p_7 \to \neg 0) \leftrightarrow ((p_4 \wedge \neg p_2) \to p_1$

c) $(((p_1 \to p_2) \to p_1) \to p_2) \to p_1$

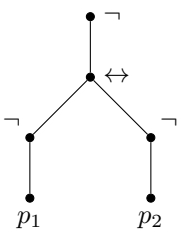
(ii) Bestimmen Sie die den folgenden Ableitungsbäumen entsprechenden Formeln:

a)

b)



c)



Boole'scher Schaltkreis **Definition 6.** Ein Boole'scher Schaltkreis mit n Eingaben ist ein 3-Tupel

$$G = (V, E, \beta),$$

wobei (V, E) ein endlicher (gerichteter) azyklischer Graph ist und

$$\beta : V \to \{p_1, \ldots, p_n, 0, 1, \neg, \wedge, \vee\}$$

eine Markierung der Knoten des Graphen mit elementaren Formeln oder Konnektoren darstellt, sodass die folgenden Bedingungen gelten:

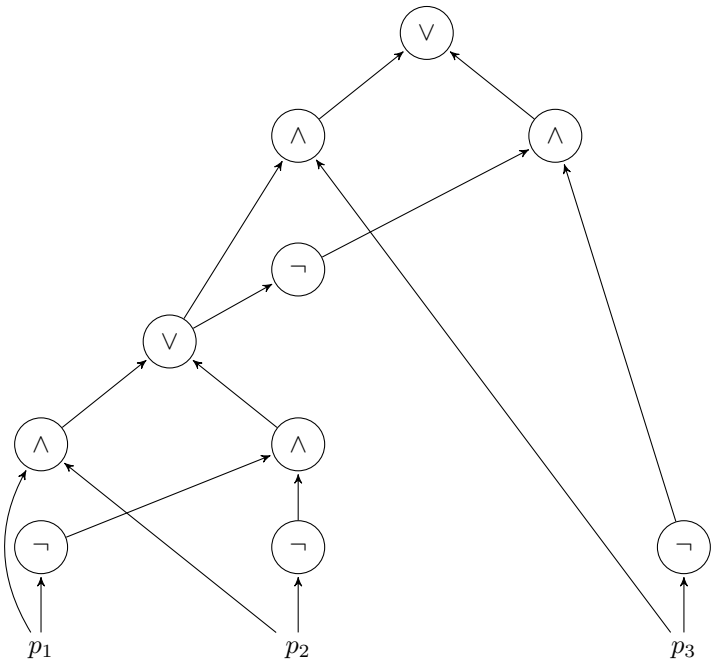
(i) Wenn $v \in V$ Eingangsgrad 0 hat, dann gilt $\beta(v) \in \{p_1, \ldots, p_n, 0, 1\}$.

(ii) Wenn $v \in V$ den Eingangsgrad 1 hat, dann ist $\beta(v) = \neg$.

(iii) Ansonsten hat v den Eingangsgrad 2 und es ist $\beta(v) \in \{\wedge, \vee\}$.

(iv) Es gibt genau einen Knoten mit Ausgangsgrad 0, die sogenannte *Ausgabe*.

Übungsaufgabe 7. Gegeben ist folgender Schaltkreis C.



(i) Welche Formel stellt C dar?

(ii) Geben Sie eine Wahrheitstafel für C an.

1.2. Die Semantik der Aussagenlogik

Unsere – bisher rein syntaktisch definierten – aussagenlogischen Formeln sind für sich noch bedeutungslos. Dies ändern wir, indem wir ihnen Wahrheitswerte zuweisen.

Definition 8. Eine *aussagenlogische Belegung* $\mathfrak{I}$ ist eine Abbildung

$$\mathfrak{I}\colon \mathsf{Var} \to \{0,1\}.$$

Belegung

Eine Belegung $\mathfrak{I}$ kann zu einer Abbildung

$$\hat{\mathfrak{I}}\colon \mathsf{Form}_{\mathrm{AL}} \to \{0,1\}$$

erweitert werden vermöge von:

(i) $\hat{\mathfrak{I}}(p) = \mathfrak{I}(p)$ für $p \in \mathsf{Var}$.

(ii) $\hat{\mathfrak{I}}(0) = 0$ und $\hat{\mathfrak{I}}(1) = 1$.

(iii) $\hat{\mathfrak{I}}(\neg\varphi) = \begin{cases} 1 & \text{wenn } \hat{\mathfrak{I}}(\varphi) = 0,\ \varphi \in \mathsf{Form}_{\mathrm{AL}} \\ 0 & \text{sonst} \end{cases}$

(iv) $\hat{\mathfrak{I}}(\varphi \wedge \psi) = \begin{cases} 1 & \text{wenn } \hat{\mathfrak{I}}(\varphi) = \hat{\mathfrak{I}}(\psi) = 1,\ \varphi, \psi \in \mathsf{Form}_{\mathrm{AL}} \\ 0 & \text{sonst} \end{cases}$

(v) $\hat{\mathfrak{I}}(\varphi \vee \psi) = \begin{cases} 1 & \text{wenn } \hat{\mathfrak{I}}(\varphi) = 1 \text{ oder } \hat{\mathfrak{I}}(\psi) = 1,\ \varphi, \psi \in \mathsf{Form}_{\mathrm{AL}} \\ 0 & \text{sonst} \end{cases}$

(vi) Für $\varphi, \psi \in \mathsf{Form}_{\mathrm{AL}}$ ergibt sich für die Konnektoren $\circ \in \{\rightarrow, \leftrightarrow\}$ die Definition von $\hat{\mathfrak{I}}(\varphi \circ \psi)$ aus nachfolgender Tabelle, in der wir auch die Definitionen für die schon behandelten Konnektoren wiederholen

$\hat{\mathfrak{I}}(\varphi)$	$\hat{\mathfrak{I}}(\psi)$	$\hat{\mathfrak{I}}(\neg\varphi)$	$\hat{\mathfrak{I}}(\varphi \wedge \psi)$	$\hat{\mathfrak{I}}(\varphi \vee \psi)$	$\hat{\mathfrak{I}}(\varphi \rightarrow \psi)$	$\hat{\mathfrak{I}}(\varphi \leftrightarrow \psi)$
0	0	1	0	0	1	1
0	1	1	0	1	1	0
1	0	0	0	1	0	0
1	1	0	1	1	1	1

Um die Notation zu vereinfachen, schreiben wir meist nur $\mathfrak{I}$ statt $\hat{\mathfrak{I}}$.

Belegung für eine Formel φ

Wir nennen $\mathfrak{I}$ eine *Belegung für eine Formel* φ (oder passend für φ), falls $\mathfrak{I}$ eine Abbildung

$$\mathfrak{I} : \mathsf{Var}(\varphi) \rightarrow \{0, 1\}$$

ist, die zu einer gewöhnlichen Belegung erweitert werden kann, indem $\mathfrak{I}$ für $\mathsf{Var} \setminus \mathsf{Var}(\varphi)$ beliebig definiert wird.

erfüllende Belegung

$\mathfrak{I}$ ist eine *erfüllende Belegung* für eine Formel φ, falls $\hat{\mathfrak{I}}(\varphi) = 1$. Hierfür schreiben wir auch

$$\mathfrak{I} \vDash \varphi$$

Modell

und nennen $\mathfrak{I}$ ein *Modell* für φ. Ist Φ eine Formelmenge, so schreiben wir

$$\mathfrak{I} \vDash \Phi \ ,$$

$\vDash$

falls $\mathfrak{I} \vDash \varphi$ für alle $\varphi \in \Phi$ gilt.

Alfred Tarski (1901–1983) polnisch-amerikanischer Mathematiker und Logiker

Die hier präsentierte und heute allgemein-übliche Definition der Semantik in der Logik wurde Alfred Tarski im Jahre 1936 entwickelt und wird dementsprechend als „Tarski-Semantik“ bezeichnet.Da der Wahrheitswert zusammengesetzter Formeln sich aus den Wahrheitswerter der Teilformeln ergibt, spricht man auch von *kompositionaler Semantik*.

kompositionale Semantik

Koinzidenzlemma

Lemma 9. Der Wert einer aussagenlogischen Formel φ unter einer Belegung $\mathfrak{I}$ hängt nur von der Belegung der in φ auftretenden aussagenlogischen Variablen ab, d.h. sind $\mathfrak{I}_1$ und $\mathfrak{I}_2$ zwei Belegungen mit $\mathfrak{I}_1(p) = \mathfrak{I}_2(p)$ für alle $p \in \mathsf{Var}(\varphi)$, so gilt $\mathfrak{I}_1(\varphi) = \mathfrak{I}_2(\varphi)$.

Übungsaufgabe 10. Beweisen Sie das Koinzidenzlemma durch vollständige Induktion über den Formelaufbau.

Definition 11. Seien $\Phi, \Psi \subseteq \mathsf{Form}_{\mathrm{AL}}$. Dann folgt Ψ aus Φ, symbolisch

$$\Phi \vDash \Psi,$$

falls für alle Belegungen $\mathfrak{I}$ mit $\mathfrak{I} \vDash \Phi$ auch $\mathfrak{I} \vDash \Psi$ gilt. Sind φ und ψ aussagenlogische Formeln, so schreiben wir statt $\{\varphi\} \vDash \{\psi\}$ auch einfach $\varphi \vDash \psi$. Gilt sowohl $\varphi \vDash \psi$ als auch $\psi \vDash \varphi$, so heißen φ und ψ *semantisch äquivalent*. Wir schreiben dann auch $\varphi \equiv \psi$.

semantisch äquivalent

Im nächsten Satz fassen wir wichtige Äquivalenzen zusammen:

Satz 12. Für beliebige aussagenlogische Formeln φ, ψ und θ gelten die folgenden Äquivalenzen: Äquivalenzen

- $\varphi \wedge \varphi \equiv \varphi$ und $\varphi \vee \varphi \equiv \varphi$ (Idempotenz)
- $\varphi \wedge \psi \equiv \psi \wedge \varphi$ und $\varphi \vee \psi \equiv \psi \vee \varphi$ (Kommutativität)
- $(\varphi \wedge \psi) \wedge \theta \equiv \varphi \wedge (\psi \wedge \theta)$ und $(\varphi \vee \psi) \vee \theta \equiv \varphi \vee (\psi \vee \theta)$ (Assoziativität)
- $(\varphi \wedge \psi) \vee \theta \equiv (\varphi \vee \theta) \wedge (\psi \vee \theta)$ und
  $(\varphi \vee \psi) \wedge \theta \equiv (\varphi \wedge \theta) \vee (\psi \wedge \theta)$ (Distributivität)
- $\neg\neg\varphi \equiv \varphi$ (Doppelnegation)
- $\neg(\varphi \wedge \psi) \equiv \neg\varphi \vee \neg\psi$ und $\neg(\varphi \vee \psi) \equiv \neg\varphi \wedge \neg\psi$ (de Morgansche Regeln)
- $\varphi \to \psi \equiv \neg\varphi \vee \psi$ (Auflösen der Implikation)
- $\varphi \leftrightarrow \psi \equiv (\varphi \to \psi) \wedge (\psi \to \varphi)$ (Auflösen der Biimplikation)
- $\varphi \vee (p \wedge \psi) \equiv \varphi$ und $\varphi \wedge (\varphi \vee \psi) \equiv \varphi$ (Absorption)

Übungsaufgabe 13. Beweisen Sie Satz 12 durch Aufstellen von Wahrheitstafeln.

Bemerkung. $\varphi \wedge \psi$ und $\psi \wedge \varphi$ sind semantisch äquivalent, d.h. $\varphi \wedge \psi \equiv \psi \wedge \varphi$. Es gilt jedoch $\varphi \wedge \psi \neq \psi \wedge \varphi$, da es sich syntaktisch um zwei verschiedene Wörter der Sprache $\mathsf{Form}_{\mathrm{AL}}$ handelt.

Übungsaufgabe 14. (NAND-/Sheffer-Funktion) Wir erweitern den aussagenlogischen Formelbegriff der Vorlesung, indem wir zur Definition hinzufügen: Sind φ und ψ Formeln, so ist auch $\varphi \uparrow \psi$ eine Formel. Für zu Formeln φ und ψ passende Belegungen $\mathfrak{I}$ wird $\mathfrak{I}(\varphi \uparrow \psi)$ definiert durch NAND-/Sheffer-Funktion

$\mathfrak{I}(\varphi)$	$\mathfrak{I}(\psi)$	$\mathfrak{I}(\varphi \uparrow \psi)$
0	0	1
0	1	1
1	0	1
1	1	0

Zeigen Sie, dass es zu jeder Formel φ eine äquivalente Formel ψ gibt, die nur den Operator $\uparrow$ enthält.

In aussagenlogischen Formeln dürfen Teilformeln beliebig durch andere, logisch äquivalente Teilformeln ersetzt werden, ohne dass sich die Erfüllbarkeit bzw. Gültigkeit der Formel ändert. Dies sagt der folgende Satz:

Satz 15. Seien φ und ψ äquivalente Formeln. Die Formel σ' gehe aus der Formel σ hervor, indem einige Vorkommen von φ in σ durch ψ ersetzt werden. Dann gilt $\sigma \equiv \sigma'$. Äquivalenzsatz, Ersetzbarkeitstheorem

Übungsaufgabe 16. Beweisen Sie Satz 15 induktiv über den Aufbau von σ.

Übungsaufgabe 17. Sei φ wie in Aufgabe 37. Sind φ und

$$\psi := (\neg x \vee y \vee \neg z) \wedge (\neg x \vee \neg y \vee z) \wedge (\neg x \vee z) \wedge (\neg y \vee z)$$

(semantisch) äquivalent?

(i) Verifizieren oder widerlegen Sie dies durch eine Wahrheitstafel.

(ii) Falls $\varphi \equiv \psi$ gilt, so verifizieren Sie dies durch geeignete Äquivalenzumformungen.

Übungsaufgabe 18. Beweisen Sie die allgemeinen *de Morgan'schen Regeln*:

$$\neg \bigwedge_{i=1}^{k} \varphi_1 \equiv \bigvee_{i=1}^{k} \neg\varphi_1 \ , \quad \neg \bigvee_{i=1}^{k} \varphi_1 \equiv \bigwedge_{i=1}^{k} \neg\varphi_1$$

1.3. Normalformen

Formeln lassen sich oft einfacher behandeln, wenn sie in einer bestimmten syntaktischen (Normal-)Form sind. In der Aussagenlogik interessieren besonders die konjunktive und die disjunktive Normalform.

Literal
Klausel

Definition 19. *Literale* sind negierte oder unnegierte Variablen. Eine Klausel ist eine endliche Menge von Literalen.

konjunktive Normalform, KNF

Definition 20. Eine Formel φ ist in *konjunktiver Normalform* (KNF), falls

$$\varphi = \bigwedge_{i=1}^{n} \bigvee_{j=1}^{m_i} L_{ij}$$

mit Literalen L_{ij}.

Alternativ fassen wir φ als eine Menge $\{C_1, \ldots, C_n\}$ von Klauseln auf mit

$$C_i = \{L_{ij} \mid j = 1, \ldots, m_i\}.$$

Beispiel 21. Die Formel $\gamma = (p_1 \vee p_2) \wedge (p_1 \vee \neg p_2) \wedge (\neg p_1 \vee p_2) \wedge (\neg p_1 \vee \neg p_2)$ in KNF kann wie folgt als Klauselmenge dargestellt werden:

$$\Gamma = \big\{\{p_1, p_2\}, \{p_1, \neg p_2\}, \{\neg p_1, p_2\}, \{\neg p_1, \neg p_2\}\big\}$$

disjunktive Normalform, DNF

Definition 22. Eine Formel φ ist in *disjunktiver Normalform* (DNF), falls

$$\varphi = \bigvee_{i=1}^{n} \bigwedge_{j=1}^{m_i} L_{ij}$$

mit Literalen L_{ij}.

Die Modellbeziehung $\vDash$ übertragen wir folgendermaßen auf Klauseln und Klauselmengen. Eine Klausel C wird durch eine Belegung $\mathfrak{I}$ erfüllt, falls $\mathfrak{I}$ mindestens ein Literal aus C erfüllt, wofür wir wieder $\mathfrak{I} \vDash C$ schreiben. Eine Klauselmenge $\Gamma = \{C_1, \ldots, C_n\}$ wird durch $\mathfrak{I}$ erfüllt (Schreibweise $\mathfrak{I} \vDash \Gamma$), falls $\mathfrak{I} \vDash C_i$ für $i = 1, \ldots, n$. Die leere Klausel wird mit $\square$ bezeichnet und ist unerfüllbar. (Statt $\square$ findet man auch die Notationen $\sqcup$, $\bot$, $\{\}$ oder $\emptyset$.)

leere Klausel

Satz 23. Jede aussagenlogische Formel ist äquivalent zu einer aussagenlogischen Formel in KNF sowie zu einer aussagenlogischen Formel in DNF.

Der Beweis des Satzes ergibt sich zusammen mit dem Äquivalenzsatz (Satz 15) aus dem folgenden algorithmischen Verfahren zum Umformen einer beliebigen aussagenlogischen Formel in eine äquivalente Formel in KNF:

Eingabe: eine Formel φ

1: Ersetze in φ jedes Vorkommen einer Teilformel der Form

$\psi \to \theta$ durch $\neg\psi \vee \theta$

$\psi \leftrightarrow \theta$ durch $(\psi \wedge \theta) \vee (\neg\psi \wedge \neg\theta)$

bis keine derartige Teilformel mehr vorkommt.

2: Ersetze jedes Vorkommen einer Teilformel der Form

$\neg\neg\psi$ durch ψ

$\neg(\psi \wedge \theta)$ durch $\neg\psi \vee \neg\theta$

$\neg(\psi \vee \theta)$ durch $\neg\psi \wedge \neg\theta$

bis keine derartige Teilformel mehr vorkommt.

3: Ersetze jedes Vorkommen einer Teilformel der Form

$\psi \vee (\sigma \wedge \theta)$ durch $(\psi \vee \sigma) \wedge (\psi \vee \theta)$

$(\sigma \wedge \theta) \vee \psi$ durch $(\sigma \vee \psi) \wedge (\theta \vee \psi)$

bis keine derartige Teilformel mehr vorkommt.

Analog kann man einen Algorithmus zum Umformen in DNF angeben.

Übungsaufgabe 24. Geben Sie jeweils eine äquivalente KNF und DNF an:

(i) $\varphi_1 := p_1 \leftrightarrow (p_2 \to \neg p_3)$

(ii) $\varphi_2 := (p_1 \to p_2) \vee (p_1 \to p_3)$

(iii) $\varphi_3 := q_1 \wedge (q_2 \leftrightarrow q_3)$

(iv) $\varphi_4 := (x_2 \leftrightarrow x_3) \vee (x_1 \vee x_3)$

Eine zweite Möglichkeit für die Erstellung einer DNF ist das direkte Ablesen aus der Wahrheitstafel. Für jede Zeile der Wahrheitstafel die den Wert 1 hat, gibt es ein Disjunktionsglied. Ist p_i in der Zeile mit 1 belegt, so enthält das Disjunktionsglied (das selbst eine Konjunktion ist) das Literal p_i. Andernfalls das Literal $\neg p_i$.

Für die Erstellung einer KNF aus der Wahrheitstafel, werden die Zeilen mit Wert 0 als Konjunktionsglieder kodiert. Ist p_i in der Zeile mit 0 belegt, so enthält das Konjunktionsglied (das selbst eine Disjunktion ist) das Literal p_i. Andernfalls das Literal $\neg p_i$.

Beispiel 25. Gegeben sei folgende Wahrheitstafel für $\varphi := (p_1 \vee p_2) \wedge \neg p_3$:

p_1	p_2	p_3	$\varphi(p_1, p_2, p_3)$
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	0

Dann lassen sich folgende DNF und KNF ablesen:

$$\varphi_{\text{DNF}} = (\neg p_1 \wedge p_2 \wedge \neg p_3) \vee (p_1 \wedge \neg p_2 \wedge \neg p_3) \vee (p_1 \wedge p_2 \wedge \neg p_3)$$

$$\varphi_{\text{KNF}} = (p_1 \vee p_2 \vee p_3) \wedge (p_1 \vee p_2 \vee \neg p_3) \wedge (p_1 \vee \neg p_2 \vee \neg p_3) \\ \wedge (\neg p_1 \vee p_2 \vee \neg p_3) \wedge (\neg p_1 \vee \neg p_2 \vee \neg p_3)$$

Etwas formaler: Aus der Wahrheitstafel einer Formel φ lässt sich eine äquivalente DNF φ_{DNF} aufstellen, indem man die folgende Disjunktion

$$\bigvee_{\mathfrak{I} \vDash \varphi} p_1^{\mathfrak{I}(p_1)} \wedge \ldots \wedge p_n^{\mathfrak{I}(p_n)}$$

aus Literalen $p_i^1 = p_i$, $p_i^0 = \neg p_i$ entsprechend der Belegungen $\mathfrak{I}$, die φ wahr werden lassen, bildet.

Übungsaufgabe 26. Gegeben sei folgende Wahrheitstafel:

x_1	x_2	x_3	$\varphi(x_1, x_2, x_3)$
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	1

Lesen Sie hieraus eine DNF und eine KNF ab.

Übungsaufgabe 27. Formalisieren Sie das Verfahren für KNF.

Wir haben also gesehen, dass sich jede Wahrheitstafel durch eine aussagenlogische Formel in KNF oder DNF beschreiben lässt. Das ist die Motivation für die folgende Definition:

funktional vollständig

Definition 28. Sei B eine endliche Menge von logischen Konnektoren. Solche Mengen heißen *Basis*. B heißt *funktional vollständig*, falls sich jede Wahrheitstafel durch eine aussagenlogische Formel beschreiben lässt, die nur Konnektoren aus B enthält.

Beispiele für funktional vollständige Mengen sind also

- $\{\wedge, \vee, \neg\}$ (wegen KNF- oder DNF-Normalform)
- $\{\uparrow\}$ (siehe Übungsaufgabe 14)

Um zu zeigen, dass eine Menge B funktional vollständig ist, muss man also zeigen, dass alle aussagenlogischen Konnektoren oder einfacher die aus einer anderen vollständigen Menge mit den Konnektoren aus B simuliert werden können.

Daraus ergibt sich zum Beispiel, dass $\{\wedge, \neg\}$ und $\{\vee, \neg\}$ funktional vollständig sind (nach den Regeln von de Morgan).

Es ist nicht möglich, beliebige Formeln effizient in KNF oder DNF umzuformen. Dies hat folgenden Grund: Es existiert eine Folge von Formeln φ_n mit $2n$ Literalen, so dass φ_n polynomiell groß ist in n, jedoch jede zu φ_n äquivalente Formel ψ_n in KNF mindestens 2^n Klauseln besitzt.

Betrachte z. B. die Formeln

$$\varphi_n = \bigvee_{i=1}^{n} p_{i,1} \wedge p_{i,2}.$$

Die zu φ_n minimale äquivalente Formel in KNF ist

$$\psi_n = \bigwedge_{j_1,\ldots,j_n \in \{1,2\}} p_{1,j_1} \vee \cdots \vee p_{n,j_n}.$$

Die Formel enthält genau 2^n viele Klauseln.

Korollar 29. Es gibt keinen effizienten Algorithmus (d.h. mit polynomieller Laufzeit), der beliebige aussagenlogische Formeln in äquivalente Formeln in KNF umformt.

Oft werden auch folgende noch striktere Normalformen betrachtet.

Definition 30. Eine Klausel φ heißt *k-Klausel*, wenn φ höchstens k Literale enthält. k-Klausel

Definition 31. Eine Formel φ ist in *k-KNF* genau dann, wenn φ eine Konjunktion von k-Klauseln ist. k-KNF

Aus unserer Definition für die k-KNF ergibt sich für $k = 3$ folgender Spezialfall:

Definition 32. Eine Formel ist in 3-KNF genau dann, wenn φ sie in KNF ist und jede Disjunktion genau drei Literale hat.

Neben der semantischen Äquivalenz gibt es weitere Äquivalenzen, wie z. B. die *Erfüllbarkeitsäquivalenz*:

Definition 33. Zwei Formeln φ und ψ heißen *erfüllbarkeitsäquivalent*, symbolisch $\varphi \overset{\text{sat}}{\approx} \psi$, genau dann, wenn gilt:

$$\varphi \text{ ist erfüllbar} \Leftrightarrow \psi \text{ ist erfüllbar.}$$

Beispiel 34. $\varphi := p_1$, $\psi := p_2 \to p_3$ sind beide erfüllbar. Also: $\varphi \overset{\text{sat}}{\approx} \psi$

Satz 35. Sei φ eine Formel in KNF und φ' die durch Algorithmus 1 konstruierte 3-KNF. φ und φ' sind erfüllbarkeitsäquivalent.

Eingabe: KNF-Formel φ
Ausgabe: 3-KNF-Formel φ'

```
/*
Für jede Disjunktion C_i = l_i^1 ∨ l_i^2 ∨ ... ∨ l_i^{n_i}
wird eine geeignete Transformation durchgeführt.
Diese ist abhängig vom Wert von n_i:
*/
```

1: **if** $n_i = 1$ **then**
2: Bilde zwei neue Variablen p_i^1, p_i^2 und ersetze C_i durch:
$(l_i^1 \vee p_i^1 \vee p_i^2) \wedge (l_i^1 \vee \neg p_i^1 \vee p_i^2) \wedge (l_i^1 \vee p_i^1 \vee \neg p_i^2) \wedge (l_i^1 \vee \neg p_i^1 \vee \neg p_i^2)$
3: **else if** $n_i = 2$ **then**
4: Bilde eine neue Variable p_i^1 und ersetze C_i durch:
$(l_i^1 \vee l_i^2 \vee p_i^1) \wedge (l_i^1 \vee l_i^2 \vee \neg p_i^1)$
5: **else if** $n_i = 3$ **then**
6: Führe keine Veränderung durch.
7: **else if** $n_i > 3$ **then**
8: Bilde $n-3$ neue Variablen $p_i^1, p_i^2, \ldots, p_i^{n-3}$ und ersetze C_i durch:
$(l_i^1 \vee l_i^2 \vee p_i^1) \wedge (\neg p_i^1 \vee l_i^3 \vee p_i^2) \wedge (\neg p_i^2 \vee l_i^4 \vee p_i^3) \wedge \ldots \wedge (\neg p_i^{n-3} \vee l_i^{n-1} \vee l_i^n)$
9: **end if**

Algorithmus 1: Überführung einer Formel in 3-KNF

1.4. Das Erfüllbarkeitsproblem

SAT

Die Menge aller erfüllbaren Formeln wird bezeichnet mit

$$\mathrm{SAT} = \{\varphi \in \mathsf{Form}_{\mathrm{AL}} \mid \text{es gibt eine Belegung } \mathfrak{I} \text{ mit } \mathfrak{I} \vDash \varphi\}.$$

allgemeingültig
Tautologie

Eine Formel φ heißt *allgemeingültig* oder *Tautologie*, falls sie durch alle Belegungen erfüllt wird. Wir verwenden die Bezeichnung $\vDash \varphi$. Die Menge aller Tautologien ist

$$\mathrm{TAUT} = \{\varphi \in \mathsf{Form}_{\mathrm{AL}} \mid \vDash \varphi\}.$$

Stephen C. Cook (*1939)
Richard Karp(*1935)
amerikanische Informatiker

Ein klassisches Resultat von Cook (1971) und Karp (1972) besagt, dass SAT NP-vollständig und TAUT coNP-vollständig ist.

Bemerkung. Es gilt $\varphi \notin \mathrm{SAT}$ genau dann, wenn $\neg\varphi \in \mathrm{TAUT}$.

Beweis

$$\begin{aligned} \varphi \notin \mathrm{SAT} &\iff \text{Für alle Belegungen } \mathfrak{I} \text{ gilt } \mathfrak{I} \nvDash \varphi \\ &\iff \text{Für alle Belegungen } \mathfrak{I} \text{ gilt } \mathfrak{I} \vDash \neg\varphi \\ &\iff \neg\varphi \in \mathrm{TAUT} \end{aligned}$$

■

Mit Hilfe von Wahrheitstafeln lässt sich leicht testen, ob eine gegebene Formel φ eine Tautologie ist: φ wird unter allen Belegungen ausgewertet.

Beispiel 36. $\varphi := (p \to q) \wedge (q \to r) \to (p \to r)$

p	q	r	$p \to q$	$q \to r$	$p \to r$	φ
0	0	0	1	1	1	1
0	0	1	1	1	1	1
0	1	0	1	0	0	1
0	1	1	1	1	1	1
1	0	0	0	1	0	1
1	0	1	0	1	1	1
1	1	0	1	0	0	1
1	1	1	1	1	1	1

Auf gleiche Weise kann die Erfüllbarkeit von φ getestet werden. Leider ist dieser Algorithmus nicht effizient.

Um die Wahrheitstafelmethode für eine Formel mit n Variablen durchzuführen, muss man 2^n Belegungen untersuchen. Hat die Formel also 100 Variablen, und solche Formeln treten häufig bei automatisch generierten Prozessen auf, so sind das 2^{100} Belegungen. Angenommen, ein Computer kann eine Milliarde solcher Operationen pro Sekunde überprüfen, so ergibt sich für 2^{100} Belegungen immer noch eine Rechenzeit von mehr als 10^{13} Jahren. Dieser Algorithmus für SAT ist also praktisch nicht durchführbar. Wesentlich bessere Algorithmen, d.h. mit subexponentieller Laufzeit, sind bislang nicht bekannt. Gerade für SAT-Solver gibt es einen aktiven Wettbewerb um die schnellsten Verfahren. Die beste Laufzeit für 3-SAT (mittels eines probabilistischen Algorithmus) liegt derzeit bei $1{,}32216^n$. Somit ist SAT mit heutigen Methoden zwar effektiv, aber nicht effizient lösbar.

Übungsaufgabe 37. Gegeben sei die Formel $\varphi := x \vee y \to (z \leftrightarrow y) \wedge (x \to z)$. Untersuchen Sie mit Hilfe einer Wahrheitstafel, ob φ erfüllbar, unerfüllbar oder eine Tautologie ist. Falls φ erfüllbar ist, so geben Sie ein Modell an.

1.5. Der Endlichkeitssatz

Ein zentrales Resultat bzgl. der Semantik der Aussagenlogik ist der folgende Endlichkeitssatz.

Satz 38. Eine Menge Φ aussagenlogischer Formeln ist erfüllbar genau dann, wenn jede endliche Teilmenge $\Phi_0 \subseteq \Phi$ erfüllbar ist. Endlichkeitssatz

Beweis „⇒“: Ergibt sich unmittelbar aus der Tatsache, dass jedes Modell von Φ nach Definition auch ein Modell jeder Teilmenge von Φ ist.

„⇐“: Wir nehmen an, dass jede endliche Teilmenge von Φ erfüllbar ist. Die Menge der Variablen von Φ sei $\{p_1, p_2, \dots\}$. Ferner sei für $i \in \mathbb{N}$

$$\Phi_i = \{\varphi \in \Phi \mid \mathsf{Var}(\varphi) \subseteq \{p_1, \dots, p_i\}\}.$$

Zwar können die Mengen Φ_i unendlich viele Formeln enthalten, jedes Φ_i enthält aber nur endlich viele nicht äquivalente Formeln (genauer: höchstens 2^{2^i} viele, Beweis siehe

Übungsaufgabe 39). Laut Voraussetzung besitzt also jede Menge Φ_i ein Modell $\mathfrak{I}_i$. Da für $j \leq i$ offenbar $\Phi_j \subseteq \Phi_i$ gilt, ist $\mathfrak{I}_i$ sogar ein Modell für $\bigcup_{j \leq i} \Phi_j$.

Unsere Aufgabe besteht nun darin, aus diesen unendlich vielen, und im allgemeinen natürlich verschiedenen Belegungen eine erfüllende Belegung für die gesamte Menge Φ zu konstruieren. Dies geschieht in einer stufenweise Konstruktion gemäß Algorithmus 2.

```
1:  I := ℕ
2:  ℑ := ℑ_1
3:  for i := 1 to ∞ do
4:      if es gibt unendlich viele Indizes j ∈ I mit ℑ_j(p_i) = 0 then
5:          ℑ(p_i) := 0
6:          I := I \ {j | ℑ_j(p_i) = 1}
7:      else
8:          ℑ(p_i) := 1
9:          I := I \ {j | ℑ_j(p_i) = 0}
10:     end if
11: end for
```

Algorithmus 2: Konstruktion von $\mathfrak{I}$

Dort wird die Belegung $\mathfrak{I}$ schrittweise konstruiert, wobei im i-ten Durchlauf der **for**-Schleife der Wert $\mathfrak{I}(p_i)$ festgelegt wird. Die Indexmenge I gibt zu jedem Zeitpunkt die Menge der Belegungen aus $\{\mathfrak{I}_j \mid j \in \mathbb{N}\}$ an, die mit den bislang konstruierten Einträgen von $\mathfrak{I}$ konsistent sind, d.h. für die im i-ten Durchlauf

$$\mathfrak{I}_j(p_1) = \mathfrak{I}(p_1) \quad \ldots \quad \mathfrak{I}_j(p_i) = \mathfrak{I}(p_i)$$

mit $j \in I$ gilt. Dabei wird I natürlich in jedem Durchlauf der **for**-Schleife kleiner, bleibt aber nach Konstruktion stets unendlich.

Zu zeigen ist noch, dass die so konstruierte Belegung $\mathfrak{I}$ tatsächlich ein Modell für Φ ist. Sei dazu $\varphi \in \Phi$. Wir wählen ein $i \in \mathbb{N}$, so dass $\mathsf{Var}(\varphi) \subseteq \{p_1, \ldots, p_i\}$ gilt. Zwar muss $\mathfrak{I}$ nicht mit $\mathfrak{I}_i$ auf $\{p_1, \ldots, p_i\}$ übereinstimmen, nach Konstruktion gibt es aber unendlich viele Indizes $j \geq i$, für die

$$\mathfrak{I}_j(p_1) = \mathfrak{I}(p_1) \quad \ldots \quad \mathfrak{I}_j(p_i) = \mathfrak{I}(p_i)$$

gilt. Nach Voraussetzung gilt $\mathfrak{I}_j \models \varphi$, und mit dem Koinzidenzlemma folgt dann auch $\mathfrak{I} \models \varphi$. ■

Übungsaufgabe 39. Sei $\Phi_i := \{\varphi \in \mathsf{Form}_{\mathrm{AL}} \mid \mathsf{Var}(\varphi) \subseteq \{p_1, \ldots, p_i\}\}$. Zeigen Sie: Φ_i enthält genau 2^{2^i} verschiedene, paarweise nicht-äquivalente Formeln.

Übungsaufgabe 40. Sei $M = \{\varphi_n \mid n \in \mathbb{N}\}$ eine Formelmenge, so dass für alle natürlichen Zahlen n gilt: $\models \varphi_{n+1} \to \varphi_n$ und $\not\models \varphi_n \to \varphi_{n+1}$.

(i) Zeigen Sie mit Hilfe des Endlichkeitssatzes: M hat ein Modell.

(ii) Geben Sie ein Beispiel für eine solche Formelmenge an.

(Zur Erinnerung: $\models \psi$ bedeutet „ψ ist Tautologie“. Entsprechend bedeutet $\not\models \psi$ „ψ ist keine Tautologie“.)

Übungsaufgabe 41. Sei $M = \{\varphi_n : n \in \mathbb{N}\}$ eine Menge aussagenlogischer Formeln, so dass für alle natürlichen Zahlen n gilt:

$$\models \varphi_{n+1} \wedge \varphi_{n+2} \to \varphi_n \quad \text{und} \quad \not\models \varphi_n \to (\neg\varphi_{n+2} \to \neg\varphi_{n+1})$$

Zeigen Sie mit Hilfe des Endlichkeitssatzes der Aussagenlogik: M hat ein Modell. Ferner geben Sie ein Beispiel für eine solche Formelmenge M an.

Ableitbarkeit und Vollständigkeit

Im weiteren Verlauf der Vorlesung wollen wir neben der *semantischen* Folgerungsbeziehung $\models$ verschiedene *syntaktische*Ableitbarkeitsbeziehungen $\vdash$ (Beweissysteme) untersuchen. Es gilt dann meist ein *Vollständigkeitssatz:*

Folgern
Ableiten

Satz 42. Für $\Phi \subseteq \mathsf{Form}_{\mathrm{AL}}$ und $\varphi \in \mathsf{Form}_{\mathrm{AL}}$ gilt $\Phi \models \varphi$ genau dann, wenn $\Phi \vdash \varphi$.

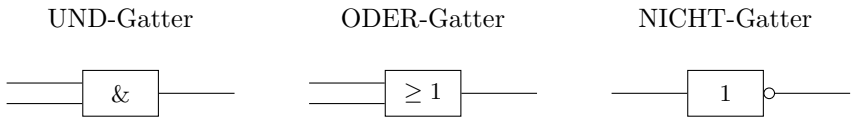
Vollständigkeitssatz

1.6. Anwendungen

1.6.1. Formeln und logische Gatter

Logische Schaltkreise und Gatter. Schaltkreis Bekanntlich werden die logischen Grundfunktionen *und*, *oder* und *nicht* als Gatter realisiert.

Symbolisch werden sie üblicherweise wie folgt dargestellt:



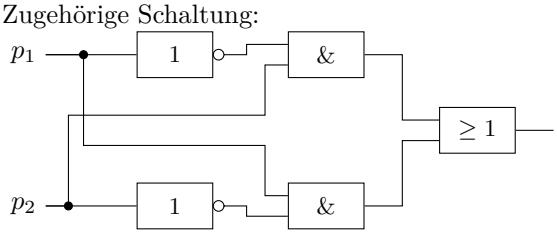
Logische Gatter werden beim Schaltungsentwurf zu komplizierteren Schaltungen kombiniert:

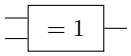
Beispiel 43. Exklusives ODER

Wahrheitstafel:

$\oplus$	0	1
0	0	1
1	1	0

Zugehörige Schaltung:



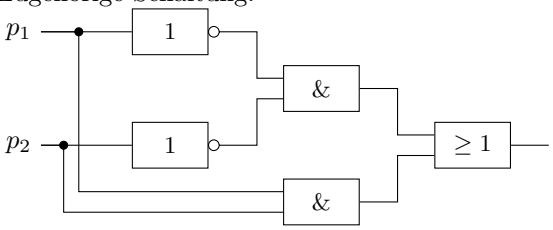
Das exklusive ODER wird auch kurz wie folgt dargestellt: 

Beispiel 44. Bi-Implikation (Äquivalenz)

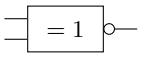
Wahrheitstafel:

$\leftrightarrow$	0	1
0	1	0
1	0	1

Zugehörige Schaltung:



Zeichen für die Bi-Implikation (Äquivalenz):



Übungsaufgabe 45. Geben Sie Schaltungen für NAND und NOR an, die durch folgende Wahrheitstafeln definiert sind:

NAND

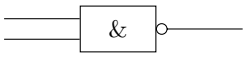
$\uparrow$	0	1
0	1	1
1	1	0

NOR

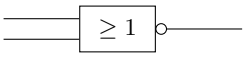
$\downarrow$	0	1
0	1	0
1	0	0

Diese Schaltungen werden üblicherweise wie folgt abgekürzt:

NAND-Gatter



NOR-Gatter



1.6.2. Entscheidungsdiagramme

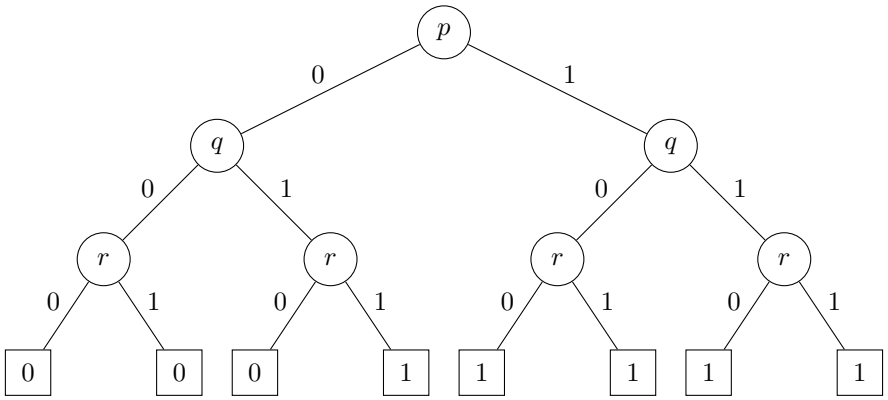
Binäre Entscheidungsdiagramme, BDD (Binary Decision Diagrams). Eine weitere Möglichkeit aussagenlogische Formeln grafisch darzustellen sind binäre Entscheidungsdiagramme (im Folgenden: BDDs).

Definition 46. Ein BDD D zu einer aussagenlogischen Formel φ ist ein gerichteter azyklischer Graph. Jedes Blatt wird mit einem Wahrheitswert 0 oder 1 markiert. Jeder innere Knoten wird mit einer aussagenlogischen Variablen markiert und hat zwei von ihm ausgehende Kanten: Eine, die sogenannte falsche Kante, wird mit einer Null markiert, die andere, sogenannte wahre Kante, wird mit einer 1 markiert.

Sei $\mathfrak{I}$ eine Belegung. Dazu gehört ein eindeutiger Pfad in D, nämlich der Pfad der wahren bzw. falschen Kanten gemäß dem Wort der Variablen in $\mathfrak{I}$. Die Ausgabe von D ist das Wort des Blattes auf diesem Pfad.

D stellt die aussagenlogische Formel φ dar, falls D für jede Belegung $\mathfrak{I}$ den Wert $\mathfrak{I}(\varphi)$ ausgibt.

Beispiel 47. Ein BDD, der die Formel $\varphi := p \vee (q \wedge r)$ darstellt:



1.6.3. Minimierung von DNF

Minimierung einer DNF mit Hilfe des Quine-McCluskey-Verfahrens: Gegeben sei

$$\varphi := (p_1 \wedge p_2 \wedge p_3) \vee (p_1 \wedge p_2 \wedge \neg p_3) \vee (\neg p_1 \wedge p_2 \wedge \neg p_3) \vee (\neg p_1 \wedge \neg p_2 \wedge p_3) \vee (\neg p_1 \wedge \neg p_2 \wedge \neg p_3).$$

Willard Van Orman Quine (1908–2000) amerikanischer Philosoph und Logiker

Edward J. McCluskey (*1929) amerikanischer Informatiker

1. Schritt Wir verwenden die Äquivalenzen

(i) $(p \wedge q) \vee (p \wedge \neg q) \equiv p$,

(ii) $p \wedge q \equiv q \wedge p$ sowie

(iii) $p \vee p \equiv p$,

um möglichst viele Konjunktionen zusammenzufassen.

Dabei gehen wir von der ersten geordneten Liste der Konjunktionen aus (die Konjunktionen aus der DNF sind nach der Anzahl der auftretenden Negationszeichen geordnet):

Nummer	Konjunktion	Block
1	$p_1 \wedge p_2 \wedge p_3$	1
2	$p_1 \wedge p_2 \wedge \neg p_3$	2
3	$\neg p_1 \wedge p_2 \wedge \neg p_3$	3
4	$\neg p_1 \wedge \neg p_2 \wedge p_3$	
5	$\neg p_1 \wedge \neg p_2 \wedge \neg p_3$	4

Die fünf Konjunktionsglieder unserer DNF haben wir in vier Blöcke aufgeteilt. Jetzt sind Zusammenfassungen möglich (mit Hilfe der Regel (i)), allerdings nur bei Konjunktionen, die zu benachbarten Blöcken gehören.

Damit erhalten wir folgende 1. Zusammenfassung:

Zusammengefasste Konjunktionen	Ergebnis
1., 2.	$p_1 \wedge p_2$
2., 3.	$p_2 \wedge \neg p_3$
3., 4.	$\neg p_1 \wedge \neg p_3$
4., 5.	$\neg p_1 \wedge p_2$

Wir sortieren die erhaltenen Konjunktionen nach gleichen Variablen und anschließend nach der Anzahl der auftretenden Negationszeichen. Damit erhalten wir die zweite geordnete Liste der Konjunktionen:

Nummer	Konjunktion
1	$p_1 \wedge p_2$
2	$\neg p_1 \wedge \neg p_2$
3	$\neg p_1 \wedge \neg p_3$
4	$p_2 \wedge \neg p_3$

2. Schritt Aus der im 1. Schritt ermittelten Darstellung für φ lassen wir alle überflüssigen Konjunktionen weg. Hier ist die Konjunktion $p_2 \wedge \neg p_3$ überflüssig, wie wir anhand der Tabelle sehen.

p_1	p_2	p_3	$p_1 \wedge p_2$	$\neg p_1 \wedge p_2$	$\neg p_1 \wedge \neg p_3$	$p_2 \wedge \neg p_4$	φ
0	0	0	0	1	1	0	1
0	0	1	0	1	0	0	1
0	1	0	0	0	1	1	1
0	1	1	0	0	0	0	0
1	0	0	0	0	0	0	0
1	0	1	0	0	0	0	0
1	1	0	1	0	0	1	1
1	1	1	1	0	0	0	1

Letztlich erhalten wir folgende minimale DNF:

$$\varphi \equiv (p_1 \wedge p_2) \vee (\neg p_1 \wedge \neg p_2) \vee (\neg p_1 \wedge \neg p_3)$$

Übungsaufgabe 48. Gegeben sei folgende Wahrheitstafel:

p_1	p_2	p_3	p_4	$\varphi(p_1, p_2, p_3, p_4)$	$\psi(p_1, p_2, p_3, p_4)$
0	0	0	0	0	1
0	0	0	1	0	1
0	0	1	0	0	1
0	0	1	1	1	1
0	1	0	0	0	1
0	1	0	1	1	1
0	1	1	0	1	1
0	1	1	1	1	0
1	0	0	0	0	1
1	0	0	1	1	1
1	0	1	0	1	1
1	0	1	1	1	0
1	1	0	0	1	1
1	1	0	1	1	0
1	1	1	0	1	0
1	1	1	1	1	0

Stellen Sie zunächst jeweils eine DNF zu φ und zu ψ auf und minimieren Sie diese anschließend.

Kapitel 2

Hornformeln

Die nun folgenden Hornformeln sind nach dem amerikanischen Logiker Alfred Horn benannt.

Alfred Horn (1918–2001)
amerikanischer Logiker

Definition 49. Eine *Hornformel* ist eine Formel in konjunktiver Normalform, die höchstens ein positives Literal pro Klausel (sog. Hornklausel) enthält.

Hornformel

Ein Beispiel für eine Hornformel ist

$$\underbrace{(\neg p_1 \vee \neg p_2 \vee p_3)} \wedge \underbrace{(p_1 \vee \neg p_2)} \wedge \underbrace{(\neg p_2 \vee \neg p_4)} \wedge \underbrace{p_2}$$

Die in den Klammern gekennzeichneten Konjunktionen sind die *Hornklauseln*. Jede Hornklausel ist äquivalent zu einer Implikation der Form

- $p_1 \wedge \cdots \wedge p_k \to q$,
- $p_1 \wedge \cdots \wedge p_k \to 0$ oder
- $1 \to p$

Für die Hornformel

$$(\neg p_1 \vee \neg p_2 \vee p_3) \wedge (p_1 \vee \neg p_2) \wedge (\neg p_2 \vee \neg p_4) \wedge p_2$$

ergibt sich

Klausel	äquivalente Implikation
$\neg p_1 \vee \neg p_2 \vee p_3$	$p_1 \wedge p_2 \to p_3$
$p_1 \vee \neg p_2$	$p_2 \to p_1$
$\neg p_2 \vee \neg p_4$	$p_2 \wedge p_4 \to 0$
p_2	$1 \to p_2$

Übungsaufgabe 50. Geben Sie eine zu φ äquivalente Hornformel φ_{Horn} an:

$$\varphi := q_1 \wedge (q_1 \wedge q_3 \to \neg q_2) \wedge \big((q_5 \wedge \neg q_4) \vee (q_3 \wedge q_5)\big) \wedge \neg q_2 \wedge (q_1 \wedge \neg q_4 \to \neg q_5)$$

Stellen Sie φ_{Horn} auch als Konjunktion von Implikationen dar.

Das Erfüllbarkeitsproblem für Hornformeln wird mit HORNSAT bezeichnet.

HORNSAT

Problem:	HORNSAT
Eingabe:	Eine Hornformel Formel φ
Frage:	Ist φ erfüllbar?

Im Gegensatz zu SAT ist für HORNSAT ein effizienter Algorithmus bekannt. Es gilt nämlich:

Satz 51. Für erfüllbare Hornformeln lässt sich eine erfüllende Belegung in Polynomialzeit konstruieren. HORNSAT ist also effizient lösbar.

Markierungsalgorithmus

Beweis Die Aussage ergibt sich aus Algorithmus 3, dem sogenannten *Markierungsalgorithmus für Hornformeln*:

Eingabe: Hornformel φ

1: $\mathfrak{I}(p) := \begin{cases} 1 & \text{falls } 1 \to p \text{ in } \varphi \text{ vorkommt} \\ 0 & \text{sonst} \end{cases}$
2: **while** φ enthält eine Teilformel $p_1 \wedge \cdots \wedge p_k \to q$
mit $\mathfrak{I}(p_1) = \cdots = \mathfrak{I}(p_k) = 1$ und $\mathfrak{I}(q) = 0$ **do**
3: $\mathfrak{I}(q) := 1$
4: **end while**
5: **if** $\mathfrak{I}$ erfüllt φ **then**
6: **return** Erfüllbar
7: **else**
8: **return** Unerfüllbar
9: **end if**

Algorithmus 3: Der Horn-Algorithmus

Zur Korrektheit des Algorithmus müssen wir zeigen, dass er für jede Eingabe terminiert und die Ausgaben in Zeilen 6 und 8 korrekt sind. Erstere Behauptung ergibt sich aus der Laufzeitanalyse (siehe unten).

Die Korrektheit der Ausgabe „Erfüllbar“ in Zeile 6 ist klar, denn in diesem Fall haben wir mit $\mathfrak{I}$ sogar eine erfüllende Belegung gefunden. Um die Korrektheit der Ausgabe „Unerfüllbar“ in Zeile 8 zu zeigen, nehmen wir an, dass diese Ausgabe falsch ist und $\mathfrak{I}'$ eine erfüllende Belegung für φ ist. Zunächst beobachten wir, dass die in $\mathfrak{I}$ auf 1 gesetzten Variablen auch in $\mathfrak{I}'$ auf 1 gesetzt sein müssen. Dies ist unmittelbar klar für die Variablen p, die in Zeile 1 den Wert 1 erhalten, da ja die Klausel $\{p\}$ in φ vorkommt. Ebenso überträgt sich dies auf die Variablen, die in Zeile 3 auf 1 gesetzt werden. Wird die while-Schleife verlassen, erfüllt $\mathfrak{I}$ also alle Klauseln aus φ der Bauart p und $p_1 \wedge \cdots \wedge p_k \to q$. Gilt trotzdem $\mathfrak{I} \not\models \varphi$, so kann das nur daran liegen, dass φ eine Klausel $p_1 \wedge \cdots \wedge p_k \to 0$ enthält und außerdem $\mathfrak{I}(p_1) = \cdots = \mathfrak{I}(p_k) = 1$ ist. In diesem Fall gilt aber auch, wie wir uns gerade klar gemacht haben, $\mathfrak{I}'(p_1) = \cdots = \mathfrak{I}'(p_k) = 1$, und mithin erfüllt im Widerspruch zur Annahme auch $\mathfrak{I}'$ nicht φ.

Zur Laufzeitanalyse des Algorithmus sei φ eine Formel der Größe n. Dann enthält φ höchstens n Variablen. In jedem Durchlauf der while-Schleife wird eine Variable von

0 auf 1 gesetzt. Mithin wird die while-Schleife maximal n mal durchlaufen. Da jeder Durchlauf der while-Schleife in linearer Zeit möglich ist, ergibt sich insgesamt eine Laufzeit von $O(n^2)$. ■

Übungsaufgabe 52. Untersuchen Sie mit dem Markierungsalgorithmus für Hornformeln, ob die folgende Formel φ erfüllbar ist und geben Sie jeden Schritt des Algorithmus an. Wie viele Zeilen hat die Wahrheitstafel dieser Formel?

$$\begin{aligned}\varphi \quad = \quad & (1 \to x_1) \wedge (1 \to x_2) \wedge (x_1 \wedge x_2 \wedge x_3 \wedge x_4 \wedge x_5 \to 0) \\ & \wedge (x_1 \wedge x_2 \wedge x_3 \to x_4) \wedge (x_4 \to x_5) \wedge (x_2 \to x_3)\end{aligned}$$

minimale erfüllende Belegung

Wir bemerken noch, dass in obigem Erfüllbarkeitstest für Hornformeln eine minimale erfüllende Belegung $\mathfrak{I}$ für φ konstruiert wird (falls φ erfüllbar ist), d.h. gilt $\mathfrak{I}' \vDash \varphi$, so folgt aus $\mathfrak{I}(p) = 1$ auch $\mathfrak{I}'(p) = 1$ für alle $p \in \mathsf{Var}$. $\mathfrak{I}$ setzt also nur so wenige Variablen wie notwendig auf 1. Ebenso folgt aus dem Beweis von Satz 51, dass Hornformeln, die keine Klauseln der Form $p_1 \wedge \cdots \wedge p_k \to 0$ enthalten, stets erfüllbar sind.

Kapitel 3

Resolution

Martin Davis (*1928) amerikanischer Logiker

Hilary Putnam (*1926) amerikanischer Philosoph und Logiker

John A. Robinson (*1928) amerikanischer Philosoph und Informatiker

3.1. Resolutionsbeweise

Das Resolutionssystem geht zurück auf Davis und Putnam (1960) und Robinson (1965). Resolutionsbeweise operieren mit Klauseln und sind Widerlegungsbeweise.

Definition 53. Seien C und D Klauseln mit $p \in C$ und $\neg p \in D$. Dann liefert die *Resolutionsregel* angewendet auf C und D die Klausel $R = (C \setminus \{p\}) \cup (D \setminus \{\neg p\})$.

$\frac{C \quad D}{R}$

$$\text{Schreibweise:} \quad \frac{C \qquad D}{(C \setminus \{p\}) \cup (D \setminus \{\neg p\})}$$

Die Klausel R heißt *Resolvente* von C und D.

Resolvente

Sei Γ eine Menge von Klauseln. Eine *Resolutionsableitung* einer Klausel C aus Γ ist eine Folge

Resolutionsableitung

$$C_1, \ldots, C_k = C$$

von Klauseln, so dass für alle $i = 1, \ldots, k$ gilt

(i) $C_i \in \Gamma$ oder

(ii) es existieren $1 \leq j_1 \leq j_2 < i$ mit

$$\frac{C_{j_1} \quad C_{j_2}}{C_i}.$$

Eine *Resolutionswiderlegung* von Γ ist eine Resolutionsableitung von $\square$ aus Γ. Schreibweise: $\Gamma \vdash_{\mathsf{Res}} C$ bzw. $\Gamma \vdash_{\mathsf{Res}} \square$.

Resolutionswiderlegung

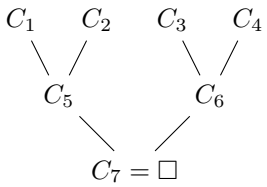
$\vdash_{\mathsf{Res}}$

Beispiel 54.

$$\frac{C = \{q, p\} \quad D = \{\neg p\}}{R = \{q\}}$$

Bemerkung. Neben $\frac{C \quad D}{R}$ ist auch die Schreibweise $\begin{smallmatrix} C & & D \\ & \diagdown \; \diagup & \\ & R & \end{smallmatrix}$ üblich.

Beispiel 55. Sei $\Gamma = \{C_1, C_2, C_3, C_4\}$ mit $C_1 = \{p, q\}$, $C_2 = \{\neg p, q\}$, $C_3 = \{p, \neg 1\}$ und $C_4 = \{\neg p.\neg q\}$. Die Resolutionsableitung



kann durch $(C_1, C_2, C_5, C_3, C_4, C_6, C_7)$ bzw. $(C_3, C_4, C_6, C_1, C_2, C_5, C_7)$ als Folge dargestellt werden.

Übungsaufgabe 56. Gegeben sei die Formel

$$\theta := (\neg p \vee \neg q \vee r) \wedge p \wedge \neg r \wedge (\neg p \vee q \vee r).$$

Untersuchen Sie, ob aus θ die leere Klausel $\square$ abgeleitet werden kann.

$\mathsf{res}(\Gamma)$ **Definition 57.** Für eine Menge Γ von Klauseln sei

$$\mathsf{res}(\Gamma) := \Gamma \cup \{R \mid R \text{ ist Resolvente zweier Klauseln aus } \Gamma\}$$

$\mathsf{res}^*(\Gamma)$ und

$$\begin{aligned} \mathsf{res}^0(\Gamma) &:= \Gamma \\ \mathsf{res}^{n+1}(\Gamma) &:= \mathsf{res}\big(\mathsf{res}^n(\Gamma)\big) \quad \text{für } n \geq 0 \\ \mathsf{res}^*(\Gamma) &:= \bigcup_{i \geq 0} \mathsf{res}^i(\Gamma) \end{aligned}$$

Beispiel 58. Gegeben sei die Klauselmenge $C = \{C_1, C_2, C_3\}$ mit

$$C_1 = \{p_1, p_2, \neg p_3\}, \quad C_2 = \{p_1, \neg p_2, p_3\}, \quad C_3 = \{\neg p_1, p_2, \neg p_3\}$$

Dann erhalten wir die folgenden Resolventen:

Klauseln aus C	Resolventen
C_1 und C_2	$R_1 = \{p_1, p_3, \neg p_3\}, R_2 = \{p_1, p_2, \neg p_3\}$
C_1 und C_3	$R_3 = \{p_2, \neg p_3\}$
C_2 und C_3	$R_4 = \{p_2, \neg p_2, p_3, \neg p_3\}, R_5 = \{p_1, \neg p_1, p_3, \neg p_3\}$, $R_6 = \{p_1, \neg p_1, p_2, \neg p_2\}$
C_i und C_i, $1 \leq i \leq 3$	C_i

Also:

$$\begin{aligned} \mathsf{res}(C) = \Big\{ & \{p_1, p_2, \neg p_3\}, \{p_1, \neg p_2, p_3\}, \{\neg p_1, p_2, \neg p_3\}, \\ & \{p_1, p_3, \neg p_3\}, \{p_1, p_2, \neg p_3\}, \\ & \{p_2, \neg p_3\}, \\ & \{p_2, \neg p_2, p_3, \neg p_3\}, \{p_1, \neg p_1, p_3, \neg p_3\}, \{p_1, \neg p_1, p_2, \neg p_2\} \Big\} \end{aligned}$$

Beispiel 59. Wir betrachten $C' = \{C_1, C_2\} \subset C$.

Nach unserer Definition ergeben sich

$$\mathsf{res}^0(C') = C' = \{C_1, C_2\}$$
$$\mathsf{res}^1(C') = \big\{C_1, C_2, \{p_1, p_3, \neg p_3\}, \{p_1, p_2, \neg p_3\}\big\}$$

Mit $R_1 := \{p_1, p_3, \neg p_3\}$ und $R_2 := \{p_1, p_2, \neg p_3\}$ ergibt sich folgende Tabelle:

Klauseln	Resolventen
C_1 und C_2	R_1, R_2
C_1 und R_1	C_1
C_1 und R_2	C_1
C_2 und R_1	C_2
C_2 und R_2	C_2
R_1 und R_2	keine Resolvente

Da keine neue Klausel erzeugt wurde, gilt:

$$\mathsf{res}^2(C') = \mathsf{res}^1(C') = \mathsf{res}(C')$$

Insgesamt gilt:

$$\mathsf{res}^*(C') = \bigcup_{i \geq 0} \mathsf{res}^i(C') = \{C_1, C_2, R_1, R_2\}.$$

Satz 60. Sei Γ eine endliche Menge von Klauseln. Dann gibt es $n \in \mathbb{N}$, so dass $\mathsf{res}^{n+1}(\Gamma) = \mathsf{res}^n(\Gamma)$ und $\mathsf{res}^*(\Gamma) = \mathsf{res}^n(\Gamma)$. Endlichkeit der Resolution

Beispiel für eine Resolutionswiderlegung

Sei $\Gamma = \{\{p, q\}, \{p, \neg q\}, \{\neg p, q\}, \{\neg p, \neg q\}\}$. Eine Resolutionswiderlegung von Γ ist:

$$\dfrac{\dfrac{\{p,q\} \quad \{\neg p, q\}}{\{q\}} \qquad \dfrac{\{\neg p, \neg q\} \quad \{p, \neg q\}}{\{\neg q\}}}{\square}$$

Der Resolutionskalkül ist nicht vollständig, z.B. $p \vDash p \vee q$, aber $p \nvdash_{\mathsf{Res}} p \vee q$. Er ist jedoch *widerlegungsvollständig*. Es gilt: widerlegungsvollständig

Satz 61. Sei Γ eine endliche Klauselmenge. Dann ist Γ genau dann unerfüllbar, wenn es eine Resolutionswiderlegung von Γ gibt. (Symbolisch: $\Gamma \vDash \square \iff \Gamma \vdash_{\mathsf{Res}} \square$.) Vollständigkeitssatz für Resolution

Beweis

„$\Longleftarrow$“ (*Korrektheit*) Wir zeigen per Induktion über die Beweislänge k die folgende

Behauptung: Wenn $C \in \mathsf{res}^k(\Gamma)$ in k Schritten und $\mathfrak{I} \models \Gamma$, so $\mathfrak{I} \models C$.

Hieraus folgt die Korrektheit. Ist nämlich $\Gamma \vdash_{\mathsf{Res}} \square$, so folgt aus der Unerfüllbarkeit von $\square$ auch die Unerfüllbarkeit von Γ.

Zum Beweis der Behauptung sei zunächst als Induktionsanfang $k = 0$. Dann ist C eine Klausel aus Γ, und die Behauptung gilt offenbar.

Für den Induktionsschritt sei nun

$$\frac{C \cup \{p\} \quad D \cup \{\neg p\}}{C \cup D}$$

ein Resolutionsschritt und gelte $\mathfrak{I} \models C \cup \{p\}$ sowie $\mathfrak{I} \models D \cup \{\neg p\}$ für eine Belegung $\mathfrak{I}$. Sei etwa $\mathfrak{I}(p) = 1$. Wegen $\mathfrak{I} \models D \cup \{\neg p\}$ gilt dann auch $\mathfrak{I} \models D$. Mithin erfüllt $\mathfrak{I}$ auch $C \cup D$. Für $\mathfrak{I}(p) = 0$ ist die Argumentation analog.

„$\Longrightarrow$“ (*Vollständigkeit*) Sei Γ unerfüllbar. Per Induktion über die Anzahl n der Variablen in Γ zeigen wir $\Gamma \vdash_{\mathsf{Res}} \square$.

Für $n = 0$ ist $\Gamma = \{\square\}$, und die Behauptung gilt. Sei nun $n > 0$ und seien $p_1, \ldots, p_n$ die Variablen in Γ. Wir zerlegen Γ in die Mengen

$$\Gamma = \Gamma_{00} \cup \Gamma_{01} \cup \Gamma_{10} \cup \Gamma_{11}$$

mit

$$\begin{aligned}
\Gamma_{00} &= \{C \in \Gamma \mid p_n \notin C \text{ und } \neg p_n \notin C\} \\
\Gamma_{01} &= \{C \in \Gamma \mid p_n \notin C \text{ und } \neg p_n \in C\} \\
\Gamma_{10} &= \{C \in \Gamma \mid p_n \in C \text{ und } \neg p_n \notin C\} \\
\Gamma_{11} &= \{C \in \Gamma \mid p_n \in C \text{ und } \neg p_n \in C\}.
\end{aligned}$$

Per Resolution aller Klauselpaare aus Γ_{01} und Γ_{10} über die Variable p_n bilden wir die Klauselmenge

$$\Delta = \{C \cup D \mid C \cup \{p_n\} \in \Gamma_{10} \text{ und } D \cup \{\neg p_n\} \in \Gamma_{01}\}.$$

Weiter setzen wir

$$\Gamma' = \Gamma_{00} \cup \Delta.$$

Dann enthält Γ' nur die Variablen $p_1, \ldots, p_{n-1}$, und es gilt:

Behauptung: Γ' ist unerfüllbar.

Beweis der Behauptung: Angenommen, es gibt eine Γ' erfüllende Belegung $\mathfrak{I}_1$. Es seien

$$\begin{aligned}
\Delta_1 &= \{C \mid C \cup \{p_n\} \in \Gamma_{10}\} \quad \text{und} \\
\Delta_2 &= \{D \mid D \cup \{\neg p_n\} \in \Gamma_{01}\}.
\end{aligned}$$

Dann gilt $\mathfrak{I}_1 \vDash \Delta_1$ oder $\mathfrak{I}_1 \vDash \Delta_2$. Falls nämlich $\mathfrak{I}_1 \nvDash C$ mit $C \in \Delta_1$ und $\mathfrak{I}_1 \nvDash D$ mit $D \in \Delta_2$, so gälte auch $\mathfrak{I}_1 \nvDash C \cup D$. Es ist aber $C \cup D \in \Gamma'$ und mithin $\mathfrak{I}_1 \vDash C \cup D$ wegen $\mathfrak{I}_1 \vDash \Gamma'$.

Sei also o.B.d.A. $\mathfrak{I}_1 \vDash \Delta_1$. Wir setzen

$$\mathfrak{I}_2(p_i) = \begin{cases} \mathfrak{I}_1(p_i) & i \neq n \\ 0 & i = n. \end{cases}$$

Dann gilt

(i) $\mathfrak{I}_2 \vDash \Gamma_{00}$ wegen $\Gamma_{00} \subseteq \Gamma'$,

(ii) $\mathfrak{I}_2 \vDash \Gamma_{10}$ wegen $\mathfrak{I}_1 \vDash \Delta_1$,

(iii) $\mathfrak{I}_2 \vDash \Gamma_{01}$ wegen $\mathfrak{I}_2(p_n) = 0$ und

(iv) $\mathfrak{I}_2 \vDash \Gamma_{11}$ weil $\vDash \Gamma_{11}$.

Mithin gilt $\mathfrak{I}_2 \vDash \Gamma$ im Widerspruch zur Voraussetzung. Damit ist die Behauptung gezeigt.

Da Γ' also unerfüllbar ist, gibt es nach Induktionsvoraussetzung eine Resolutionswiderlegung von Γ' und damit auch von Γ. ■

Resolution ist ein Widerlegungskalkül, d.h. wir zeigen $\varphi \in \mathrm{TAUT}$ mittels

$$\neg\varphi \vdash_{\mathsf{Res}} \square,$$

(siehe Satz 61).

Problem: $\neg\varphi$ muss in KNF vorliegen.

Dies ist natürlich gegeben für Formeln φ, die in DNF vorliegen. In diesem Fall nämlich erhalten wir nach den de Morganschen Regeln direkt eine KNF-Formel für $\neg\varphi$. Allgemein bieten sich folgende Möglichkeiten an:

1. Möglichkeit: $\neg\varphi$ wird in eine KNF-Formel Γ umgeformt.

Der Nachteil hierbei ist, dass die Größe von Γ exponentiell in der Größe von φ sein kann (siehe Korollar 29).

2. Möglichkeit: Wir formen $\neg\varphi$ in eine erfüllbarkeitsäquivalente Klauselmenge $\Gamma_{\neg\varphi}$ um.

Ganz allgemein wird hierzu eine beliebige Formel χ (in unserem Falle $\neg\varphi$) in eine erfüllbarkeitsäquivalente Klauselmenge Γ_χ umgeformt. Dazu führen wir für jede Teilformel ψ von χ eine neue Erweiterungsvariable q_ψ ein. Weiter definieren wir für eine beliebige Formel ψ die Klauselmenge Δ_ψ wie folgt:

$$\Delta_\psi = \begin{cases} \{\{q_\psi, \neg p\}, \{\neg q_\psi, p\}\} & \text{falls } \psi = p \\ \{\{q_\psi, q_\theta\}, \{\neg q_\psi, \neg q_\theta\}\} & \text{falls } \psi = \neg\theta \\ \{\{\neg q_\psi, q_{\theta_1}, q_{\theta_2}\}, \{q_\psi, \neg q_{\theta_1}\}, \{q_\psi, \neg q_{\theta_2}\}\} & \text{falls } \psi = \theta_1 \vee \theta_2 \\ \{\{q_\psi, \neg q_{\theta_1}, \neg q_{\theta_2}\}, \{\neg q_\psi, q_{\theta_1}\}, \{\neg q_\psi, q_{\theta_2}\}\} & \text{falls } \psi = \theta_1 \wedge \theta_2 \end{cases}$$

Nun setzen wir

$$\Gamma_\chi = \bigcup\{\Delta_\psi \mid \psi \text{ ist eine Teilformel von } \chi\} \cup \{\{q_\chi\}\}.$$

Es gilt dann: χ ist erfüllbar genau dann, wenn Γ_χ erfüllbar ist (χ und Γ_χ sind erfüllbarkeitsäquivalent).

Vorteil: Die Größe von Γ_χ ist linear in der Größe von χ.

Nachteil: χ und Γ_χ sind nicht logisch äquivalent, sondern nur erfüllbarkeitsäquivalent.

Wir betrachten ein Beispiel für diese Konstruktion:

- Sei $\chi = (x \wedge y) \vee \neg z$.
- Teilformeln: $(x \wedge y) \vee \neg z$, $x \wedge y$, $\neg z$, z, x, y
- Variablen: $q_{(x\wedge y)\vee z}$, $q_{x\wedge y}$, $q_{\neg z}$, q_z, q_x, q_y

Teilformel ψ	Δ_ψ
$(x \wedge y) \vee \neg z$	$\{\neg q_{(x\wedge y)\vee\neg z}, q_{x\wedge y}, q_{\neg z}\}$, $\{q_{(x\wedge y)\vee\neg z}, \neg q_{x\wedge y}\}$, $\{q_{(x\wedge y)\vee\neg z}, \neg q_{\neg z}\}$
$x \wedge y$	$\{q_{x\wedge y}, \neg q_x, \neg q_y\}$, $\{\neg q_{x\wedge y}, q_x\}$, $\{\neg q_{x\wedge y}, q_y\}$
$\neg z$	$\{q_{\neg z}, q_z\}$, $\{\neg q_{\neg z}, \neg q_z\}$
z	$\{q_z, \neg z\}$, $\{\neg q_z, z\}$
x	$\{q_x, \neg x\}$, $\{\neg q_x, x\}$
y	$\{q_y, \neg y\}$, $\{\neg q_y, y\}$

Die Menge Γ_χ ergibt sich dann aus allen obigen Klauseln zusammen mit $\{q_{(x\wedge y)\vee\neg z}\}$.

Übungsaufgabe 62. Sei $\varphi = p \vee \neg p$. Geben Sie $\Gamma_{\neg\varphi}$ an und geben Sie eine Resolutionswiderlegung von $\Gamma_{\neg\varphi}$ an.

Auch der Resolutionskalkül liefert keinen effizienten Algorithmus für SAT oder TAUT, wie der folgende Satz zeigt:

Satz von Haken

Satz 63. Es gibt eine Folge Γ_n von Klauselmengen mit folgenden Eigenschaften:

(i) Γ_n ist eine unerfüllbare Klauselmenge in n Variablen.

(ii) Die Anzahl der Klauseln in Γ_n ist polynomiell in n.

(iii) Die minimale Resolutionswiderlegung von Γ_n enthält exponentiell viele Klauseln.

3.2. Erfüllbarkeitstests

Aus dem Beweis des vorigen Satzes ergibt sich sofort einErfüllbarkeitstest für aussagenlogische Formeln, der auch als *Davis-Putnam-Algorithmus* (oder kurz DP-Algorithmus) bekannt ist, sh. Algorithmus 4.

Davis-Putnam- Algorithmus

Eingabe: KNF-Formel φ
Ausgabe: Boole'scher Wert, der die Erfüllbarkeit von φ angibt

```
1:  DP-RESOLUTION(KNF-Formel φ)
2:      if es gibt keine Variable, die sowohl positiv als auch negativ vorkommt then
3:          return true
4:      wähle eine Variable p aus φ, die sowohl positiv als auch negativ vorkommt
5:      Res^1(φ,p) := {C_1 ∪ C_2 | C_1 ∪ {p}, C_2 ∪ {¬p} sind Klauseln in φ}
6:      if □ ∈ Res^1(φ,p) then return false
7:      φ := φ ∪ Res^1(φ,p)
    /* entferne aus φ alle Klauseln mit der Variable p */
8:      φ_{p,¬p} := {C ∈ φ | p ∈ C oder ¬p ∈ C}
9:      φ := φ \ φ_{p,¬p}
10:     return DP-RESOLUTION(φ)
```

Algorithmus 4: Der Davis-Putnam-Algorithmus

Übungsaufgabe 64. Gegeben sei die Klauselmenge

$$\Gamma := \{p_1, p_2, \neg p_3\}, \{\neg p_1, p_4\}, \{\neg p_2, p_3\}, \{p_1, p_4\}, \{p_3, \neg p_4\}, \{\neg p_3\}\}.$$

(i) Geben Sie eine Resolutionswiderlegung von Γ an. Gehen Sie dabei nach dem Davis-Putnam-Algorithmus vor und verwenden Sie als Heuristik für die Auswahl der Variablen die Ordnung $p_1 < p_2 < p_3 < p_4$.

(ii) Eine Klauselmenge Δ heißt *minimal unerfüllbar*, falls Δ unerfüllbar ist und jede echte Teilmenge $\Delta' \subset \Delta$ erfüllbar ist. Ist obige Klauselmenge Γ minimal unerfüllbar? Begründen Sie Ihre Antwort.

Übungsaufgabe 65. Sei

$$\varphi = (\neg u \vee w \vee y) \wedge w \wedge (\neg w \vee x) \wedge (z \vee \neg y) \wedge (\neg z \vee w) \wedge (u \vee w) \wedge \neg x \wedge (v \vee u \vee y) \wedge \neg y$$

und h die Heuristik, die aus einer Klauselmenge Γ die aussagenlogische Variable p mit dem kleinsten Wert $h(p) = |\{C \in \Gamma \mid p \in C\}| \cdot |\{C \in \Gamma \mid \neg p \in C\}| > 0$ auswählt. Haben zwei Variablen denselben Wert, so soll die lexikalisch kleinere ausgewählt werden. Überprüfen Sie mit dem Davis-Putnam-Algorithmus und der Heuristik h, ob φ erfüllbar ist.

Übungsaufgabe 66. Gegeben sind die folgende Klauselmenge Γ

$$\{\{a, b, \neg c\}, \{\neg a, d\}, \{\neg b, c\}, \{a, d\}, \{c, \neg d\}, \{\neg c\}\}$$

und die Heuristik h, die aus einer Klauselmenge Γ die aussagenlogische Variable p mit dem kleinsten Wert $h(p) = |\{C \in \Gamma \mid p \in C\}| \cdot |\{C \in \Gamma \mid \neg p \in C\}| > 0$ auswählt. Haben zwei Variablen den gleichen Wert, so soll die lexikalisch kleinere ausgewählt werden.

(i) Geben Sie eine Resolutionswiderlegung von Γ an. Gehen Sie dabei nach dem Davis-Putnam-Algorithmus vor und verwenden Sie für die Auswahl der Variablen die Heuristik h.

(ii) Eine Klauselmenge Δ heißt *minimal unerfüllbar*, falls jede echte Teilmenge $\Delta' \subset \Delta$ erfüllbar ist. Ist obige Klauselmenge Γ minimal unerfüllbar? Begründen Sie Ihre Antwort.

DPLL-Algorithmus

Der Davis-Putnam-Algorithmus wurde von Davis, Longeman und Loveland aufgeriffen und zu einem einfachen rekursiven Algorithmus, bekannt als *DPLL-Algorithmus*, modifiziert.

Wir brauchen zunächst etwas Notation:

Definition 67. Sei φ eine Formel und $\mathfrak{I}$ eine partielle Belegung. Dann ist $\varphi|_{\mathfrak{I}}$ die vereinfachte Formel, die aus φ durch Ersetzen aller Variablen x im Wertebereich von $\mathfrak{I}$ durch $\mathfrak{I}(x)$ entsteht.

Der DPLL-Algorithmus basiert auf folgender Idee: Ist eine Formel φ als Klauselmenge gegeben, so prüfen wir zunächst, ob φ trivial erfüllbar ist (φ ist die leere Klauselmenge) oder trivial unerfüllbar ist (φ enthält die leere Klausel). Falls dies nicht so ist, wählen wir eine Variable x und betrachten die Formeln $\varphi|_{x=0}$ und $\varphi|_{x=1}$. Dann ist φ erfüllbar, falls wenigstens eine der beiden Formeln $\varphi|_{x=0}$ oder $\varphi|_{x=1}$ erfüllbar ist. Weiterhin ist φ unerfüllbar, wenn beide Formeln $\varphi|_{x=0}$ und $\varphi|_{x=1}$ unerfüllbar sind. Um die Erfüllbarkeit von φ zu entscheiden, können wir also stattdessen die Erfüllbarkeit der einfacheren Formeln $\varphi|_{x=0}$ und $\varphi|_{x=1}$ untersuchen. Führt man diesen Prozess rekursiv für weitere Variablen durch, so erhält man den DPLL-Algorithmus.

Dieser nutzt folgende rekursive Prozedur:

```
1: function DPLL(φ, ℑ_1)
2:     if φ|_{ℑ_1} = 0 then return unerfüllbar
3:     if φ|_{ℑ_1} = 1 then return ℑ_1
4:     wähle eine Variable x in φ|_{ℑ_1} und a ∈ {0, 1}
5:     ℑ_2 := DPLL(φ, ℑ_1 ∪ [ℑ(x) := a])
6:     if ℑ_2 ≠ „unerfüllbar“ then return ℑ_2
7:     else return DPLL(φ, ℑ_1 ∪ [ℑ(x) := (1 − a)])
```

Der Aufruf des Algorithmus geschieht mittels $\mathrm{DPLL}(\varphi, \emptyset)$, wobei $\emptyset$ die leere Belegung ist, die keiner Variablen einen Wert zuweist.

Für die Wahl der Variablen x in Zeile 4 des DP-oder DPLL-ALgorithmus werden Heuristiken benutzt, z.B. können wir beim DP-Algorithmus ein x mit minimaler Anzahl von Vorkommen wählen (dann wird die Menge Δ klein). DPLL-Algorithmen gehören zu den in der Praxis am häufigsten eingesetzten Verfahren und bilden die Grundlage moderner SAT-Solver.

Ohne Beweis bemerken wir hier, dass die Laufzeit der DP- und DPLL-Algorithmen im schlechtesten Fall exponentiell in der Länge der Eingabeformel ist. Insbesondere sind DP- und DPLL-Algorithmen also nicht effizient (d.h. sie besitzen keine polynomielle Laufzeit). Dies ergibt sich aus Satz 63.

Übungsaufgabe 68. Prüfen Sie mit dem DPLL-Algorithmus, ob die Klauselmenge

$$\big\{\{x_1, x_2\}, \{x_1, \neg x_2\}, \{\neg x_1, \neg x_2\}\big\}$$

erfüllbar ist. Geben Sie die einzelnen Schritte an. Verwenden Sie für die Auswahl der Variablen die Ordnung $x_1 < x_2$.

Kapitel 4

Folgern und Schließen

In diesem Kapitel richten wir unseren Blick auf das aussagenlogische Schlussfolgern. Ziel unserer Betrachtung wird letztlich ein formales System von mechanisch anzuwendenden Regeln sein. Diese sollen es uns ermöglichen zu prüfen, ob eine Schlussfolgerung (Begründung) korrekt ist.

Eine Schlussfolgerung besteht aus *Prämissen* (Voraussetzungen) und einer *Konklusion* (Folgerung). Formal dargestellt wird sie beispielsweise auf eine der folgenden Weisen:

Schlussfolgerung
Prämisse
Konklusion

$$\frac{\begin{array}{c}\varphi\\ \psi\end{array}}{\theta} \qquad \frac{\begin{array}{c}\varphi\\ \psi\end{array}}{\therefore \theta} \qquad \frac{\varphi \qquad \psi}{\theta}$$

Hier bezeichnen φ und ψ Prämissen und θ eine Konklusion.

Als Hilfsmittel benötigen wir noch folgenden Begriff (den wir später noch formaler fassen): Eine Menge von Aussagen heißt *konsistent* (widerspruchsfrei), wenn die Aussagen alle gleichzeitig wahr sein können.

Konsistenz

Wenn wir überprüfen möchten, ob eine gegebene Schlussfolgerung (Begründung) korrekt ist, ob also die Konklusion logisch aus den Prämissen folgt, können wir dies wie folgt tun: Wir prüfen, ob die Negation der Konklusion im Widerspruch zu den Prämissen steht (sog. Widerspruchsbeweis).

Beispiel 69. Wir wollen prüfen, ob

$$\frac{\begin{array}{c}p_1 \to (\neg p_2 \to p_3)\\ p_1 \to \neg p_2\end{array}}{\therefore\ p_1 \to p_3}$$

richtig ist. Mit anderen Worten: Folgt $p_1 \to p_3$ semantisch aus $\{p_1 \to (\neg p_2 \to \neg p_3),\ p_1 \to \neg p_2\}$? In Zeichen: Gilt $\{p_1 \to (\neg p_2 \to \neg p_3),\ p_1 \to \neg p_2\} \vDash p_1 \to p_3$? Dazu negieren wir $p_1 \to p_3$ und prüfen die Konsistenz von

$$\varphi = (p_1 \to (\neg p_2 \to p_3)) \land (p_1 \to \neg p_2) \land \neg(p_1 \to p_3)$$

Das könnten wir jetzt – wie gehabt – mit Hilfe einer Wahrheitstafel klären:

p_1	p_2	p_3	φ
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	0
1	0	1	0
1	0	1	0
1	1	0	0
1	1	1	0

Dass die Negation $\neg(p_1 \to p_3)$ der Konklusion $p_1 \to p_3$ unverträglich mit den Prämissen ist, ergibt sich hierbei aus der letzten Spalte, die zeigt, dass die Aussagen nicht gleichzeitig wahr sein können (also φ unerfüllbar ist).

Wir behandeln im Rest dieses Kapitels verschiedene „mechanische“ (oder „syntaktische“) Wege, solche Schlussfolgerungen durchzuführen.

4.1. Ein Ableitungskalkül

Für eine Menge Φ von Voraussetzungen (Prämissen, Axiomen) definieren wir $\Phi^{\vDash}$ als die Menge aller Konklusionen (Folgerungen) aus Φ.

Folgerungsoperator

Definition 70. $\Phi^{\vDash} =_{\text{def}} \{\varphi : \Phi \vDash \varphi\}$ ist die Menge aller Formeln, die aus Φ gefolgert werden können. $^{\vDash}$ heißt *Folgerungsoperator*.

Wir wollen uns nun der klassischen zentralen Fragestellung der (philosophischen) Logik zuwenden, ob man das schwierige, semantische (und algorithmisch aufwendige) Folgern durch ein „mechanisches“, syntaktisches Ableiten (auch Schließen, Beweisen) mit Hilfe einfacher Regeln ersetzen kann. Wie sich zeigen wird, ist die Antwort hier im Falle der Aussagenlogik positiv.

Beweisen
Schließen
Ableiten

$\vdash$

Wir werden formal einen syntaktischen Begriff $\vdash$ für Ableiten/Schließen/Beweisen einführen. Das Schließen soll das Folgern nachbilden/imitieren, d. h. die eingeführten Schlussregeln werden *korrekt* sein und gültige Folgerungen wiederspiegeln. Das äußerst sich im sogenannten

Korrektheitssatz

Korrektheitssatz: $\Phi^{\vdash} \subseteq \Phi^{\vDash}$. (Was durch die Schlussregeln erzeugt wird, kann auch gefolgert werden, ist also semantisch korrekt.)

Ein wichtiges Ziel bei der Definition von $\vdash$ ist auch ein Vollständigkeitssatz, der zeigen soll, dass die eingeführten Schlussregeln tatsächlich ausreichend sind, um mit ihnen alle gültigen Folgerungen beweisen zu können:

Vollständigkeitssatz: $\Phi^{\models} \subseteq \Phi^{\vdash}$ (Alles, was gefolgert werden kann, kann auch durch die Schlussregeln erzeugt werden.) Vollständigkeitssatz

Es sei nochmals darauf hingewiesen, dass wir wegen der Möglichkeit der äquivalenten Ersetzbarkeit die Konnektoren $\rightarrow$ und $\leftrightarrow$ aus unseren Formeln eliminiert hatten. Wir betrachten also nur Formeln mit den Konnektoren $\wedge$, $\vee$ und $\neg$.

Definition 71. Induktive Definition des *Schließens* oder *Ableitens* oder *Beweisens*, d.h. induktive Definition von $\Phi \vdash \varphi$. Schließen Ableiten Beweisen

Induktionsanfang: Falls $\varphi \in \Phi$, so ist $\Phi \vdash \varphi$. $\vdash$

Induktionsschritt: Es gelten die folgenden *Schlussregeln*: Schlussregel

- Regel der Fallunterscheidung: Falls $(\Phi \cup \{\varphi'\}) \vdash \varphi$ und $(\Phi \cup \{\neg\varphi'\}) \vdash \varphi$, dann auch $\Phi \vdash \varphi$.
- Regel des indirekten Beweises: Falls $(\Phi \cup \{\neg\varphi\}) \vdash \varphi'$ und $(\Phi \cup \{\neg\varphi\}) \vdash \neg\varphi'$, dann auch $\Phi \vdash \varphi$.
- Abtrennungsregel, modus ponens: Falls $\Phi \vdash (\varphi' \vee \varphi)$ und $\Phi \vdash \neg\varphi'$, dann auch $\Phi \vdash \varphi$. (Man beachte hier, dass $(\varphi' \vee \varphi) \equiv (\neg\varphi' \rightarrow \varphi)$; der modus ponens stellt also sozusagen die klassische Deduktionsregel schlechthin dar.)
- Einführung der Alternative: Falls $\Phi \vdash \varphi$, dann auch $\Phi \vdash (\varphi \vee \varphi')$ und $\Phi \vdash (\varphi' \vee \varphi)$ für beliebige $\varphi' \in \mathsf{Form}_{\mathrm{AL}}$.
- Einführung der Konjunktion: Falls $\Phi \vdash \varphi$ und $\Phi \vdash \varphi'$, dann auch $\Phi \vdash (\varphi \wedge \varphi')$.
- Auflösung der Konjunktion: Falls $\Phi \vdash (\varphi \wedge \varphi')$, dann auch $\Phi \vdash \varphi$ und $\Phi \vdash \varphi'$.

$\Phi^{\vdash} =_{\mathrm{def}} \{\varphi : \Phi \vdash \varphi\}$ ($^{\vdash}$ heißt *Ableitungsoperator*).

Ein derartiges System von Schlussregeln wird in der mathematischen Logik auch *Ableitungskalkül* oder kurz *Kalkül* genannt. Ableitungskalkül Kalkül

Ein Beweis sollte stets endlich sein, das heißt, durch einen endlichen Text beschreibbar sein. Diese natürlich Forderung spiegelt sich in folgendem Satz wieder.

Satz 72. Ist $\varphi \in \Phi^{\vdash}$, so gibt es ein endliches $\Phi_0 \subseteq \Phi$ mit $\varphi \in \Phi_0^{\vdash}$. Endlichkeitssatz der Ableitung

Beweis Bei der Ableitung einer Formel φ durch einmalige Anwendung einer Ableitungsregel werden höchstens zwei Formeln benötigt. Also werden bei der Ableitung von φ aus Φ werden nur endlich viele Formeln aus Φ benötigt. Demnach existiert eine endliche Menge $\Phi_0 \subseteq \Phi$ mit $\varphi \in \Phi_0^{\vdash}$. ■

4.2. Regeln des natürlichen Schließens

Die im vorherigen Abschnitt angegebenen Schlussregeln gehen auf den deutschen Logiker Gerhard Gentzen zurück. Da sie allgemein üblichen Schlussweisen der Mathematik nachgebildet sind, spricht man auch vom Kalkül des natürlichen Schließens. Gentzen Kalkül des natürlichen Schließens

Die Regeln werden oft graphisch eingängig wie folgt dargestellt.

FU {Regel der Fallunterscheidung}

$$\frac{\begin{array}{cc}[\varphi'] & [\neg\varphi'] \\ \vdots & \vdots \\ \varphi & \varphi\end{array}}{\varphi}$$

RAA {Regel des indirekten Beweises}

$$\frac{\begin{array}{cc}[\neg\varphi] & [\neg\varphi] \\ \vdots & \vdots \\ \varphi' & \neg\varphi'\end{array}}{\varphi}$$

MP {Modus ponens}

$$\frac{\varphi' \vee \varphi \qquad \neg\varphi'}{\varphi}$$

$\vee$I {$\vee$-Einführung}

$$\frac{\varphi}{\varphi \vee \varphi'} \qquad \frac{\varphi}{\varphi' \vee \varphi}$$

$\wedge$I {$\wedge$-Einführung}

$$\frac{\varphi \qquad \varphi'}{\varphi \wedge \varphi'}$$

$\wedge$E {$\wedge$-Auflösung}

$$\frac{\varphi \wedge \varphi'}{\varphi} \qquad \frac{\varphi \wedge \varphi'}{\varphi'}$$

Zu den Bezeichnungen: „RAA“ steht für *reduction ad absurdum*, die lateinische Bezeichnung für einen Widerspruchsbeweis. Die Buchstaben „E“ und „I“ in den weiteren Regeln stehen für engl. *elimination* und *introduction*.

Die Regeln sind so zu verstehen, wie zu Beginn dieses Kapitels erläutert: Über dem Strich werden die Prämissen und unter dem Strich wird die Konklusion aufgeführt. Die Regeln der Fallunterscheidung und des indirekten Beweises sind etwas komplizierter, da sie jeweils temporär Annahmen (Hypothesen) einführen, die aber durch Anwendung der Regel wieder verschwinden. Diese Hypothesen werden in eckigen Klammern $[\cdot]$ dargestellt. Die Regel der Fallunterscheidung lässt sich also wie folgt lesen: Falls eine Abteilung von φ unter der Annahme einer Formel φ' sowie eine Abteilung von φ unter der Annahme $\neg\varphi'$ existieren, so kann φ (ohne Annahme) abgeleitet werden (denn einer der beiden Fälle φ' oder $\neg\varphi'$ muss ja immer eintreten). Die Regel des indirekten Beweises sagt aus: Falls unter der Annahme der Formel $\neg\varphi$ für eine beliebige Formel φ' sowohl φ' als auch die Negation $\neg\varphi'$ abgeleitet werden kann (also ein Widerspruch abgeleitet werden kann), dann schließe dass φ gilt, also die Annahme falsch war.

Die restlichen vier Regeln geben an, wie aus einer oder zwei Formeln als Prämisse(n) jeweils eine Konklusion geschlossen wird. Der Inhalt der Regeln ist leicht nachzuvollziehen. Man beachte, dass der *modus ponens* mithilfe des Konnektors $\rightarrow$ einfacher so geschrieben werden kann:

$$\frac{\varphi' \rightarrow \varphi \quad \varphi'}{\varphi}\ (\mathrm{MP}).$$

Alle Schlussregeln entsprechen damit üblichen beweistheoretischen Methoden. Die Bezeichnung „Kalkül des natürlichen Schließens“ ist also wohl begründet.

Die Ableitung einer Formel lässt sich unter Verwendung dieser graphischen Regelnotationen, ähnlich wie im Falle des Resolutionskalküls, in Baumform darstellen, wie wir in den folgenden Beispiel zeigen. Die Wurzel des Baumes bildet dabei die Formel, die letzendlich abgeleitet wird. Die Blätter stellen die Prämissen der Ableitung dar. Formeln, die an Blättern in eckigen Klammern $[\cdot]$ vorkommen, sind die Annahmen (Hypothesen) zu den Regeln der Fallunterscheidung oder des indirekten Beweises

und stellen daher keine eigentlichen Prämissen dar. Die inneren Knoten des Baumes entsprechen Formeln, die als Zwischenschritt im Laufe der Ableitung entstehen. Ist die Wurzel des Baumes also eine Formel $\varphi \in \mathsf{Form}_{\mathrm{AL}}$ und sind die Formeln der Blätter (ohne die Hypothesen) alle in der Formelmenge $\Phi \subseteq \mathsf{Form}_{\mathrm{AL}}$ enthalten, so wird $\Phi \vdash \varphi$ gezeigt.

Beispiel 73. Wir betrachten als erstes die Ableitung eines einfachen logischen Gesetzes.

$$\dfrac{\dfrac{[\varphi]}{\neg\varphi \vee \varphi}\,(\vee\mathrm{I}) \quad \dfrac{[\neg\varphi]}{\neg\varphi \vee \varphi}\,(\vee\mathrm{I})}{\neg\varphi \vee \varphi}\,(\mathrm{FU})$$

Der Ableitungsbaum hat zwei Blätter, die mit den Formeln φ und $\neg\varphi$ markiert sind. Diese bilden jedoch die Annahmen für die Anwendung der Regel der Fallunterscheidung. Für die Ableitung der Wurzel werden also keine Prämissen benötigt. Wir haben gezeigt: $\{\} \vdash \neg\varphi \vee \varphi$. Für jede Formel φ kann also $\neg\varphi \vee \varphi$ ohne Prämissen bewiesen werden. (Unter Verwendung des noch zu beweisenden Korrektheitssatzes handelt es sich also um aussagenlogische Tautologien). Dieses logische Gesetz wird mit *tertium non datur* („etwas Drittes existiert nicht“) bezeichnet. Es soll ausdrücken, dass stets entweder $\neg\varphi$ oder φ gelten muss und es keine dritte Möglichkeit gibt.

Wir wollen diese Ableitung in den weiteren Beispielen verwenden und stellen sie daher in Form einer neuen Regel ohne Prämissen

$$\dfrac{}{\neg\varphi \vee \varphi}\,(\mathrm{TND})$$

dar.

Beispiel 74. Wir betrachten als nächstes eine Reihe nützliche Beispielableitungen. Zunächst wenden wir uns der doppelten Negation zu.

$$\dfrac{\neg\neg\varphi \quad [\neg\varphi]}{\varphi}\,(\mathrm{RAA})$$

Dieser Ableitungsbaum hat als Blätter die Formeln $\neg\neg\varphi$ und $\neg\varphi$. Letzere bildet jedoch die Annahme zu der Regel (RAA). Als Prämisse bleibt lediglich $\neg\neg\varphi$. Die Ableitung zeigt also, dass für eine beliebige Formel φ gilt, dass $\{\neg\neg\varphi\} \vdash \phi$. Auch diese Ableitung werden wir in den folgenden Beispielen wiederverwenden und notieren dies in Form der Regel

$$\dfrac{\neg\neg\varphi}{\varphi}\,(\neg\neg\mathrm{E}).$$

Eine „umgekehrte“ Wirkung erzielt die folgende Ableitung.

$$\dfrac{\varphi \quad \dfrac{[\neg\neg\neg\varphi]}{\neg\varphi}\,(\neg\neg\mathrm{E})}{\neg\neg\varphi}\,(\mathrm{RAA})$$

Wir werden dafür die Abkürzung

$$\dfrac{\varphi}{\neg\neg\varphi}\,(\neg\neg\mathrm{I})$$

verwenden.

Eine weitere hilfreiche Ableitung beweist die Kommutativität der Disjunktion.

$$\dfrac{\dfrac{\dfrac{\varphi \vee \psi \quad [\neg\varphi]}{\psi}\,(\mathrm{MP})}{\psi \vee \varphi}\,(\vee\mathrm{I}) \qquad \dfrac{[\varphi]}{\psi \vee \varphi}\,(\vee\mathrm{I})}{\psi \vee \varphi}\,(\mathrm{FU})$$

Wir wollen hierfür die Abkürzung

$$\dfrac{\varphi \vee \psi}{\psi \vee \varphi}\,(\vee\mathrm{KG})$$

verwenden.

Übungsaufgabe 75. Leiten Sie das Kommutativitätsgesetz für die Konjunktion ab: $\{\varphi \wedge \psi\} \vdash \psi \wedge \varphi$.

Beispiel 76. In diesem Beispiel und der folgenden Aufgabe geht es um die Gesetze von de Morgan. Wir betrachten folgende Ableitung.

$$\dfrac{\dfrac{\varphi \wedge \psi}{\psi}\,(\wedge\mathrm{E}) \qquad \dfrac{\dfrac{[\neg\neg(\neg\varphi \vee \neg\psi)]}{\neg\varphi \vee \neg\psi}\,(\neg\neg\mathrm{E}) \qquad \dfrac{\dfrac{\varphi \wedge \psi}{\varphi}\,(\wedge\mathrm{E})}{\neg\neg\varphi}\,(\neg\neg\mathrm{I})}{\neg\psi}\,(\mathrm{MP})}{\neg(\neg\varphi \vee \psi)}\,(\mathrm{RAA})$$

Wir haben also gezeigt: $\{\varphi \wedge \psi\} \vdash \neg(\neg\varphi \vee \neg\psi)$.

Übungsaufgabe 77. Beweisen Sie:

(i) $\{\neg(\neg\varphi \vee \neg\psi)\} \vdash \varphi \wedge \psi$.

(ii) $\{\varphi \vee \psi\} \vdash \neg(\neg\varphi \wedge \neg\psi)$.

(iii) $\{\neg(\neg\varphi \wedge \neg\psi)\} \vdash \varphi \vee \psi$.

4.3. Der Vollständigkeitssatz

Wir wollen als nächstes zeigen, dass die gegebene Definition des Ableitungskalküls unsere eingangs gestellt Forderung erfüllt, nämlich $\Phi^{\vDash} = \Phi^{\vdash}$, d. h. also dass semantisches Folgern und syntaktisches Beweisen das Gleiche leisten. Das geschieht, wie auch bereits erläutert, in zwei Schritten: Wir weisen zunächst Korrektheit und sodann Vollständigkeit des Kalküls nach.

4.3.1. Korrektheit

Korrektheitssatz

Satz 78. Es gilt $\Phi^{\vdash} \subseteq \Phi^{\vDash}$.

Beweis Wir definieren zunächst die Menge $\Phi_n^\vdash$ aller Formeln, die aus Φ durch Anwendungen von n Schlussregeln abgeleitet werden können.

Wir definieren $\Phi_n^\vdash$ gleichzeitig für alle Φ durch Induktion über n:

Im Induktionsanfang ist $\Phi_0^\vdash = \Phi$.

Im Induktionsschritt soll $\Phi_{n+1}^\vdash$ die Menge aller Formeln sein, die zu $\Phi_n^\vdash$ gehören oder die zu $\Phi^\vdash$ gehören und durch Anwendung einer Schlussregel aus Formeln irgendwelcher $\Psi_n^\vdash$ abgeleitet werden können. Wir setzen also

$$\begin{aligned}\Phi_{n+1}^\vdash = \Phi_n^\vdash &\cup \{\varphi \mid \text{es existiert } \varphi' \text{ mit } \varphi \in (\Phi \cup \{\varphi'\})_n^\vdash \cap (\Phi \cup \{\neg\varphi'\})_n^\vdash\} \\ &\cup \{\varphi \mid \text{es existiert } \varphi' \text{ mit } \{\varphi', \neg\varphi'\} \subseteq (\Phi \cup \{\neg\varphi\})_n^\vdash\} \\ &\cup \{\varphi \mid \text{es existiert } \varphi' \text{ mit } \{\varphi' \vee \varphi, \neg\varphi'\} \subseteq \Phi_n^\vdash\} \\ &\cup \{(\varphi \vee \varphi') \mid \varphi \in \Phi_n^\vdash \text{ und } \varphi' \text{ beliebig}\} \\ &\cup \{(\varphi' \vee \varphi) \mid \varphi \in \Phi_n^\vdash \text{ und } \varphi' \text{ beliebig}\} \\ &\cup \{(\varphi \wedge \varphi') \mid \varphi, \varphi' \in \Phi_n^\vdash\} \\ &\cup \{\varphi \mid \text{es existiert } \varphi' \text{ mit } (\varphi \wedge \varphi') \in \Phi_n^\vdash\} \\ &\cup \{\varphi \mid \text{es existiert } \varphi' \text{ mit } (\varphi' \wedge \varphi) \in \Phi_n^\vdash\}.\end{aligned}$$

Offensichtlich gilt $\Phi = \Phi_0^\vdash \subseteq \Phi_1^\vdash \subseteq \Phi_2^\vdash \subseteq \ldots$ und $\Phi^\vdash = \bigcup_{n\geq 0} \Phi_n^\vdash$.

Wir zeigen nun $\Phi_n^\vdash \subseteq \Phi^\vDash$ gleichzeitig für alle Φ durch Induktion über n:

Induktionsanfang Offenbar ist $\Phi_0^\vdash = \Phi \subseteq \Phi^\vDash$.

Induktionsschritt Sei nun $\varphi \in \Phi_{n+1}^\vdash$. Wir betrachten die Fälle, die gemäß der obigen Definition auftreten können:

Fall 1 Ist $\varphi \in \Phi_n^\vdash$, so ist $\varphi \in \Phi^\vDash$ nach Induktionsvoraussetzung.

Fall 2 Es existiert ein φ' mit $\varphi \in (\Phi \cup \{\varphi'\})_n^\vdash$ und $\varphi \in (\Phi \cup \{\neg\varphi'\})_n^\vdash$. Nach Induktionsvoraussetzung gilt $\varphi \in (\Phi \cup \{\varphi'\})^\vDash$ und $\varphi \in (\Phi \cup \{\neg\varphi'\})^\vDash$. Ist nun $\mathfrak{I}(\Phi) = 1$, so ergibt sich $\mathfrak{I}(\Phi \cup \{\varphi'\}) = 1$ oder $\mathfrak{I}(\Phi \cup \{\neg\varphi'\}) = 1$. Es folgt $\mathfrak{I}(\varphi) = 1$.

Fall 3 Es existiert ein φ' mit $\varphi', \neg\varphi' \in (\Phi \cup \{\neg\varphi\})_n^\vdash$. Nach Induktionsvoraussetzung gilt $\varphi', \neg\varphi' \in (\Phi \cup \{\neg\varphi\})^\vDash$. Ist nun $\mathfrak{I}(\Phi) = 1$, so ist $\mathfrak{I}(\neg\varphi) = 0$. Es folgt $\mathfrak{I}(\varphi) = 1$.

Fall 4 Es existiert ein φ' mit $(\varphi' \vee \varphi), \neg\varphi' \in \Phi_n^\vdash$. Nach Induktionsvoraussetzung gilt $(\varphi' \vee \varphi), \neg\varphi' \in \Phi^\vDash$. Ist nun $\mathfrak{I}(\Phi) = 1$, so ist $\mathfrak{I}(\varphi' \vee \varphi) = \mathfrak{I}(\neg\varphi') = 1$. Es folgt $\mathfrak{I}(\varphi) = 1$.

Fall 5 $\varphi = (\varphi' \vee \varphi'')$ für geeignete $\varphi' \in \Phi_n^\vdash$ und φ''. Nach Induktionsvoraussetzung gilt $\Longrightarrow \varphi' \in \Phi^\vDash$. Ist nun $\mathfrak{I}(\Phi) = 1$, so ist $\mathfrak{I}(\varphi') = 1$. Es folgt $\mathfrak{I}(\varphi' \vee \varphi'') = 1$.

Fall 6 Analog zu Fall 5.

Fall 7 $\varphi = (\varphi' \wedge \varphi'')$ für geeignete $\varphi', \varphi'' \in \Phi_n^\vdash$. Nach Induktionsvoraussetzung gilt $\Longrightarrow \varphi', \varphi'' \in \Phi^\vDash$. Ist nun $\mathfrak{I}(\Phi) = 1$, so ist $\mathfrak{I}(\varphi') = \mathfrak{I}(\varphi'') = 1$. Es folgt $\mathfrak{I}(\varphi' \wedge \varphi'') = 1$.

Fall 8 Es existiert ein φ' mit $(\varphi \wedge \varphi') \in \Phi_n^\vdash$. Nach Induktionsvoraussetzung gilt $\Longrightarrow (\varphi \wedge \varphi') \in \Phi^\vDash$. Ist nun $\mathfrak{I}(\Phi) = 1$, so ist $\mathfrak{I}(\varphi \wedge \varphi') = 1$. Es folgt $\mathfrak{I}(\varphi) = 1$.

Fall 9 Es existiert ein φ' mit $(\varphi' \wedge \varphi) \in \Phi_n^\vdash$. Analog zu Fall 8.

■

4.3.2. Widerspruchsfreiheit (Konsistenz) von Formelmengen

Ein wichtiges Hilfsmittel auf dem Weg zum Beweis des Vollständigkeitssatzes ist folgender Begriff.

Konsistenz
Widerspruchsfreiheit

Definition 79. Eine Formelmenge Φ heißt *konsistent* (*widerspruchsfrei*), falls es keine Formel φ gibt mit $\varphi, \neg\varphi \in \Phi^{\vdash}$.

Aus einer konsistenten Formelmenge ist also kein Widerspruch ableitbar. Intuitiv gesehen sollten solche Formelmengen also erfüllbar sein. Dieses wichtige Resultat werden wir später beweisen. Zuvor halten wir jedoch folgende Eigenschaften fest.

Satz 80. (i) Ist Φ nicht konsistent, so gilt $\Phi^{\vdash} = \mathsf{Form}_{\mathrm{AL}}$.

(ii) Ist Φ konsistent, so ist $\Phi \cup \{\varphi\}$ oder $\Phi \cup \{\neg\varphi\}$ konsistent.

Beweis

(i) Ist Φ nicht konsistent, so es existiert ein φ' mit $\varphi', \neg\varphi' \in \Phi^{\vdash}$. Also gilt auch $\Longrightarrow \varphi', \neg\varphi' \in (\Phi \cup \{\neg\varphi\})^{\vdash}$ für alle $\varphi \in \mathsf{Form}_{\mathrm{AL}}$. Mithilfe der Regel des indirekten Beweises ergibt sich direkt, dass $\varphi \in \Phi^{\vdash}$ für alle $\varphi \in \mathsf{Form}_{\mathrm{AL}}$.

(ii) Angenommen $(\Phi \cup \{\varphi\})$ und $(\Phi \cup \{\neg\varphi\})$ seien nicht konsistent. Aus Aussage (i) dieses Satzes folgt dann, dass $\varphi, \neg\varphi \in (\Phi \cup \{\varphi\})^{\vdash}$ und $\varphi, \neg\varphi \in (\Phi \cup \{\neg\varphi\})^{\vdash}$. Mithilfe der Regel der Fallunterscheidung ergibt sich direkt, dass $\varphi, \neg\varphi \in \Phi^{\vdash}$; also ist Φ nicht konsistent.

■

4.3.3. Vollständigkeit von Formelmengen

Als letztes Hilfsmittel auf dem Weg zum Vollständigkeitssatz benötigen wir noch den Begriff der Vollständigkeit von Formelmengen.

Definition 81. Eine Formelmenge Φ heißt *vollständig*, falls für jedes φ gilt: $\varphi \in \Phi$ oder $\neg\varphi \in \Phi$.

Interessante Eigenschaften ergeben sich für Formelmengen Φ, die zugleich konsistent und vollständig sind. Mitgliedschaft in solchen Φ ist sozusagen konsistent mit der semantischen Wahrheitsdefinition, wie wir als nächstes zeigen.

Satz 82. Sei Φ konsistent und vollständig. Dann gilt:

(i) $\Phi = \Phi^{\vdash}$.

(ii) $\neg\varphi \in \Phi \Leftrightarrow \varphi \notin \Phi$.

(iii) $(\varphi_1 \vee \varphi_2) \in \Phi \Leftrightarrow \varphi_1 \in \Phi$ oder $\varphi_2 \in \Phi$.

(iv) $(\varphi_1 \wedge \varphi_2) \in \Phi \Leftrightarrow \varphi_1 \in \Phi$ und $\varphi_2 \in \Phi$.

Beweis

(i) Angenommen, $\Phi^{\vdash} \subset \Phi^{\models}$, dann existiert ein $\varphi \in \Phi^{\vdash} \setminus \Phi$. Da Φ vollständig ist und $\varphi \notin \Phi$, folgt $\neg\varphi \in \Phi \subseteq \Phi^{\vdash}$. Aus beiden Tatsachen ergibt sich, dass Φ nicht konsistent ist.

(ii) Da Φ vollständig ist, ist $\varphi \in \Phi$ oder $\neg\varphi \in \Phi$. Da Φ konsistent ist, ist $\Longrightarrow \varphi \notin \Phi^{\vdash} = \Phi$ oder $\neg\varphi \notin \Phi^{\vdash} = \Phi$. Zusammengenommen folgt also, dass $\neg\varphi \in \Phi \Leftrightarrow \varphi \notin \Phi$.

(iii) „$\Rightarrow$“: Sei $(\varphi_1 \vee \varphi_2) \in \Phi$. Ist $\varphi_1 \in \Phi \Longrightarrow$, so ist nichts zu zeigen. Sei nun $\varphi_1 \notin \Phi$. Da Φ vollständig ist, folgt $\neg\varphi_1 \in \Phi = \Phi^{\vdash}$. Da aber $(\varphi_1 \vee \varphi_2) \in \Phi = \Phi^{\vdash}$, ergibt sich wegen modus ponens, dass $\varphi_2 \in \Phi^{\vdash} = \Phi$.

„$\Leftarrow$“: Sei $\varphi_1 \in \Phi = \Phi^{\vdash}$ oder $\varphi_2 \in \Phi = \Phi^{\vdash}$. Aus der Regel von der Einführung der Alternative folgt $(\varphi_1 \vee \varphi_2) \in \Phi^{\vdash} = \Phi$.

(iv) „$\Rightarrow$“: Sei $(\varphi_1 \wedge \varphi_2) \in \Phi = \Phi^{\vdash}$. Aus der Regel der Auflösung der Konjunktion folgt $\varphi_1 \in \Phi^{\vdash} = \Phi$ und $\varphi_2 \in \Phi^{\vdash} = \Phi$.

„$\Leftarrow$“: Seien $\varphi_1 \in \Phi = \Phi^{\vdash}$ und $\varphi_2 \in \Phi = \Phi^{\vdash}$. Aus der Regel der Einführung der Konjunktion folgt $(\varphi_1 \wedge \varphi_2) \in \Phi^{\vdash} = \Phi$.

■

Die nach Definition 79 erläuterte Intuition, dass jede Konsistenz Erfüllbarkeit impliziert, ist der Inhalt des folgenden Satzes.

Satz 83. Sei Φ eine Formelmenge. Es gilt: Φ genau dann konsistent, wenn Φ ein Modell hat.

Satz von Henkin
Leon Henkin (1921–2006)
amerikanischer Logiker

Beweis („$\Leftarrow$“) Sei Φ nicht konsistent, d. h. es existiert ein φ mit $\varphi, \neg\varphi \in \Phi^{\vdash}$. Aus dem Korrektheitssatz folgt, dass $\varphi, \neg\varphi \in \Phi^{\models}$. Dann gilt für jedes Modell $\mathfrak{I}$ von Φ, dass $\mathfrak{I}(\varphi) = \mathfrak{I}(\neg\varphi) = 1$. Es folgt: Φ hat kein Modell.

(„$\Rightarrow$“) Sei nun Φ konsistent. Wir erweitern zunächst die Menge Φ zu einer vollständigen und konsistenten Menge Ψ:

Es seien $\varphi_0, \varphi_1, \varphi_2, \ldots$ alle Formeln aus $\mathsf{Form}_{\mathrm{AL}}$. Wir definieren induktiv die konsistenten Mengen Φ_n für $n = 0, 1, 2, \ldots$:

$$\Phi_0 =_{\mathrm{def}} \Phi \quad \text{konsistent}$$

$$\Phi_{n+1} =_{\mathrm{def}} \begin{cases} \Phi_n \cup \{\varphi_n\} & \text{falls } \Phi_n \cup \{\varphi_n\} \text{ konsistent} \\ \Phi_n \cup \{\neg\varphi_n\} & \text{sonst} \end{cases}$$

Offenbar ist jede Menge Φ_{n+1} konsistent, da entweder $\Phi_n \cup \{\varphi_n\}$ oder $\Phi_n \cup \{\neg\varphi_n\}$ konsistent ist (Satz 80, (ii)).

Wir definieren nun $\Psi =_{\mathrm{def}} \bigcup_{n\geq 0} \Phi_n$.

Direkt aus der Definition ergibt sich $\Phi = \Phi_0 \subseteq \Phi_1 \subseteq \Phi_2 \subseteq \Phi_3 \subseteq \cdots \subseteq \bigcup_{n\geq 0} \Phi_n = \Psi$. Bevor wir ein Modell für Φ angeben, beweisen wir einige Eigenschaften von Ψ:

Behauptung 1 Ψ ist vollständig.

Beweis der Behauptung 1 Für jedes φ_n gilt:

$$\varphi_n \in \Phi_{n+1} \subseteq \Psi \text{ oder } \neg\varphi_n \in \Phi_{n+1} \subseteq \Psi.$$

Behauptung 2 Ψ ist konsistent.

Beweis der Behauptung 2 Angenommen, Ψ sei nicht konsistent. Dann existiert $\varphi \in \mathsf{Form}_{\mathrm{AL}}$ mit $\varphi, \neg\varphi \in \Psi^{\vdash}$. Nach dem Endlichkeitssatz der Ableitung existiert eine endliche Menge $\Psi_0 \subseteq \Psi$ mit $\varphi, \neg\varphi \in \Psi_0^{\vdash}$. Da Ψ_0 endlich ist, existiert ein $n_0 \geq 0$ mit $\Psi_0 \subseteq \Phi_{n_0}$. Also ergibt sich: $\varphi, \neg\varphi \in \Psi_0^{\vdash} \subseteq \Phi_{n_0}^{\vdash}$ und demnach ist Φ_{n_0} nicht konsistent. Dies ist ein Widerspruch zu der Tatsache, dass alle Φ_n konsistent sind. Es folgt: Ψ ist konsistent, und damit ist Behauptung 2 bewiesen.

Wir definieren nun eine Belegung $\mathfrak{I}_\Psi$ durch

$$\mathfrak{I}_\Psi(p_i) =_{\mathrm{def}} \begin{cases} 1 & \text{falls } p_i \in \Psi \\ 0 & \text{sonst} \end{cases}$$

Die folgende Aussage zeigt, dass $\mathfrak{I}_\Psi$ ein Modell für Ψ und damit auch für Φ ist.

Behauptung 3 Für alle $\varphi \in \mathsf{Form}_{\mathrm{AL}}$ gilt: $\varphi \in \Psi \Leftrightarrow \mathfrak{I}_\Psi(\varphi) = 1$.

Beweis der Behauptung 3 Wir beweisen die Behauptung durch Induktion über den Aufbau der Formeln.

(IA) $\varphi = p \in \mathsf{Var}$. Es gilt $p \in \Psi \Leftrightarrow \mathfrak{I}_\Psi(p_i) = 1$ nach Definition von $\mathfrak{I}_\Psi$.

(IS) Fall 1 $\varphi = \neg\varphi'$. Es gilt:

$$\begin{aligned} \neg\varphi' \in \Psi &\Leftrightarrow \varphi' \notin \Psi && \text{(Satz 82 (ii))} \\ &\Leftrightarrow \mathfrak{I}_\Psi(\varphi') = 0 && \text{(Induktionsvoraussetzung)} \\ &\Leftrightarrow \mathfrak{I}_\Psi(\neg\varphi') = 1 \end{aligned}$$

Fall 2 $\varphi = (\varphi_1 \vee \varphi_2)$. Es gilt:

$$\begin{aligned} (\varphi_1 \vee \varphi_2) \in \Psi &\Leftrightarrow \varphi_1 \in \Psi \text{ oder } \varphi_2 \in \Psi && \text{(Satz 82 (iii))} \\ &\Leftrightarrow \mathfrak{I}_\Psi(\varphi_1) = 1 \text{ oder } I_\Psi(\varphi_2) = 1 && \text{(Induktionsvoraussetzung)} \\ &\Leftrightarrow \mathfrak{I}_\Psi(\varphi_1 \vee \varphi_2) = 1 \end{aligned}$$

Fall 3 $\varphi = (\varphi_1 \wedge \varphi_2)$. Es gilt:

$$\begin{aligned} (\varphi_1 \wedge \varphi_2) \in \Psi &\Leftrightarrow \varphi_1 \in \Psi \text{ und } \varphi_2 \in \Psi && \text{(Satz 82 (iv))} \\ &\Leftrightarrow \mathfrak{I}_\Psi(\varphi_1) = 1 \text{ und } \mathfrak{I}_\Psi(\varphi_2) = 1 && \text{(Induktionsvoraussetzung)} \\ &\Leftrightarrow \mathfrak{I}_\Psi(\varphi_1 \wedge \varphi_2) = 1 \end{aligned}$$

■

Satz 84. $\Phi^{\models} \subseteq \Phi^{\vdash}$

Vollständigkeitssatz

Der Vollständigkeitssatz in seiner allgemeinen prädikatenlogischen Form (siehe nächstes Kapitel) wurde zuerst von Kurt Gödel erbracht.

Kurt Gödel (1906–1978) österreichisch-amerikanischer Logiker

Beweis Sei $\varphi \in \Phi^{\models}$. Für eine beliebige Interpretation $\mathfrak{I}$ gilt dann: Falls $\mathfrak{I}(\Phi) = 1$, dann ist auch $\mathfrak{I}(\varphi) = 1$ und also $\mathfrak{I}(\neg\varphi) = 0$. Daher ist auch $\mathfrak{I}(\Phi \cup \{\neg\varphi\}) = 0$. Da $\mathfrak{I}$ beliebig, folgt: $\Phi \cup \{\neg\varphi\}$ hat kein Modell. Nach dem Satz von Henkin ist also $\Phi \cup \{\neg\varphi\}$ nicht konsistent. Also es existiert ein φ' mit $\varphi', \neg\varphi' \in (\Phi \cup \{\neg\varphi\})^{\vdash}$. Nach der Regel des indirekten Beweises folgt: $\varphi \in \Phi^{\vdash}$. ■

Aus Satz 78 und Satz 84 folgt direkt

Satz 85. Für jede Menge Φ von Formeln gilt $\Phi^{\models} = \Phi^{\vdash}$.

Wir haben also unser Ziel erreicht und gezeigt: Das semantische Folgern und das syntaktische Ableiten leisten das Gleiche.

4.4. Exkurs: Semantische Tableaus

Eine andere Möglichkeit, die Gültigkeit von Schlussfolgerungen zu überprüfen, bieten semantische Tableaus.

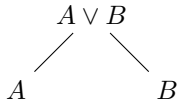
Ein solches semantisches Tableau können wir uns – informell – als eine Folge von Aussagen, die nach bestimmten Regeln gebildet und als Baum dargestellt werden, vorstellen. Die Regeln lauten dabei:

Regel 1: $A \wedge B$

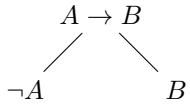
$A \wedge B$

B

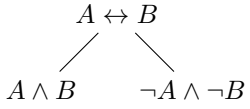
Regel 2: $A \vee B$



Regel 3: $A \to B$



Regel 4: $A \leftrightarrow B$

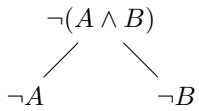


Regel 5: $\neg\neg A$

$\neg\neg A$

A

Regel 6: $\neg(A \wedge B)$



Regel 7: $\neg(A \vee B)$

$\neg(A \vee B)$

$\neg A$

$\neg B$

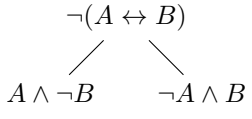
Regel 8: $\neg(A \to B)$

$\neg(A \to B)$

A

$\neg B$

Regel 9: $\neg(A \leftrightarrow B)$

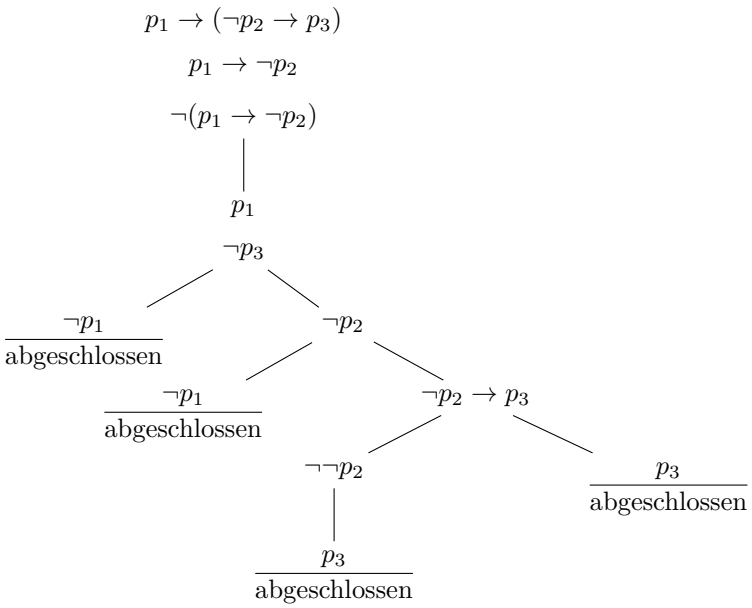


Regel 10:
Immer dann, wenn bei einem Ast eines Tableaus sowohl die logische Form A als auch ihre Negation $\neg A$ erscheint, liegt ein Widerspruch bei diesem Ast vor. Wir nennen den Ast dann „abgeschlossen“. Er wird nicht mehr erweitert (, da A und $\neg A$ nicht gleichzeitig wahr sein können).
Wenn alle Äste eines semantischen Tableaus abgeschlossen sind, stehen die Aussagen, aus denen das Tableau erstellt wurde, im Widerspruch zueinander.

Beispiel 86.

$$\begin{array}{lll} p_1 & \to & (\neg p_2 \to p_3) \\ p_1 & \to & \neg p_2 \\ \hline \therefore p_1 & \to & p_3 \end{array}$$

Tableau:



Kapitel 5

Modallogik

Wir können die Aussagenlogik um einen unären Konnektor $\Box$ erweitern. Wenn wir $\Box$ die Bedeutung „es gilt notwendigerweise φ“ geben, so bedeutet $\neg\Box\neg\varphi$ „es gilt möglicherweise φ“. (Wir kürzen $\neg\Box\neg\varphi$ durch $\Diamond\varphi$ ab.) Damit können wir die Modalitäten „notwendig“ und „möglich“ ausdrücken. Die so erhaltene Logik nennen wir dementsprechend *Modallogik*. Sie ist für uns Informatiker interessant, weil sie Grundlage des *Model Checking* ist.

notwendigerweise
möglicherweise
Modalität

5.1. Syntax der Modallogik

Die modale Sprache ist eine Erweiterung der Sprache der Aussagenlogik.

Syntaktisch sind die Formeln der Modallogik (ML) durch folgende Grammatik gegeben:

$$\varphi ::= p \mid \neg\varphi \mid \varphi \wedge \varphi \mid \varphi \vee \varphi \mid \varphi \to \varphi \mid \varphi \leftrightarrow \varphi \mid \Box\varphi \mid \Diamond\varphi,$$

wobei $p \in \mathsf{Var}$.

5.2. Semantik der Modallogik

Definition 87. Ein *Kripke Rahmen* (oder Rahmen, frame) $\mathcal{F}$ ist ein Paar (W, R) mit

- W ist eine nichtleere Menge (die Menge der *Welten*) und
- R ist eine binäre Relation auf W.

Kripke-Rahmen

Saul Kripke (*1940)
amerikanischer Philosoph
und Logiker

Definition 88. Ein *Modell* $\mathcal{M}$ der Modallogik ist ein Paar $(\mathcal{F}, V)$ mit

- $\mathcal{F} = (W, R)$ ist ein Rahmen
- $V : \mathsf{Var} \to \mathcal{P}(W)$ ist eine Abbildung, die jedem Atom p eine Menge $V(p)$ von Welten zuordnet ($\mathcal{P}(W)$ ist die Potenzmenge von W).

Definition 89. Seien φ, ψ modale Formeln. Sei $\mathcal{M} = (W, R, V)$ ein Modell und $w \in W$ eine Welt. Wir definieren induktiv die Gültigkeit einer Formel im Modell $\mathcal{M}$ in der Welt w:

- $\mathcal{M}, w \vDash 1$ immer
- $\mathcal{M}, w \vDash 0$ nie
- $\mathcal{M}, w \vDash p$ falls $w \in V(p)$ mit $p \in \mathsf{Var}$,
- $\mathcal{M}, w \vDash \neg\varphi$ falls nicht $\mathcal{M}, w \vDash \varphi$,
- $\mathcal{M}, w \vDash \varphi \wedge \psi$ falls $\mathcal{M}, w \vDash \varphi$ und $\mathcal{M}, w \vDash \psi$
- $\mathcal{M}, w \vDash \varphi \vee \psi$ falls $\mathcal{M}, w \vDash \varphi$ oder $\mathcal{M}, w \vDash \psi$
- $\mathcal{M}, w \vDash \Box\varphi$ falls für alle $v \in W$ mit $(w, v) \in R$ gilt $\mathcal{M}, v \vDash \varphi$.

Extension

Die Extension einer Formel φ in $\mathcal{M}$ ist definiert als $[\![\varphi]\!] = \{w \mid \mathcal{M}, w \vDash \varphi\}$. Ist $[\![\varphi]\!] = W$, d. h. gilt $\mathcal{M}, w \vDash \varphi$ für alle $w \in W$, so schreiben wir kurz: $\mathcal{M} \vDash \varphi$.

Da wir $\Diamond$ als Abkürzung für $\neg\Box\neg$ definiert haben, ergibt sich:

Satz 90. Sei φ eine modale Formel. Sei $\mathcal{M} = (W, R, V)$ ein Modell und $w \in W$ eine Welt. Dann gilt $\mathcal{M}, w \vDash \Diamond\varphi$ falls es eine Welt $v \in W$ gibt mit $(w, v) \in R$ und $\mathcal{M}, v \vDash \varphi$.

Die Interpretation der Standardoperatoren folgt der klassischen Aussagenlogik. Die modalen Operatoren $\Box$ und $\Diamond$ können je nach Anwendungsbereich verschieden interpretiert werden.

Die gängige modale Deutung ist

$\Box\varphi$ „φ ist notwendigerweise wahr“
$\Diamond\varphi$ „φ ist möglicherweise wahr“.

Daneben gibt es auch die deontische Deutung

$\Box\varphi$ „φ ist geboten“
$\Diamond\varphi$ „φ ist erlaubt“.

und die temporale Deutung

$\Box\varphi$ „φ gilt immer in der Zukunft“
$\Diamond\varphi$ „φ gilt irgendwann in der Zukunft“.

Definition 91.

erfüllbar

- Eine modale Formel φ ist *erfüllbar* wenn es ein Modell $\mathcal{M} = (W, R, V)$ und eine Welt $w \in W$ mit $\mathcal{M}, w \vDash \varphi$ gibt.

Tautologie

- Dual dazu ist φ eine modale *Tautologie* falls für jedes Modell $\mathcal{M} = (W, R, V)$ und jedes $w \in W$ gilt $\mathcal{M}, w \vDash \varphi$.

Definition 92. Sei R eine binäre Relation über W und seien $v, w \in W$ sowie $n \geq 1$. Dann heißt v von w aus *in n Schritten sichtbar* (kurz $wR^n v$) genau dann, wenn folgendes gilt:

(i) wRv, falls $n = 1$

(ii) Es existieren $u_1, \ldots, u_{n-1}$ mit

$$wRu_1, u_1Ru_2, \ldots, u_{n-2}Ru_{n-1}, u_{n-1}Rv,$$

falls $n > 1$.

5.3. Beispiele

Jede klassische Tautologie ist auch eine modale Tautologie. Die Formel

$$\Box(\varphi_1 \to \varphi_2) \to (\Box\varphi_1 \to \Box\varphi_2)$$

ist ein Beispiel für eine modale Tautologie.

Wir betrachten als komplexeres Beispiel die Modellierung einer Ampelschaltung in Modallogik, sh. Abb. 5.1. Die Welten $W = \{s_r, s_{ry}, s_g, s_y\}$ sind verknüpft durch die Relation $R = \{(s_{ry}, s_g), (s_g, s_y), (s_y, s_r), (s_r, s_{ry})\}$ (siehe Abb. 5.1). Die Variablen $\Phi = \{r, g, y\}$ (rot, grün, gelb) beschreiben, in welchen Zuständen eine Lampe aktiviert ist

$$\begin{aligned} V(r) &= \{s_{ry}, s_r\} \\ V(y) &= \{s_{ry}, s_y\} \\ V(g) &= \{s_g\}. \end{aligned}$$

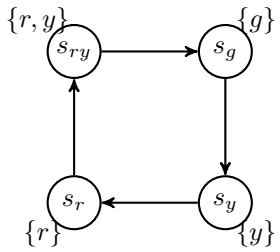


Abbildung 5.1.: Der Kripke-Rahmen der Ampelschaltung

In diesem Modell $\mathcal{M} = (W, R, V)$ gilt dann z.B.

$$\mathcal{M}, s_{ry} \vDash \Box(\neg r \wedge \neg y \wedge g).$$

Als zweite Formel betrachten wir $(r \wedge y \wedge \neg g) \to \neg\Diamond(r \wedge \neg y \wedge \neg g)$. Diese Formel gilt sogar für alle $w \in W$

$$\mathcal{M}, w \vDash (r \wedge y \wedge \neg g) \to \neg\Diamond(r \wedge \neg y \wedge \neg g).$$

Inhaltlich bedeutet dies: nach Rot-Gelb folgt niemals Rot. Da es hier offenbar nicht auf die Angabe der Welt ankommt, schreiben wir auch kurz

$$\mathcal{M} \vDash (r \wedge y \wedge \neg g) \to \neg\Diamond(r \wedge \neg y \wedge \neg g).$$

Wir betrachten eine dritte Formel. Für alle Formeln φ, alle Belegungen V' und alle Welten $w \in W$ gilt

$$(W, R, V'), w \vDash \varphi \leftrightarrow \Diamond\Diamond\Diamond\Diamond\varphi.$$

Inhaltlich heißt das, dass der Rahmen ein Viererzyklus ist. Da es hier nicht auf die Belegung ankommt, schreiben wir auch kurz

$$(W, R) \vDash \varphi \leftrightarrow \Diamond\Diamond\Diamond\Diamond\varphi.$$

Übungsaufgabe 93. Betrachten Sie den Rahmen $\mathcal{F} = (\{w_1, w_2, w_3, w_4, w_5\}, R)$, wobei $(w_i, w_j) \in R$ genau dann, wenn $j = i + 1$. Die Belegung V sei gegeben durch:

$$V(p) = \{w_2, w_3\}$$
$$V(q) = \{w_1, w_2, w_3, w_4, w_5\}$$
$$V(r) = \emptyset$$

(i) Zeichnen Sie die Kripke-Struktur als Graphen.

(ii) Sind a) bis g) jeweils richtig oder falsch? Geben Sie eine knappe Begründung an.

a) $\mathcal{M}, w_2 \vDash \Box p$

b) $\mathcal{M}, w_1 \vDash \Diamond\Box p$

c) $\mathcal{M}, w_1 \vDash \Diamond\Box p \rightarrow p$

d) $\mathcal{M}, w_2 \nvDash \Diamond(p \wedge \neg r)$

e) $\mathcal{M}, w_3 \vDash q \wedge \Diamond(q \wedge \Diamond q)$

f) $\mathcal{M} \vDash \Box q$

g) $\mathcal{M}, w_5 \vDash \Diamond p$

Übungsaufgabe 94. Sei φ eine beliebige modallogische Formel.
Zeigen Sie: $\Box\varphi \leftrightarrow \neg\Diamond\neg\varphi$ ist eine Tautologie.

Übungsaufgabe 95. Gegeben ist folgender Kripke-Rahmen

$$\mathcal{F} = (\{w_1, w_2, w_3, w_4, w_5\}, R)$$

mit der Relation

$$R = \{(w_1, w_2), (w_2, w_3), (w_3, w_4), (w_4, w_5), (w_4, w_1), (w_3, w_2), (w_5, w_1)\}.$$

Zusätzlich ist folgende Belegung V gegeben:

$$V(p) = \{w_2, w_3\}$$
$$V(q) = \{w_1, w_4, w_5\}$$
$$V(r) = \emptyset$$

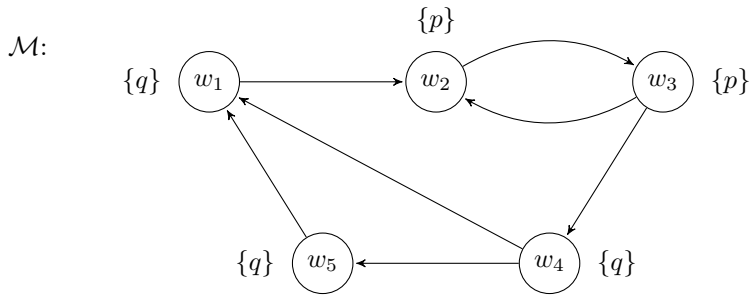
Es sei $\mathcal{M} = ((\mathcal{F}, R), V)$. Sind folgende Behauptungen wahr oder falsch? Geben Sie jeweils eine kurze Begründung an.

(i) $\mathcal{M}, w_3 \vDash \Diamond\Box q$

(ii) $\mathcal{M}, w_1 \vDash \Box(p \wedge \neg r)$

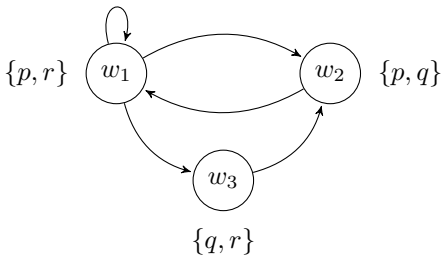
(iii) $\mathcal{M}, w_4 \vDash \Diamond\Box p \to p$

Übungsaufgabe 96. Gegeben ist folgendes Modell $\mathcal{M}$:



Bestimme $[\![\varphi]\!]^{\mathcal{M}}$ zu $\varphi := \Box(p \to \Diamond q)$.

Übungsaufgabe 97. Gegeben ist folgendes Modell $\mathcal{M} = (\{w_1, w_2, w_3\}, R, V)$, mit R und V wie aus folgendem Diagramm ersichtlich.



Was können Sie über die Gültigkeit von $\varphi := \Diamond(q \to \Box(p \to r))$ in diesem Modell sagen? Die Antwort ist zu begründen.

Übungsaufgabe 98. Überprüfen Sie die folgenden modallogischen Formeln auf Erfüllbarkeit, Allgemeingültigkeit und Unerfüllbarkeit. Begründen Sie Ihre Antworten ggf. durch Angabe geeigneter modallogischer Modelle.

(i) $\Phi_a := \Box p \to \Diamond p$

(ii) $\Phi_b := \Box(p \wedge r) \wedge \Box(r \to \neg p) \wedge \Diamond q$

5.4. Charakterisierung von Rahmeneigenschaften

Wie im letzten Beispiel lassen sich mittels modaler Formeln Rahmeneigenschaften charakterisieren. Wir geben einige Beispiele:

Satz 99. Ein Rahmen $\mathcal{F} = (W, R)$ ist *transitiv* genau dann, wenn für alle Formeln φ gilt $\mathcal{F} \vDash \Diamond\Diamond\varphi \to \Diamond\varphi$. transitive Rahmen

Beweis

„⇒“ Sei $\mathcal{F} = (W, R)$ transitiv. Sei weiter $\mathcal{M} = (\mathcal{F}, V)$ mit einer beliebigen Belegung V und $w \in W$. Wir müssen zeigen, dass

$$\mathcal{M}, w \vDash \Diamond\Diamond\varphi \to \Diamond\varphi$$

gilt. Nehmen wir dazu an, dass

$$\mathcal{M}, w \vDash \Diamond\Diamond\varphi$$

gültig ist, so folgt, dass es eine Welt $w' \in W$ mit $(w, w') \in R$ gibt mit

$$\mathcal{M}, w' \vDash \Diamond\varphi.$$

Dies wiederum heisst, dass eine Welt $w'' \in W$ mit $(w', w'') \in R$ existiert mit

$$\mathcal{M}, w'' \vDash \varphi. \tag{5.1}$$

Weil R transitiv ist, gibt es mit $(w, w'), (w', w'') \in R$ dann auch die Kante $(w, w'') \in R$. Mit (5.1) folgt dann

$$\mathcal{M}, w \vDash \Diamond\varphi$$

und damit die Behauptung

$$\mathcal{M}, w \vDash \Diamond\Diamond\varphi \to \Diamond\varphi.$$

„⇐“ Es gelte $\mathcal{F} \vDash \Diamond\Diamond\varphi \to \Diamond\varphi$. Seien weiter w, w', w'' Welten in W mit (w, w'), $(w', w'') \in R$. Wir wählen eine Variable x und eine Belegung V mit $w'' \in V(x)$ und $u \notin V(x)$ für alle Welten $u \in W \setminus \{w''\}$. Dann gilt nach Voraussetzung

$$\mathcal{M}, w \vDash \Diamond\Diamond x \to \Diamond x.$$

Da wegen $\mathcal{M}, w'' \vDash x$ offenbar die Prämisse $\mathcal{M}, w \vDash \Diamond\Diamond x$ gilt, erhalten wir

$$\mathcal{M}, w \vDash \Diamond x.$$

Die einzige Welt, in der x erfüllt ist, ist jedoch w''. Also muss es eine Kante $(w, w'') \in R$ geben und damit ist R transitiv. ■

reflexive Rahmen

Satz 100. Ein Rahmen $\mathcal{F} = (W, R)$ ist *reflexiv* genau dann, wenn für alle Formeln φ gilt $\mathcal{F} \vDash \varphi \to \Diamond\varphi$.

symmetrische Rahmen

Satz 101. Ein Rahmen $\mathcal{F} = (W, R)$ ist *symmetrisch* genau dann, wenn für alle Formeln φ gilt $\mathcal{F} \vDash \varphi \to \Box\Diamond\varphi$.

Übungsaufgabe 102. Beweisen Sie die Sätze 100 und 101.

Modallogische Systeme

Durch Einschränkung der erlaubten Rahmenklasse erhält man so interessante Teilsysteme modaler Logik wie in Tabelle 5.1.

Definition 103. Sei Λ ein modales System.

(i) Ein Rahmen $\mathcal{F}$ heißt Λ-Rahmen (oder Rahmen für Λ) genau dann, wenn $\mathcal{F} \vDash \Lambda$ gilt, d.h. $\mathcal{F} \vDash \varphi$ für alle $\varphi \in \Lambda$.

(ii) Ein Modell heißt Λ-Modell (oder Modell für Λ) genau dann, wenn es auf einem Λ-Rahmen basiert.

Übungsaufgabe 104. Geben Sie für die in der Tabelle genannten Systeme K, K4, ..., S5 jeweils einen Rahmen und ein Modell an.

System	Klasse aller ...
K	Rahmen
K4	transitiven Rahmen
T	reflexiven Rahmen
B	symmetrischen Rahmen
S4	reflexiven und transitiven Rahmen
S5	Äquivalenzrelationen

Tabelle 5.1.: Modale Systeme

5.5. Weitere Themengebiete

5.5.1. Bisimulation

Definition 105. Seien $\mathcal{F} = (W, R)$ ein Rahmen und $\mathcal{M} = (W, R, V)$ ein Modell.

(i) Ein Rahmen $\mathcal{F}' = (W', R')$ ist ein *Teilrahmen* von $\mathcal{F}$ genau dann, wenn $W' \subseteq W$ und $R' = R \cap (W' \times W')$ gelten. — Teilrahmen

(ii) Ein Modell $\mathcal{M}' = (W', R', V)$ ist ein *Teilmodell* von $\mathcal{M}$ genau dann, wenn es auf einem Teilrahmen von $\mathcal{F}$ basiert. — Teilmodell

(iii) Für eine Teilmenge W' von W bezeichnet das *auf W' eingeschränkte Modell* $\mathcal{M}_{\restriction W'}$ dasjenige Teilmodell von $\mathcal{M}$, dessen Menge von Welten die Menge W' ist. — eingeschränktes Modell

Definition 106. Seien $\mathcal{M} = (W, R, V)$ ein Modell und $W' \subseteq W$.

(i) Ein Teilmodell $\mathcal{M}' = (W', R', V)$ von $\mathcal{M}$ heißt *erzeugtes Teilmodell* von $\mathcal{M}$ genau dann, wenn für alle $w \in W'$ und alle $v \in W$ gilt: Wenn wRv, dann $v \in W'$. — erzeugtes Teilmodell

(ii) Das kleinste erzeugte Teilmodell von $\mathcal{M}$, dessen Menge von Welten die Menge W' enthält, heißt das *von W' erzeugte Teilmodell* von $\mathcal{M}$.

(iii) Ein von einer einelementigen Menge erzeugtes Teilmodell heißt *punktgeneriertes Teilmodell*. — punktgeneriertes Teilmodell

Definition 107. Seien $\mathcal{M}_1 = (W_1, R_1, V_1)$ und $\mathcal{M}_2 = (W_2, R_2, V_2)$ zwei Modelle und $f : W_1 \to W_2$ eine Abbildung.

(i) f heißt *Homomorphismus* von $\mathcal{M}_1$ nach $\mathcal{M}_2$ genau dann, wenn folgende Eigenschaften erfüllt sind: — Homomorphismus

a) Für alle $v, w \in W_1$ gilt: Wenn vR_1w, dann $f(v)R_2f(w)$.

b) Für alle $p \in \mathsf{Var}$ und alle $w \in W_1$ gilt: Wenn $w \in V_1(p)$, dann $f(w) \in V_2(p)$.

(ii) f heißt *Isomorphismus* von $\mathcal{M}_1$ nach $\mathcal{M}_2$ genau dann, wenn f bijektiv ist und folgende Eigenschaften erfüllt sind: — Isomorphismus

a) Für alle $v, w \in W_1$ gilt: vR_1w genau dann, wenn $f(v)R_2f(w)$.

b) Für alle $p \in \mathsf{Var}$ und alle $w \in W_1$ gilt: $w \in V_1(p)$ genau dann, wenn $f(w) \in V_2(p)$.

beschränkter Morphismus

(iii) f heißt *beschränkter Morphismus* von $\mathcal{M}_1$ nach $\mathcal{M}_2$ genau dann, wenn folgende Eigenschaften erfüllt sind:

a) Für alle $v, w \in W_1$ gilt: Wenn vR_1w, dann $f(v)R_2f(w)$.

b) Für alle $p \in \mathsf{Var}$ und alle $w \in W_1$ gilt: $w \in V_1(p)$ genau dann, wenn $f(w) \in V_2(p)$.

c) Für alle $v_1 \in W_1$ und $w_2 \in W_2$ mit $f(v_1)R_2w_2$ existiert ein $W_1 \in W_1$ mit $v_1R_1w_1$ und $f(w_1) = w_2$.

Bisimulation

Definition 108. Seien $\mathcal{M} = (W, R, V)$ und $\mathcal{M}' = (W', R', V')$ zwei Modelle. Eine nicht-leere Relation $Z \subseteq W \times W$ heißt *Bisimulation* zwischen $\mathcal{M}$ und $\mathcal{M}'$ (symbolisch: $Z \colon \mathcal{M} \leftrightarroweq \mathcal{M}'$), wenn die folgenden Bedingungen erfüllt sind:

(i) Wenn wZw' gilt, dann erfüllen w und w' die gleichen elementaren Aussagen.

(ii) Wenn wZw' und wRv gelten, dann gibt es ein v' (in $\mathcal{M}'$), sodass vZv' und $w'R'v'$ (Hin-Eigenschaft).

(iii) Wenn wZw' und $w'R'v'$ gelten, dann gibt es ein v (in $\mathcal{M}$), sodass vZv' und wRv (Rück-Eigenschaft).

Wenn Z eine Bisimulation ist, die zwei Welten – w in $\mathcal{M}$ und w' in $\mathcal{M}'$ – verbindet, dann sagen wir, w und w' sind bisimilar. Schreibweise: $Z \colon \mathcal{M}, w \leftrightarroweq \mathcal{M}', w'$ oder kurz: $w \leftrightarroweq w'$ (wenn klar ist, um welche Modelle es sich handelt).

Satz 109. Seien $\mathcal{M}$ und $\mathcal{M}'$ Modelle, zwischen denen es eine Bisimulation gibt, seien w und w' bisimilar und sei φ eine modallogische Formel. Dann gilt:

$$\mathcal{M}, w \vDash \varphi \quad \text{gdw.} \quad \mathcal{M}, w' \vDash \varphi.$$

5.5.2. Multi-Modallogik

Wir können die Modallogik nun erweitern: Statt einem unären Konnektor $\Box$ soll nun eine Menge $\{\Box_i \mid i \in I\}$, $I = \{1, 2, \ldots\}$ von modalen Konnektoren gegeben sein.

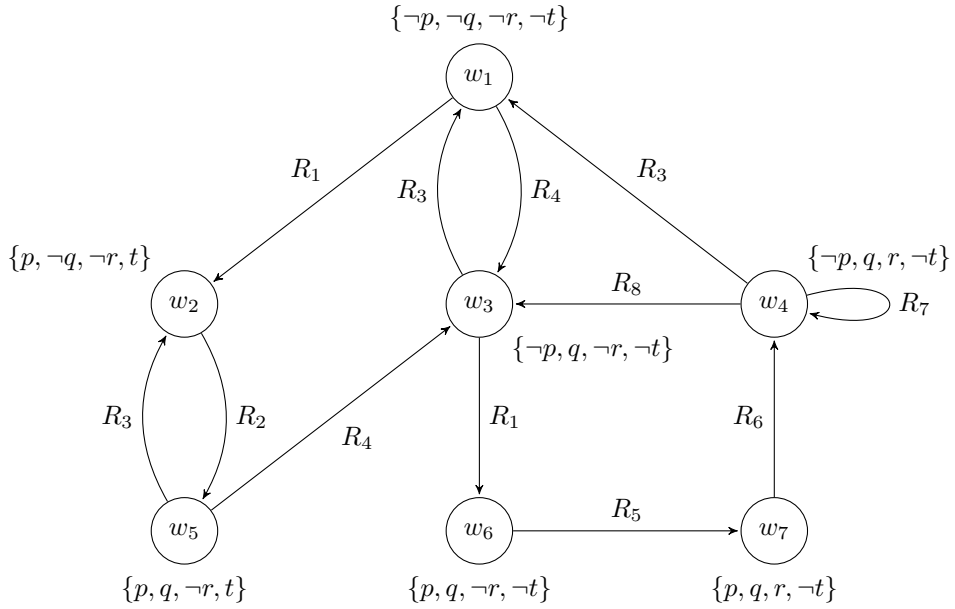
Der Kripke-Rahmen aus Definition 88 wird dann entsprechend erweitert:

$$\mathcal{F} = (W, \{R_i \mid i \in I\})$$

Die Semantik aus Definition 89 weist folgende Änderungen auf:

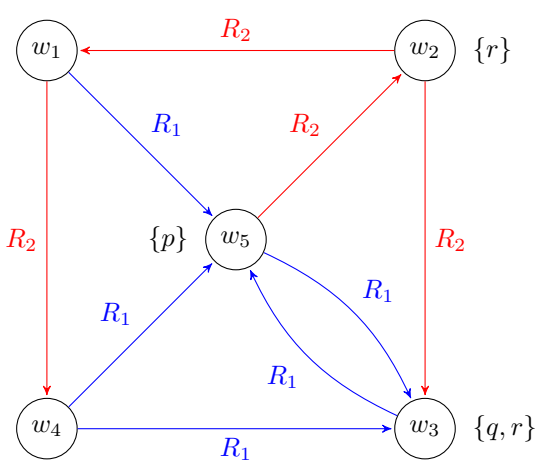
$$\mathcal{M}, w \vDash \Box_i\varphi \text{ falls für alle } v \in W \text{ mit } (w, v) \in R_i \text{ gilt } \mathcal{M}, v \vDash \varphi.$$

Beispiel 110.



Übungsaufgabe 111. Gegeben ist das folgende multi-modallogische Modell $\mathcal{M}$:

$\mathcal{M}$:



(i) Geben Sie $\mathcal{M}$ formal an.

(ii) a) $\varphi_1 := \Box_1(\Diamond_1 q \vee \Diamond_2 r)$, $[\![\varphi_1]\!]^{\mathcal{M}} = ?$

b) $\varphi_2 := \Box_2(r \vee \Box_1 \Diamond_1 \Diamond_2 r)$, $[\![\varphi_2]\!]^{\mathcal{M}} = ?$

(iii) Geben Sie ML-Formeln ψ_1 und ψ_2 an, sodass

a) $[\![\psi_1]\!]^{\mathcal{M}} = \{w_3, w_4\}$

b) $[\![\psi_2]\!]^{\mathcal{M}} = \{w_2, w_5\}$

c) $[\![\psi_3]\!]^{\mathcal{M}} = \{w_1, w_3, w_4\}$

5.5.3. Temporale Logik

Wir betrachten zwei Logiken, die etwas ausdrucksstärker als ML sind: LTL und CTL.

LTL (linear time temporal logic)

$$\varphi ::= p \mid \neg\varphi \mid \varphi \wedge \varphi \mid \mathbf{X}\,\varphi \mid \varphi\,\mathbf{U}\,\varphi$$

CTL (computational tree logic)

$$\varphi ::= p \mid \neg\varphi \mid \varphi \wedge \varphi \mid \mathbf{EX}\,\varphi \mid \mathbf{AX}\,\varphi \mid \mathbf{E}(\varphi\,\mathbf{U}\,\varphi) \mid \mathbf{A}(\varphi\,\mathbf{U}\,\varphi),$$

wobei $p \in \mathsf{Var}$.

In beiden Logiken betrachten wir die Eigenschaften von *Pfaden* in einer gegebenen Kripke-Struktur. Dabei ist ein Pfad in $\mathcal{M}$ eine unendliche Folge von Welten $\pi = w_1, w_2, \ldots$, sodass $(w_i, w_{i+1}) \in R$ für alle i gilt.

Bedeutung der Konnektive:

- **X** „next time“, d.h. „für den nächsten Knoten im Pfad“
- **U** „until“, d.h. „φ gilt solange, bis eine Welt kommt, in der φ' gilt“
- **E** „es gibt einen Pfad“
- **A** „für alle Pfade“

Wir definieren die Semantik dieser Konnektive nun formal, indem wir eine Logik CTL* einführen, die LTL und CTL umfasst. CTL* enthält zwei Arten von Formeln:

(i) Zustandsformeln φ und

(ii) Pfadformeln ψ

CTL* (full computational tree logic)

$$\varphi ::= p \mid \neg\varphi \mid \varphi \wedge \varphi \mid \mathbf{E}\,\varphi \mid \mathbf{A}\,\varphi,$$
$$\psi ::= p \mid \neg\psi \mid \psi \wedge \psi \mid \mathbf{X}\,\psi \mid \psi\,\mathbf{U}\,\psi,$$

wobei $p \in \mathsf{Var}$.

Wir vereinbaren folgende Schreibweise: Wenn $\pi = w_1 w_2 w_3 \ldots$ ist, dann bezeichnen wir mit π^k den Pfad, der mit w_k startet: $\pi^k = w_k w_{k+1} \ldots$

1.	$\mathcal{M}, w \vDash p$	gdw.	$w \in V(p)$
2.	$\mathcal{M}, w \vDash \neg\varphi$	gdw.	$\mathcal{M}, w \nvDash \varphi$
3.	$\mathcal{M}, w \vDash \varphi \wedge \psi$	gdw.	$\mathcal{M}, w \vDash \varphi$ und $\mathcal{M}, w \vDash \psi$
4.	$\mathcal{M}, w \vDash \varphi \vee \psi$	gdw.	$\mathcal{M}, w \vDash \varphi$ oder $\mathcal{M}, w \vDash \psi$
5.	$\mathcal{M}, w \vDash \mathbf{E}\,\psi$	gdw.	es gibt einen Pfad π von w, sodass $\mathcal{M}, w \vDash \psi$
6.	$\mathcal{M}, w \vDash \mathbf{A}\,\psi$	gdw.	für alle von s startenden Pfade π gilt: $\mathcal{M}, w \vDash \psi$
7.	$\mathcal{M}, \pi \vDash \varphi$	gdw.	w ist die erste Welt von π und $\mathcal{M}, w \vDash \varphi$
8.	$\mathcal{M}, \pi \vDash \neg\psi$	gdw.	$\mathcal{M}, \pi \nvDash \psi$
9.	$\mathcal{M}, \pi \vDash \psi_1 \wedge \psi_2$	gdw.	$\mathcal{M}, \pi \vDash \psi_1$ und $\mathcal{M}, \pi \vDash \psi_2$
10.	$\mathcal{M}, \pi \vDash \psi_1 \vee \psi_2$	gdw.	$\mathcal{M}, \pi \vDash \psi_1$ oder $\mathcal{M}, \pi \vDash \psi_2$

11.	$\mathcal{M}, \pi \vDash \mathbf{X}\,\psi$	gdw.	$\mathcal{M}, \pi^1 \vDash \psi$
12.	$\mathcal{M}, \pi \vDash \psi_1\,\mathbf{U}\,\psi_2$	gdw.	es existiert $k \geq 0$, sodass $\mathcal{M}, \pi^k \vDash \psi_2$ und für alle $0 \leq j < k$ gilt: $\mathcal{M}, \pi^j \vDash \psi_1$
13.	$\mathcal{M}, \pi \vDash \mathbf{F}\,\psi$	gdw.	es existiert $k \geq 0$, sodass $\mathcal{M}, \pi^k \vDash \psi$
14.	$\mathcal{M}, \pi \vDash \mathbf{G}\,\psi$	gdw.	für alle $i \geq 0$ gilt: $\mathcal{M}, \pi^i \vDash \psi$

Kapitel 6

Quantifizierte Boole'sche Formeln

Definition 112. Die Menge der quantifizierten Boole'schen Formeln definieren wir induktiv wie folgt: quantifizierte Boole'sche Formel

(i) 0, 1 und jede aussagenlogische Variable p sind quantifizierte Boolesche Formeln.

(ii) Sei φ eine quantifizierte Boole'sche Formel und sei p eine Variable. Dann sind $\forall p\varphi$ und $\exists p\varphi$ quantifizierte Boole'sche Formeln. Sind φ_1 und φ_2 quantifizierte Boole'sche Formeln, dann sind auch $\varphi_1 \wedge \varphi_2$ und $\varphi_1 \vee \varphi_2$ quantifizierte Boole'sche Formeln.

Quantifizierte Boole'sche Formeln sind also durch folgende Grammatik gegeben:

$$\varphi ::= p \mid \neg\varphi \mid \varphi \wedge \varphi \mid \varphi \vee \varphi \mid \exists p\varphi \mid \forall p\varphi,$$

wobei $p \in \mathsf{Var}$.

Die Quantoren $\exists$ und $\forall$ quantifizieren über Wahrheitswerte. Sie ermöglichen lediglich eine kompakte Darstellung von Formeln. Ihre Semantik ist wie folgt gegeben:

$$\exists x\varphi \equiv \varphi|_0 \vee \varphi|_1,$$
$$\forall x\varphi \equiv \varphi|_0 \wedge \varphi|_1.$$

φ heißt *abgeschlossen*, wenn alle Variablen in φ quantifiziert sind. abgeschlossen

$$\mathrm{QBF} := \{\varphi \mid \varphi \text{ ist abgeschlossen und } \varphi \equiv 1\}$$

Es gilt $\mathrm{SAT} \leq_{\mathrm{m}}^{\mathsf{P}} \mathrm{QBF}$, denn: Sei φ eine aussagenlogische Formel über den Variablen $p_1, \ldots, p_n$. Dann gilt: QBF

$$\varphi \in \mathrm{SAT} \Leftrightarrow \exists p_1 \ldots \exists p_n \varphi \in \mathrm{QBF}$$

In der Vorlesung Komplexitätstheorie wird gezeigt:

Satz 113. QBF ist PSPACE-Vollständig.

Jede quantifizierte Boolesche Formel lässt sich in eine aussagenlogische Formel umwandeln.

Beispiel 114.

$$
\begin{aligned}
& \exists x \forall y (x \vee y \vee z) \\
\equiv\ & \exists x ((x \vee 0 \vee z) \wedge (x \vee 1 \vee z)) \\
\equiv\ & ((0 \vee 0 \vee z) \wedge (0 \wedge 1 \wedge z)) \vee ((1 \vee 0 \vee z) \wedge (1 \vee 1 \vee z)) \\
\equiv\ & (z \wedge 1) \vee (1 \wedge 1) \\
\equiv\ & 1
\end{aligned}
$$

Teil II

Prädikatenlogik

Kapitel 7

Mathematische Strukturen und Abbildungen

7.1. Grundbegriffe

Kartesische Produkte. Seien A und B Mengen. Dann ist

$$A \times B := \{(a, b) \mid a \in A, b \in B\},$$
$$A^n := \underbrace{A \times A \times \cdots \times A}_{n \text{ mal}} = \{(a_1, \ldots, a_n) \mid a_i \in A \text{ für } i = 1, \ldots, n\}.$$

Relationen und Funktionen. Eine Relation auf einer Menge A ist eine Teilmenge $R \subseteq A^n$ für ein $n \in \mathbb{N}$. Wir sprechen auch von einer n-stelligen Relation. Die Zahl n heißt *Stelligkeit* von R.

Eine Funktion auf A, geschrieben $f\colon A^n \to A$ für ein $n \in \mathbb{N}$, ordnet jedem Tupel $\bar{a} \in A^n$ ein Element $f(\bar{a}) \in A$ zu. Wir sprechen auch von einer n-stelligen Funktion. Die Zahl n heißt *Stelligkeit* von f. Nullstellige Funktionen sind Konstanten.

Eigenschaften von Funktionen. Eine Funktion $f\colon A \to B$ heißt

- *injektiv*, wenn für alle $a \neq a'$ auch die Funktionswerte $f(a)$ und $f(a')$ verschieden sind.
- *surjektiv*, wenn es für jedes $b \in B$ ein $a \in A$ gibt, sodass $f(a) = b$.
- *bijektiv*, wenn sie sowohl injektiv als auch surjektiv ist.

Mächtigkeit von Mengen. Zwei Mengen A und B heißen *gleichmächtig* (kurz: $|A| = |B|$), wenn es eine bijektive Abbildung von A nach B gibt. A heißt *abzählbar*, wenn es eine surjektive Funktion $f\colon \mathbb{N} \to A$ gibt.

Potenzmenge. Sei A eine Menge. Dann ist $\mathcal{P}(A) := \{B \mid B \subseteq A\}$ die Potenzmenge von A.

7.2. Strukturen

Eine Struktur $\mathfrak{A}$ ist eine nichtleere Menge A zusammen mit einer Menge von Relationen, Funktionen und Konstanten aus A. A heißt Träger, Grundmenge oder Universum.

Universum

Sind $R_1, \ldots, R_n$ die Relationen auf A, $f_1, \ldots, f_m$ die Funktionen auf A und $c_1, \ldots, c_k$ Konstanten aus A, so schreiben wir

$$\mathfrak{A} = (A; R_1, \ldots, R_n; f_1, \ldots, f_m; c_1, \ldots, c_k).$$

Beispiele sind die Struktur der natürlichen Zahlen

$$\mathfrak{N} = (\mathbb{N}; <; +, \cdot; 0, 1),$$

die Struktur der ganzen Zahlen

$$\mathfrak{Z} = (\mathbb{Z}; <; +, \cdot; 0, 1),$$

beliebige Gruppen

$$\mathfrak{G} = (G; \circ; 1)$$

oder Abelsche Gruppen

$$\mathfrak{A} = (G; +; 0).$$

Nach diesen Beispielen definieren wir den Strukturbegriff detaillierter. Zunächst benötigen den Begriff der Signatur, der die Symbole vorgibt, mit denen wir die verwendeten Relationen, Funktionen und Konstanten bezeichnen.

Signatur

Definition 115. Eine *Signatur* σ ist ein Tupel

$$\sigma = ((R_i)_{i \in I_R}; (f_i)_{i \in I_F}; (c_i)_{i \in I_c})$$

mit

- Relationssymbolen R_i, $i \in I_R$ zusammen mit der Angabe ihrer Stelligkeit,
- Funktionssymbolen f_i, $i \in I_F$ zusammen mit der Angabe ihrer Stelligkeit, sowie
- Konstantensymbolen c_i, $i \in I_c$.

Hierbei sind I_R, I_F, I_c Indexmengen.

Wir bemerken, dass eine Signatur σ lediglich Symbole (Zeichen) ohne inhaltliche Bedeutung enthält. Diese erhalten die Symbole erst, indem sie in einer σ-Struktur $\mathfrak{A}$ interpretiert werden:

σ-Struktur

Definition 116. Eine *σ-Struktur*

$$\mathfrak{A} = (A; (R_i^{\mathfrak{A}})_{i \in I_R}; (f_i^{\mathfrak{A}})_{i \in I_F}; (c_i^{\mathfrak{A}})_{i \in I_c})$$

besteht aus

- dem nicht-leeren Universum A,
- Relationen $R_i^{\mathfrak{A}} \subseteq A^{n_i}$ für $i \in I_R$, wobei n_i die Stelligkeit des Relationssymbols R_i ist,
- Funktionen $f_i^{\mathfrak{A}} : A^{n_i} \to A$ für $i \in I_F$, wobei n_i die Stelligkeit des Funktionssymbols f_i ist, und
- Konstanten $c_i^{\mathfrak{A}} \in A$ für $i \in I_c$.

7.3. Beispiele

Mengen Sei $\sigma = \emptyset$. Die $\emptyset$-Struktur mit der Grundmenge A ist einfach die Menge A.

Graphen Die Signatur von Graphen ist $\sigma_{\mathrm{Gr}} = (E)$. E ist ein binäres Relationssymbol. Vorstellung: E gibt die Kantenbeziehung an. Ein Graph ist eine σ_{Gr}-Struktur $\mathfrak{G} = (V, E^{\mathfrak{G}})$ mit Knotenmenge V (der Grundmenge/dem Universum von $\mathfrak{G}$) und einer Relation $E^{\mathfrak{G}} \subseteq V \times V$.

In der Literatur findet man für die Strukturen, die wir als Graphen bezeichnen, oft andere Bezeichnungen (gerichteter Graph, Digraph).

Lineare und partielle Ordnungen Eine partielle Ordnung (auch Halbordnung oder Poset genannt) ist eine $(\leq)$-Struktur $(A; \leq)$, die folgende Bedingung erfüllt:

Reflexivität: Für alle $a \in A$ gilt $a \leq a$.

Antisymmetrie: Wenn $a \leq b$, dann nicht $b \leq a$.

Transitivität: Wenn $a \leq b$ und $b \leq c$, dann gilt auch $a \leq c$.

Eine strikte Halbordnung ist eine $(<)$-Struktur $(A; <)$, die antisymmetrisch und transitiv ist und folgende Bedingung erfüllt:

Irreflexivität: Für kein $a \in A$ gilt $a < a$.

Eine lineare (synonym: totale) Ordnung erfüllt folgende zusätzliche Bedingung:

Vergleichbarkeit: Für alle a, b gilt entweder $a < b$ oder $a = b$ oder $b < a$.

In der Literatur werden Halbordnungen auch verkürzend Ordnungen genannt. Wir folgen diesem Sprachgebrauch jedoch nicht, sondern sprechen nur bei einer strikten (strengen) linearen Halbordnung von einer Ordnung.

Arithmetrische Strukturen: Die Signatur der Arithmetik ist $\sigma_{\mathrm{Ar}} = (+, \cdot; 0, 1)$, die Signatur der geordneten Arithmetik $\sigma_{\mathrm{Ar}}^{<} = (<; +, \cdot; 0, 1)$.

Bekannte arithmetische Strukturen:

- Die Standard-Arithmetik der natürlichen Zahlen, $\mathfrak{N} = (\mathbb{N}; +, \cdot; 0, 1)$. Die geordnete Standard-Arithmetik ist $\mathfrak{N}^{<} = (\mathbb{N}; +, \cdot, <; 0, 1)$.
- Ringe, wie z. B. der Ring $\mathfrak{Z} = (\mathbb{Z}; +, \cdot; 0, 1)$ der ganzen Zahlen.
- Körper, wie z. B. der Körper $\mathfrak{R} = (\mathbb{R}; +, \cdot; 0, 1)$ der reellen Zahlen oder der Körper $\mathfrak{Q} = (\mathbb{Q}; +, \cdot; 0, 1)$ der rationalen Zahlen oder endliche Körper wie $\mathfrak{F} = (\mathbb{F}_2; +, \cdot)$ mit $\mathbb{F}_2 = \{0, 1\}$ und den Verknüpfungen

$+$	0	1
0	0	1
1	1	0

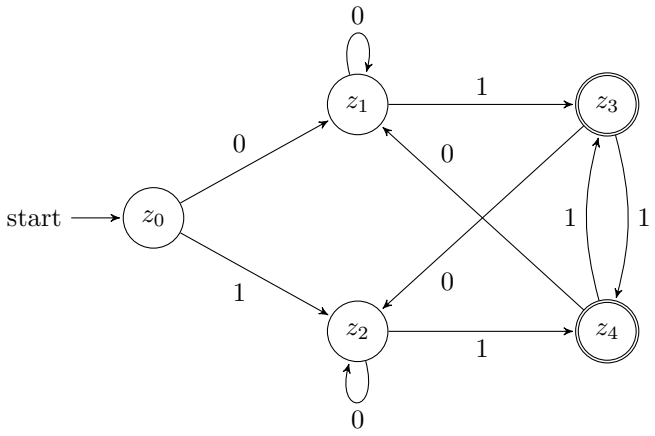
und

$\cdot$	0	1
0	0	0
1	0	1

- Wir können die Standard-Arithmetik $\mathfrak{N}$ durch Hinzunahme eines neuen Elements ω zu neuen arithmetischen Strukturen erweitern, z. B. $\mathfrak{N}^{\omega} = (\mathbb{N} \cup \{\omega\}; <; +, \cdot; 0, 1)$ wobei $n + \omega = \omega + n = n \cdot \omega = \omega \cdot n = \omega$ und $n < \omega$ für alle $n \in \mathbb{N}$.

Boole'sche Algebren Sei A eine beliebige Menge. Die Boole'sche Algebra über A ist $\mathfrak{B}(A) = (\mathcal{P}(A); \cup, \cap, \bar{}; \emptyset)$, wobei $\cup, \cap, \bar{}$ Vereinigung, Durchschnitt und Komplement in A bedeuten.

Endliche Automaten Sei $Z = \{z_0, z_1, z_2, z_3, z_4\}$ eine Zustandsmenge und $\Sigma = \{0, 1\}$ ein Alphabet. Dann lässt sich der durch folgendes Diagramm gegebene endliche Automat $\mathfrak{A}$



als Struktur $\mathfrak{A} = (Z; \delta_0, \delta_1, z_0, z_3, z_4)$ definieren, wobei δ_0 und δ_1 einstellige Funktionen und z_0, z_3, z_4 Konstanten sind.

$\delta_0 \colon Z \to Z$ und $\delta_1 \colon Z \to Z$ sind gegeben durch:

δ_0	z_0	z_1	z_2	z_3	z_4
	z_1	z_1	z_2	z_2	z_1

und

δ_1	z_0	z_1	z_2	z_3	z_4
	z_2	z_3	z_4	z_4	z_3

7.4. Mehrsortige Strukturen

Endliche Automaten können – wie aus der Theoretischen Informatik bekannt – auch als Struktur über zwei Trägermengen, der Zustandsmenge Z und den Eingabezeichen Σ, aufgefasst werden. Man spricht in diesem Zusammenhang auch von einer Struktur mit zwei *Sorten*.

Definition 117. Eine *mehrsortige Signatur* σ ist ein Tupel

$$\sigma = ((S_i)_{i \in I_S}; (R_i)_{i \in I_R}; (f_i)_{i \in I_F}; (c_i)_{i \in I_c})$$

mit

- Sortensymbolen S_i, $i \in I_S$,
- Relationssymbolen R_i, $i \in I_R$, zusammen mit der Angabe ihrer Typdeklarationen $(s_1, s_2, \ldots, s_{n_i})$, wobei $n_i \in \mathbb{N}$ die Stelligkeit des Relationssymbols R_i ist und $s_1, \ldots, s_{n_i} \in I_S$,
- Funktionssymbolen f_i, $i \in I_F$, zusammen mit der Angabe ihrer Typdeklarationen $(s_1, s_2, \ldots, s_{n_i}, s_{n_i+1})$, wobei $n_i \in \mathbb{N}$ die Stelligkeit des Funktionssymbols f_i ist und $s_1, \ldots, s_{n_i}, s_{n_i+1} \in I_S$,

– Konstantensymbolen c_i, $i \in I_c$, zusammen mit ihrem Typ $s \in I_S$.

Hierbei sind I_S, I_R, I_F, I_c Indexmengen.

Definition 118. Eine *mehrsortige σ-Struktur* mehrsortige Struktur

$$\mathfrak{A} = ((A_i)_{i\in I_S}; (R_i^{\mathfrak{A}})_{i\in I_R}; (f_i^{\mathfrak{A}})_{i\in I_F}; (c_i^{\mathfrak{A}})_{i\in I_c})$$

besteht aus

– den Trägermengen A_i für $i \in I_S$, hierbei ist A_i das *Universum der Sorte* s_i,

– Relationen $R_i^{\mathfrak{A}} \subseteq A_{s_1} \times \ldots \times A_{s_{n_i}}$, wobei $(s_1, s_2, \ldots, s_{n_i})$ die Typdeklaration von R_i ist,

– Funktionen $f_i^{\mathfrak{A}} \colon A_{s_1} \times \ldots \times A_{s_{n_i}} \to A_{s_{n_{i+1}}}$, wobei $(s_1, s_2, \ldots, s_{n_i}, s_{n_i+1})$ die Typdeklaration von f_i ist,

– Konstanten $c_i^{\mathfrak{A}} \in A_s$, wobei s der Typ von c_i ist.

Beispiel 119. Man kann endliche Automaten auch als zweisortige Strukturen $\mathfrak{A} = (Z, \Sigma; \delta, z_0, z_{i_1}, \ldots, z_{i_k})$ formalisieren, wobei

– Z die Zustandsmenge ist,

– Σ das Eingabealphabet ist,

– $\delta \colon Z \times \Sigma \to Z$ die Überführungsfunktion ist,

– $z_0 \in Z$ der Startzustand ist und

– $E \subseteq Z$ die Endzustandsmenge ist, $E = \{z_{i_1}, \ldots, z_{i_k}\}$.

Für unseren Beispielautomaten $\mathfrak{A}$ ergibt sich dann folgende Überführungsfunktion δ:

δ	0	1
z_0	z_1	z_2
z_1	z_1	z_3
z_2	z_2	z_4
z_3	z_2	z_4
z_4	z_1	z_3

7.5. Homomorphismen und Isomorphismen

Homomorphismen sind strukturerhaltende Abbildungen.

Definition 120. Sei $\sigma = ((R_i)_{i\in I_R}; (f_i)_{i\in I_F}; (c_i)_{i\in I_c})$ eine Signatur und seien $\mathfrak{A}$ und $\mathfrak{B}$ σ-Strukturen. Eine Abbildung $\pi \colon \mathfrak{A} \to \mathfrak{B}$ heißt *Homomorphismus* von $\mathfrak{A}$ nach $\mathfrak{B}$, wenn die folgenden Bedingungen erfüllt sind: Homomorphismus

(i) Für jedes Relationssymbol R_i, $i \in I_R$, mit Stelligkeit n und alle $a_1, \ldots, a_n \in A$ gilt:

$$\text{wenn } (a_1, \ldots, a_n) \in R^{\mathfrak{A}}, \text{ dann } (\pi a_1, \ldots, \pi a_n) \in R^{\mathfrak{B}}.$$

Dabei ist πa für $a \in A$ eine Kurzschreibweise für $\pi(a)$.

(ii) Für jedes Funktionssymbol R_i, $i \in I_F$, mit Stelligkeit n und alle $a_1, \ldots, a_n \in A$ gilt:

$$\pi f^{\mathfrak{A}}(a_1, \ldots, a_n) = f^{\mathfrak{B}}(\pi a_1, \ldots, \pi a_n).$$

Definition 121. Ein *starker Homomorphismus* von $\mathfrak{A}$ nach $\mathfrak{B}$ ist ein Homomorphismus $\pi\colon \mathfrak{A} \to \mathfrak{B}$, der $(i)'$ erfüllt:

(i)' Für jedes Relationssymbol $R \in R(\sigma)$ und alle $\bar{a} \in A^n$ gilt:

$$\bar{a} \in R^{\mathfrak{A}} \text{ genau dann, wenn } \pi\bar{a} \in R^{\mathfrak{B}}.$$

Einbettung

Definition 122. Eine *Einbettung* von $\mathfrak{A}$ in $\mathfrak{B}$ ist ein injektiver, starker Homomorphismus von $\mathfrak{A}$ nach $\mathfrak{B}$.

Isomorphismus

Definition 123. Ein *Isomorphismus* ist ein bijektiver, starker Homomorphismus. (Anders formuliert: Ein Isomorphismus ist eine surjektive Einbettung.)

Zwei σ-Strukturen $\mathfrak{A}$ und $\mathfrak{B}$ sind isomorph, symbolisch $\mathfrak{A} \cong \mathfrak{B}$, wenn ein Isomorphismus von $\mathfrak{A}$ nach $\mathfrak{B}$ existiert.

Schreibweisen:

$\pi\colon \mathfrak{A} \xrightarrow{\sim} \mathfrak{B}$	π ist Isomorphismus von $\mathfrak{A}$ nach $\mathfrak{B}$
$\pi\colon \mathfrak{A} \xrightarrow{\sim} \mathfrak{A}$	π ist Automorphismus von $\mathfrak{A}$ in sich
$\pi\colon \mathfrak{A} \hookrightarrow \mathfrak{B}$	π ist Einbettung

Beispiel 124. $\pi_1\colon \mathfrak{A} \xrightarrow{\sim} \mathfrak{B}$ ist ein Isomorphismus

für $\mathfrak{A} = (A, f)$ mit $A = \{0, 1\}$ und

f	0	1
0	0	0
1	0	1

sowie $\mathfrak{B} = (B, f)$ mit $B = \{\emptyset, \{0\}\}$ und

f	$\emptyset$	$\{0\}$
$\emptyset$	$\emptyset$	$\emptyset$
$\{0\}$	$\emptyset$	$\{0\}$

,

wobei $\pi_1(0) = \emptyset$, $\pi_1(1) = \{0\}$.

Bemerkung: Für f schreibt man auch $\wedge$ bzw. $\cap$.

Beispiel 125. Folgende Abbildung $\pi_2\colon \mathfrak{A} \xrightarrow{\sim} \mathfrak{A}$ ist ein Automorphismus für

$\mathfrak{A} = (A, f)$ mit $A = \{0, 1\}$ und

f	0	1
0	0	0
1	0	1

,

wobei $\pi_2(0) = 1$, $\pi_2(1) = 0$.

7.6. Kongruenzrelationen und Quotientenstrukturen

Definition 126. Eine binäre Relation $\theta \subseteq A \times A$ nennen wir *Äquivalenzrelation* auf A, falls sie die folgenden drei Bedingungen erfüllt: Äquivalenzrelation

(i) Reflexivität: Für alle $a \in A$ ist $(a, a) \in \theta$.

(ii) Symmetrie: Für alle $a, b \in A$ gilt: Wenn $(a, b) \in \theta$, dann $(b, a) \in \theta$.

(iii) Transitivität: Für alle $a, b, c, \in A$ gilt: Wenn $(a, b) \in \theta$ und $(b, c) \in \theta$, dann $(a, c) \in \theta$.

Für $(a, b) \in \theta$ schreibt man auch $a\theta b$. Ist θ eine Äquivalenzrelation auf A, so nennen wir (A, θ) *Äquivalenzstruktur*.

Beispiel 127. Auf der 5-elementigen Menge $Z = \{z_0, z_1, z_2, z_3, z_4\}$ könnte man (zunächst völlig willkürlich) wie folgt eine Äquivalenzrelation θ definieren:

$$\theta = \big\{(z_0, z_0), (z_1, z_1), (z_1, z_2), (z_2, z_1), (z_2, z_2), (z_3, z_3), (z_3, z_4), (z_4, z_3), (z_4, z_4)\big\}.$$

Dies entspricht folgender Partition von Z:

$$\pi_\theta = \big\{\{z_0\}, \{z_1, z_2\}, \{z_3, z_4\}\big\}$$

Definition 128. Sei θ eine Äquivalenzrelation auf A. Dann heißt

$$[a]_\theta := \{b \mid (a, b) \in \theta\}$$

die Äquivalenzklasse von a bezüglich θ.

Beispiel 129. In unserem Beispiel ergeben sich folgende Äquivalenzklassen:

$$\begin{aligned} [z_0]_\theta &= \{z_0\}, \\ [z_1]_\theta &= \{z_1, z_2\} = [z_2]_\theta, \\ [z_3]_\theta &= \{z_3, z_4\} = [z_4]_\theta. \end{aligned}$$

Definition 130. Sei $\mathfrak{A}$ eine σ-Struktur. Eine *Kongruenzrelation* auf $\mathfrak{A}$ ist eine Äquivalenzrelation θ auf der Grundmenge A von $\mathfrak{A}$, die mit den Relationen und Funktionen von $\mathfrak{A}$ verträglich (synonym: kompatibel) ist, d. h. Kongruenzrelation

(i) Für jede n-stellige Funktion $f^{\mathfrak{A}}$ von $\mathfrak{A}$ und alle Elemente $a_1, \ldots, a_n, b_1, \ldots, b_n \in A$ gilt: Wenn $a_1\theta b_1, \ldots, a_n\theta b_n$, dann $f^{\mathfrak{A}}(a_1, \ldots, a_n)\theta f^{\mathfrak{A}}(b_1, \ldots, b_n)$.

(ii) Für jede n-stellige Relation $R^{\mathfrak{A}}$ von $\mathfrak{A}$ und alle $a_1, \ldots, a_n, b_1, \ldots, b_n \in A$ gilt: Wenn $a_1\theta b_1, \ldots, a_n\theta b_n$, dann gilt: $(a_1, \ldots, a_n) \in R^{\mathfrak{A}}$ genau dann, wenn $(b_1, \ldots, b_n) \in R^{\mathfrak{A}}$.

Die Äquivalenzklassen bezüglich einer Kongruenzrelation heißen auch Kongruenzklassen.

Bemerkung. (i) lässt sich auch so formulieren: Wenn $[a_1]_\theta = [b_1]_\theta, \ldots, [a_n]_\theta = [b_n]_\theta$, dann $[f^{\mathfrak{A}}(a_1, \ldots, a_n)]_\theta = [f^{\mathfrak{A}}(b_1, \ldots, b_n)]_\theta$.

Quotientenstruktur $\mathfrak{A}/\theta$

Definition 131. Sei $\mathfrak{A}$ eine σ-Struktur und sei θ eine Kongruenzrelation auf $\mathfrak{A}$. Die Quotienten-Struktur $\mathfrak{A}/\theta$ (synonym: Faktor-Struktur $\mathfrak{A}/\theta$) ist die σ-Struktur, die die folgenden drei Bedingungen erfüllt:

(i) Die Grundmenge von $\mathfrak{A}/\theta$ ist die Menge $A/\theta := \{[a]_\theta \mid a \in A\}$ der Kongruenzklassen von A.

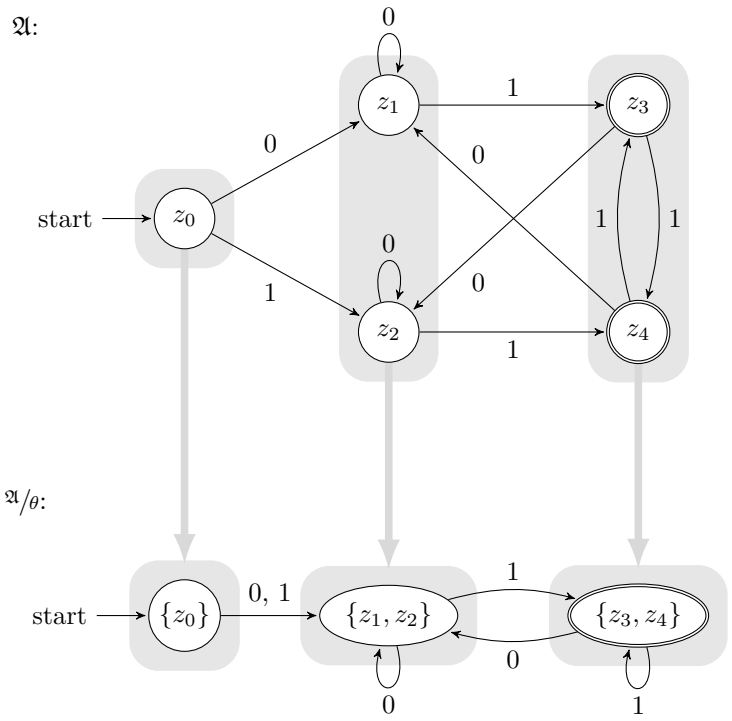
(ii) Für jede n-stellige Funktion $f^{\mathfrak{A}}$ von $\mathfrak{A}$ ist

$$f^{\mathfrak{A}/\theta}([a_0]_\theta, \ldots, [a_{n-1}]_\theta) := [f^{\mathfrak{A}}(a_0, \ldots, a_{n-1})]_\theta.$$

(iii) Für jede n-stellige Relation $R^{\mathfrak{A}}$ von $\mathfrak{A}$ ist

$$([a_0]_\theta, \ldots, [a_{n-1}]_\theta) \in R^{\mathfrak{A}/\theta} \text{ genau dann, wenn } (a_0, \ldots, a_{n-1}) \in R^{\mathfrak{A}}$$

Beispiel 132. Die Quotientenstruktur $\mathfrak{A}/\theta$ zu unserem Automaten $\mathfrak{A}$:



7.7. Kongruenzrelationen und und Homomorphismen

Wir betrachten den Zusammenhang zwischen Quotienten-Strukturen und homomorphen Bildern.

Lemma 133. Sei $\mathfrak{A}$ eine σ-Struktur und θ eine Kongruenzrelationen auf $\mathfrak{A}$. Die Quotienten-Abbildung $\pi : A \to A/\theta$ mit $a \mapsto [a]_\theta$ ist ein surjektiver Homomorphismus von $\mathfrak{A}$ auf $\mathfrak{A}/\theta$.

Satz 134. Sei σ eine funktionale Signatur (d. h. eine Signatur, die keine Relationssymbole enthält) und seien $\mathfrak{A}, \mathfrak{B}$ σ-Algebren. Für jeden Homomorphismus $\pi : \mathfrak{A} \to \mathfrak{B}$ ist die Relation

$$\theta_\pi := \{(a, a') \in A \times A \mid \pi a = \pi a'\}$$

eine Kongruenzrelation über $\mathfrak{A}$.

Satz 135. Für jeden Homomorphismus $\pi : \mathfrak{A} \to \mathfrak{B}$ zwischen σ-Algebren ist $\mathfrak{A}/\theta_\pi \cong \pi(\mathfrak{A})$. Homomorphie-Satz

Kapitel 8

Die Syntax der Prädikatenlogik

Sei eine Signatur σ vorgegeben. Wir definieren zu σ die Sprache der σ-*Formeln* wie folgt. Der zugrundeliegende Zeichenvorrat besteht aus

(i) einer abzählbar unendlichen Menge

$$\mathsf{Var} = \{x_0, x_1, \dots\}$$

von Variablen,

(ii) den Relations-, Funktions- und Konstantensymbolen aus σ,

(iii) den logischen Symbolen $\wedge$, $\neg$, $\forall$, $=$ und

(iv) Klammern (,) und dem Komma.

Wir definieren wir induktiv Terme und Formeln. Wir beginnen mit den Termen: Term

Definition 136. Die Menge der σ-Terme ist induktiv definiert als:

(i) Jede Variable ist eine σ-Term.

(ii) Jede Konstante aus σ ist ein Term.

(iii) Sind $t_1, \dots, t_n$ ein σ-Terme und ist f ein n-stelliges Funktionssymbol aus L, so ist $f(t_1, \dots, t_n)$ ein σ-Term.

Terme gemäß 1 und 2 heißen *Primterme*. Der besseren Lesbarkeit willen, schreiben wir z. B. für $f = +$ statt $+(x, y)$ auch $x + y$.

Induktiv über den Termaufbau können wir Beweise führen oder Eigenschaften definieren. Wir geben ein Beispiel.

Definition 137. Die Menge der Variablen in einem Term ist induktiv definiert als

- $\mathsf{Var}(c) = \emptyset$ für Konstanten c
- $\mathsf{Var}(x_i) = \{x_i\}$ für Variablen x_i
- $\mathsf{Var}(f(t_1, \dots, t_n)) = \mathsf{Var}(t_1) \cup \dots \cup \mathsf{Var}(t_n)$

Als nächstes definieren wir prädikatenlogische Formeln. Formel

Definition 138. Die Menge der σ-Formeln ist induktiv definiert als:

(i) Sind s und t σ-Terme, so ist $s = t$ eine σ-Formel.

(ii) Sind $t_1, \ldots, t_n$ σ-Terme und ist R ein n-stelliges Relationssymbol aus L, so ist $R(t_1, \ldots, t_n)$ eine σ-Formel.

(iii) Sind φ und ψ σ-Formeln, so auch

- $(\varphi \wedge \psi)$,
- $\neg\varphi$ und
- $\forall x\varphi$ mit $x \in \mathsf{Var}$.

Form_σ

Formeln gemäß 1 und 2 heißen *Primformeln* (oder *atomare* Formeln). Die Menge der σ-Formeln notieren wir auch mit Form_σ.

Wir verwenden außerdem die Abkürzungen:

$$\begin{aligned} \varphi \vee \psi &:= \neg(\neg\varphi \wedge \neg\psi) \\ \varphi \to \psi &:= \neg(\varphi \wedge \neg\psi) \\ \varphi \leftrightarrow \psi &:= (\varphi \to \psi) \wedge (\psi \to \varphi) \\ \exists x\varphi &:= \neg\forall x\neg\varphi. \end{aligned}$$

quantorenfreie Formel

Formeln, die $\forall$ und $\exists$ nicht enthalten, heißen *quantorenfreie* Formeln.

Beispiel 139. $\forall x\exists y\, x+y = 0$ ist eine σ-Formel in der Signatur $\sigma = (+; 0)$ der Abelschen Gruppen.

Wie in Definition 137 können wir induktiv die Menge $\mathsf{Var}(\varphi)$ der Variablen einer Formel φ definieren.

gebundene Variable

Eine Variable x kommt in einer Formel φ *gebunden* vor, falls φ das Teilwort $\forall x$ enthält. Die Menge der gebundenen Variablen von φ bezeichnen wir mit $\mathsf{gbd}(\varphi)$.

freie Variable

Die *freien* Variablen in φ werden induktiv definiert als:

- $\mathsf{frei}(\varphi) = \mathsf{Var}(\varphi)$ für Primformeln φ
- $\mathsf{frei}(\varphi \wedge \psi) = \mathsf{frei}(\varphi) \cup \mathsf{frei}(\psi)$ für Formeln φ und ψ
- $\mathsf{frei}(\neg\varphi) = \mathsf{frei}(\varphi)$
- $\mathsf{frei}(\forall x\varphi) = \mathsf{frei}(\varphi) \setminus \{x\}$

Beispiel 140. $\mathsf{frei}(\forall x\exists y\, x + y = 0) = \emptyset$ aber $\mathsf{frei}(x \leq y \wedge \forall x\exists y\, x + y = 0) = \{x, y\}$

Wie das letzte Beispiel zeigt, kann eine Variable in einer Formel also sowohl frei als auch gebunden vorkommen. Dies sollte man jedoch möglichst vermeiden.

Aussage

Formeln φ mit $\mathsf{frei}(\varphi) = \emptyset$ heißen *Aussagen* (oder *Sätze*).

Beispiel 141. $1 + 1 = 0$ und $\forall x\exists y\, x + y = 0$ sind Aussagen.

Die Schreibweise $\varphi(x_1, \ldots, x_n)$ soll bedeuten, dass $\mathsf{frei}(\varphi) \subseteq \{x_1, \ldots, x_n\}$.

Definition 142. Sei t ein Term und x eine Variable. Induktiv über den Term- und Formelaufbau definieren wir die *Substitution* Substitution

(i) für eine Variable y:

$$y[x/t] := \begin{cases} t & \text{falls } x = y \\ y & \text{sonst} \end{cases}$$

(ii) für ein Konstantensymbol c:

$$c[x/t] := c$$

(iii) für ein n-stelliges Funktionssymbol f:

$$f(t_1, \ldots, t_n)[x/t] \;:=\; f(t_1[x/t], \ldots, t_n[x/t])$$

(iv) für Terme t_1, t_2:

$$(t_1 = t_2)[x/t] \;:=\; t_1[x/t] = t_2[x/t]$$

(v) für ein n-stelliges Relationssymbol R:

$$R(t_1, \ldots, t_n)[x/t] \;:=\; R(t_1[x/t], \ldots, t_n[x/t])$$

(vi) für Formeln φ, ψ:

$$(\varphi \wedge \psi)[x/t] := \varphi[x/t] \wedge \psi[x/t]$$

(vii) $(\neg\varphi)[x/t] \;:=\; \neg\,(\varphi[x/t])$

(viii) $(\forall y \varphi)[x/t] \;:=\; \begin{cases} \forall y \varphi & \text{falls } x = y \\ \forall y\,(\varphi[x/t]) & \text{sonst.} \end{cases}$

Kapitel 9

Die Semantik der erststufigen Prädikatenlogik

Ziel des Abschnittes ist es, Termen und Formeln semantische Interpretationen zuzuordnen. Termen werden wir dabei Elemente im Träger und Formeln Wahrheitswerte 0, 1 zuweisen.

Definition 143. Sei σ eine Signatur. Eine *σ-Interpretation* ist ein Paar $\mathfrak{I} = (\mathfrak{A}, \beta)$ bestehend aus einer σ-Struktur $\mathfrak{A}$ und einer *Belegung*

σ-Interpretation
Belegung

$$\beta\colon \mathsf{Var} \to A$$

mit $\beta\colon x \mapsto x^\beta$. Wir bezeichnen $R^\mathfrak{A}$, $f^\mathfrak{A}$, $c^\mathfrak{A}$ und x^β auch mit $R^\mathfrak{I}$, $f^\mathfrak{I}$, $c^\mathfrak{I}$ und $x^\mathfrak{I}$.

Definition 144. Der Wert eines Terms in einer Interpretation $\mathfrak{I} = (\mathfrak{A}, \beta)$ ist induktiv definiert als:

(i) Der Induktionsanfang wurde für Variablen x und Konstanten c bereits mittels $x^\mathfrak{I} = w(x)$ und $c^\mathfrak{I} = c^\mathfrak{A}$ definiert.

(ii) $(f(t_1, \ldots, t_n))^\mathfrak{I} = f^\mathfrak{I}(t_1^\mathfrak{I}, \ldots, t_n^\mathfrak{I})$ für n-stellige Funktionssymbole aus σ.

Bevor wir Formeln Wahrheitswerte zuweisen können, benötigen wir folgende vorbereitende Definition.

Definition 145. Sei $\mathfrak{I} = (\mathfrak{A}, \beta)$ eine Interpretation, sei x eine Variablen und a ein Element aus dem Träger A von $\mathfrak{A}$. Wir definieren w_x^a als

$$\beta_x^a(y) = \begin{cases} a_i & \text{falls } y = x, \\ \beta(y) & \text{sonst.} \end{cases}$$

Ferner sei $\mathfrak{I}_x^a = (\mathfrak{A}, \beta_x^a)$. Ist t ein Term, so schreiben wir auch $\mathfrak{I}_x^t = \mathfrak{I}_x^{t^\mathfrak{I}}$.

In der folgenden Definition ordnen wir Formeln Wahrheitswerte in einer Interpretation zu. Genauer definieren wir eine Relation $\vDash$ zwischen σ-Interpretationen und σ-Formeln. Diese Definition wird auch als induktive Wahrheitsdefinition nach Tarski bezeichnet (benannt nach Alfred Tarski).

Gültigkeit $\models$

Definition 146. Sei eine σ-Interpretation $\mathfrak{I} = (\mathfrak{A}, \beta)$ gegeben. Wir definieren induktiv über den Formelaufbau die Gültigkeit einer σ-Formel φ in der Interpretation $\mathfrak{I}$ durch:

(i) $\mathfrak{I} \models s = t \iff s^{\mathfrak{I}} = t^{\mathfrak{I}}$ für σ-Terme s, t.

(ii) $\mathfrak{I} \models R(t_1, \ldots, t_n) \iff R^{\mathfrak{I}}(t_1^{\mathfrak{I}}, \ldots, t_n^{\mathfrak{I}})$ für n-stellige Relationen R und Terme $t_1, \ldots, t_n$.

(iii) $\mathfrak{I} \models \varphi \wedge \psi \iff \mathfrak{I} \models \varphi$ und $\mathfrak{I} \models \psi$ für σ-Formeln φ und ψ.

(iv) $\mathfrak{I} \models \neg\varphi \iff \mathfrak{I} \not\models \varphi$ (nicht $\mathfrak{I} \models \varphi$) für eine σ-Formel φ.

(v) $\mathfrak{I} \models \forall x\varphi \iff \mathfrak{I}_x^a \models \varphi$ für alle $a \in A$.

φ heißt *gültig* in $\mathfrak{I}$ falls $\mathfrak{I} \models \varphi$.

Beispiel 147. Sei $\mathfrak{N} = (\mathbb{N}; <; +, \cdot\, 0, 1)$ und $\mathfrak{I} = (\mathfrak{N}, \beta)$ mit $\beta(x_i) = i$ für alle Variablen x_i. Dann gilt

$$\begin{aligned} &\mathfrak{I} \models x_1 + x_2 = x_3 \\ &\mathfrak{I} \not\models x_7 < x_4 \\ &\mathfrak{I} \models \forall x_7\,(x_7 = 0 \vee 0 < x_7) \end{aligned}$$

Für unsere Abkürzungen $\vee$, $\rightarrow$, $\leftrightarrow$ und $\exists$ können wir zeigen:

$$\mathfrak{I} \models \varphi \vee \psi \iff \mathfrak{I} \models \varphi \text{ oder } \mathfrak{I} \models \psi. \tag{9.1}$$

$$\mathfrak{I} \models \varphi \rightarrow \psi \iff \text{Wenn } \mathfrak{I} \models \varphi, \text{ dann } \mathfrak{I} \models \psi. \tag{9.2}$$

$$\mathfrak{I} \models \varphi \leftrightarrow \psi \iff \mathfrak{I} \models \varphi \text{ genau dann, wenn } \mathfrak{I} \models \psi. \tag{9.3}$$

$$\mathfrak{I} \models \exists x\varphi \iff \text{Es gibt ein } a \in A \text{ mit } \mathfrak{I}_x^a \models \varphi. \tag{9.4}$$

Exemplarisch beweisen wir (9.4).

Beweis Es ist

$$\exists x\varphi = \neg\forall x\, \neg\varphi.$$

Nach Definition von $\models$ gelten die folgenden Äquivalenzen:

$$\begin{aligned} & \mathfrak{I} \models \neg\forall x\, \neg\varphi \\ \iff\ & \mathfrak{I} \not\models \forall x\, \neg\varphi \\ \iff\ & \text{Es gibt ein } a \in A \text{ mit } \mathfrak{I}_x^a \not\models \neg\varphi. \\ \iff\ & \text{Es gibt ein } a \in A \text{ mit } \mathfrak{I}_x^a \models \varphi. \end{aligned}$$

■

Definition 148. (i) Sei Φ eine Formelmenge. Dann schreiben wir $\mathfrak{I} \models \Phi$, falls $\mathfrak{I} \models \varphi$ für alle $\varphi \in \Phi$.

erfüllbar

(ii) Eine σ-Formel φ heißt *erfüllbar*, falls es eine σ-Interpretation $\mathfrak{I}$ gibt mit $\mathfrak{I} \models \varphi$. Wir sagen dann auch: $\mathfrak{I}$ ist ein *Modell* für φ.

(iii) φ heißt *allgemeingültig* (oder *wahr*, *logisch gültig*, *Tautologie*), falls $\mathfrak{I} \vDash \varphi$ für alle σ-Interpretationen $\mathfrak{I}$. allgemeingültig Tautologie

(iv) Zwei σ-Formeln φ und ψ heißen *logisch äquivalent*, falls für alle σ-Interpretationen $\mathfrak{I}$ gilt:

$$\mathfrak{I} \vDash \varphi \quad \Longleftrightarrow \quad \mathfrak{I} \vDash \psi.$$

Wie in der Aussagenlogik benutzen wir die Notation $\varphi \equiv \psi$.

Beispiel 149. Sei wieder $\sigma = (<; +, \cdot; 0, 1)$. Dann gilt:

(i) $\exists x\ x = 0$ ist allgemeingültig.

(ii) $0 = 1$ ist erfüllbar, aber nicht allgemeingültig.

(iii) $x < 1 + 1$ ist erfüllbar, aber nicht allgemeingültig.

(iv) $\exists z(x < y + z \wedge z = 1)$ und $x < y + 1$ sind logisch äquivalent.

Gilt für alle Belegungen $\beta\colon \mathsf{Var} \to A$

$$(\mathfrak{A}, \beta) \vDash \varphi,$$

so schreiben wir einfach

$$\mathfrak{A} \vDash \varphi.$$

Wir definieren übertragen bereits jetzt den aus der Aussagenlogik bekannten Begriff der "Folgerung" in die Prädikatenlogik. Wir werden ihn in Kap. 13 genauer untersuchen. Folgern

Definition 150. Sei Φ eine Menge von σ-Formeln und φ eine σ-Formel. Dann schreiben wir $\vDash$

$$\Phi \vDash \varphi,$$

falls für alle σ-Interpretationen $\mathfrak{A}$ mit $\mathfrak{A} \vDash \Phi$ auch $\mathfrak{A} \vDash \varphi$ gilt.

Man beachte, dass $\varphi \equiv \psi$, falls $\{\varphi\} \vDash \psi$ und $\{\psi\} \vDash \varphi$.

Beispiel 151. Für alle Strukturen $\mathfrak{A}$ mit $|A| \geq 2$ gilt

$$\mathfrak{A} \vDash \forall x \exists y\ \neg x = y.$$

Beweis Sei $a \in A$ beliebig. Sei weiter β eine beliebige Belegung und $\mathfrak{A} = (\mathfrak{A}, \beta)$. Nach Voraussetzung existiert ein $b \in A$ mit $a \neq b$. Also gilt:

$$(\mathfrak{A}_x^a)_y^b \vDash \neg x = y$$

und damit

$$\mathfrak{A}_x^a \vDash \exists y\ \neg x = y.$$

Weil a beliebig gewählt war, erhalten wir

$$\mathfrak{A} \vDash \forall x \exists y\ \neg x = y.$$

■

Ist $\mathfrak{A}$ eine σ-Interpretation und φ eine σ-Formel, so gilt entweder $\mathfrak{A} \vDash \varphi$ oder $\mathfrak{A} \vDash \neg\varphi$ (tertium non datur). Für Strukturen gilt dies aber nicht. Sei zum Beispiel wie oben $\mathfrak{A}$ eine Struktur mit $|A| \geq 2$. Dann gilt für $\varphi := x = y$ weder $\mathfrak{A} \vDash \varphi$ noch $\mathfrak{A} \vDash \neg\varphi$, weil für $a, b \in A$, $a \neq b$ und eine beliebige Interpretation $\mathfrak{A} = (\mathfrak{A}, \beta)$ gilt:

$$(\mathfrak{A}^a_x)^b_y \nvDash x = y$$

und

$$(\mathfrak{A}^a_x)^a_y \nvDash \neg x = y.$$

Allabschluss

Aus diesem Grund betrachten wir zu einer Formel φ den $\forall$-Abschluss $\varphi^\forall$, definiert mittels

$$\varphi^\forall := \forall x_1 \ldots \forall x_n \varphi,$$

wobei $\mathsf{frei}(\varphi) = \{x_1, \ldots, x_n\}$. Dann gilt:

$$\mathfrak{A} \vDash \varphi \quad \Longleftrightarrow \quad \mathfrak{A} \vDash \varphi^\forall.$$

Deshalb werden äußere $\forall$-Quantoren auch oft weggelassen.

Der folgende Satz formalisiert das sog. *Lokalitätsprinzip* der Prädikatenlogik, das aussagt, dass der Wahrheitswert einer Formel nur von der Bedeutung der vorkommenden Symbole abhängt.

Koinzidenzlemma

Satz 152. Sei V eine Menge von Variablen und φ eine Formel mit $\mathsf{frei}(\varphi) \subseteq V$. Seien weiter $\mathfrak{A}_1 = (\mathfrak{A}_1, \beta_1)$ und $\mathfrak{A}_2 = (\mathfrak{A}_2, \beta_2)$ zwei σ-Interpretationen mit $\beta_1(x) = \beta_2(x)$ für alle $x \in V$ und $g^{\mathfrak{A}_1} = g^{\mathfrak{A}_2}$ für alle logischen Zeichen, die in φ vorkommen. Dann gilt

$$\mathfrak{A}_1 \vDash \varphi \quad \Longleftrightarrow \quad \mathfrak{A}_2 \vDash \varphi.$$

Übungsaufgabe 153. Beweisen Sie Satz 152 durch Induktion über den Formelaufbau.

Nach dem Koinzidenzlemma ist die Gültgkeit von φ in einer Interpretation $\mathfrak{I} = (\mathfrak{A}, \beta)$ nur abhängig von der Belegung der freien Variablen in φ. Damit hängt zum Beispiel die Gültigkeit des $\forall$-Abschlusses einer Formel nicht von der Belegung ab, d.h. für $\mathfrak{I} = (\mathfrak{A}, \beta)$ gilt

$$\mathfrak{I} \vDash \varphi^\forall \quad \Longleftrightarrow \quad \mathfrak{I} \vDash \varphi^\forall.$$

Für $\forall$-Abschlüsse gilt also auch in Strukturen wieder das tertium non datur.

Weiterhin liefert uns das Koinzidenzlemma die Rechtfertigung zum Weglassen überflüssiger Quantoren. Es gilt nämlich:

Lemma 154. Ist φ eine Formel und $x \notin \mathsf{frei}(\varphi)$, so gilt

$$\forall x\varphi \;\equiv\; \varphi \;\equiv\; \exists x\varphi.$$

Beweis Sei $\mathfrak{I} = (\mathfrak{A}, \beta)$. Wir wählen ein beliebiges $a \in A$. Dann gilt mit dem Koinzidenzlemma für $V = \mathsf{frei}(\varphi)$:

$$\mathfrak{A} \vDash \varphi \quad \Longleftrightarrow \quad \mathfrak{A}^a_x \vDash \varphi.$$

Dann gilt also:

$$\begin{array}{ll} & \mathfrak{I} \vDash \forall x\varphi \\ \iff & \mathfrak{I}_x^a \vDash \varphi \\ \overset{\text{für alle } a}{\iff} & \mathfrak{I} \vDash \varphi \\ \iff & \mathfrak{I}_x^a \vDash \varphi \\ \overset{\text{für ein } a}{\iff} & \mathfrak{I} \vDash \exists x\varphi \end{array}$$

■

Übungsaufgabe 155. Sei $\sigma = (f_1, f_2, f_3; c_1, c_2)$ eine Signatur mit dem einstelligen Funktionssymbol f_1 und den zweistelligen Funktionssymbolen f_2 und f_3. Ferner sei

$$\begin{aligned} t_1 &= f_3(x_3, f_1(x_2)), \\ t_2 &= f_2(f_3(f_3(x_3, c_2), f_3(x_3, x_3)), c_1), \\ \varphi &= \exists x_1 \forall x_2 \forall x_3 (t_1 = t_2 \to (x_3 = x_1 \wedge x_2 = c_1)). \end{aligned}$$

Es sei $\mathfrak{A} = (\mathbb{Z}, f_1^{\mathfrak{A}}, f_2^{\mathfrak{A}}, f_3^{\mathfrak{A}}, c_1^{\mathfrak{A}}, c_2^{\mathfrak{A}})$ diejenige σ-Struktur, für die f_1^A definiert ist durch $f_1^{\mathfrak{A}}(a) := a \cdot a$, $f_2^{\mathfrak{A}}$ durch $f_2^{\mathfrak{A}}(a, b) := a + b$ und $f_3^{\mathfrak{A}}$ durch $f_3^{\mathfrak{A}}(a, b) := a \cdot b$ und für die $c_1^{\mathfrak{A}} := 0$ und $c_2^{\mathfrak{A}} := 1$. Dabei sind $\cdot, +, 1, 0$ die aus der Schulmathematik bekannten Operationen bzw. Zahlen

Beweisen oder widerlegen Sie: $\mathfrak{A} \vDash \varphi$.

Übungsaufgabe 156.

(i) Beweisen Sie:

a) $\forall x\varphi \wedge \forall x\psi \equiv \forall x(\varphi \wedge \psi)$

b) $\exists x\varphi \vee \exists x\psi \equiv \exists x(\varphi \vee \psi)$

(ii) Widerlegen Sie:

a) $\forall x\varphi \vee \forall x\psi \equiv \forall x(\varphi \vee \psi)$

b) $\exists x\varphi \wedge \exists x\psi \equiv \exists x(\varphi \wedge \psi)$

Übungsaufgabe 157. Seien $\varphi := \exists x\, \exists y\, \forall z(x = z \vee y = z)$ und $\psi := \forall x\, \exists y\, \exists z(\neg y = z \wedge \neg x = y \wedge \neg x = z)$. Sind die folgenden Formeln erfüllbar? Allgemeingültig? Unerfüllbar? (Mit kurzer Begründung)

a) ψ b) $\varphi \wedge \psi$ c) $\varphi \vee \psi$ d) $\varphi \wedge \neg\psi$

Kapitel 10

Beispiele für prädikatenlogisches Formalisieren

In diesem Abschnitt betrachten wir in einigen Beispielen, wie sich mathematische Sachverhalte in der Prädikatenlogik erster Stufe ausdrücken lassen.

10.1. Graphen

Im ersten Anwendungsbereich betrachten wir gerichtete Graphen. Als Signatur wählen wir hier $\sigma_{\mathrm{Gr}} = (E)$, d.h. die Sprache der Graphen enthält nur ein einziges Relationssymbol E für die Kantenbeziehung. Zugehörige Strukturen $\mathfrak{G} = (V, E^{\mathfrak{G}})$ enthalten dann als Träger eine Menge V von Knoten, und die Kantenbeziehung wird durch die Interpretation $E^{\mathfrak{G}}$ gegeben.

Wir formulieren folgende Sachverhalte über Graphen als σ_{Gr}-Formeln:

(i) Ein Graph $\mathfrak{G} = (V, E^{\mathfrak{G}})$ heißt *ungerichtet*, wenn seine Kantenrelation $E^{\mathfrak{G}}$ symmetrisch ist.

$\mathfrak{G}$ ist ungerichtet:

$$\varphi_{\mathrm{ung}} = \forall x \, \forall y \, E(x, y) \to E(y, x)$$

Dann gilt:

$$\mathfrak{G} \vDash \varphi_{\mathrm{ung}} \quad \text{genau dann, wenn } \mathfrak{G} \text{ ungerichtet ist.}$$

(ii) Ein Graph $\mathfrak{G} = (V, E^{\mathfrak{G}})$ heißt *schleifenfrei*, wenn kein Knoten mit sich selbst durch eine Kante verbunden ist.

$\mathfrak{G}$ ist schleifenfrei:

$$\forall x \, \neg E(x, x).$$

(iii) Zwischen den Knoten x und y existiert ein Weg der Länge 3:

$$\exists z_1 \, \exists z_2 \, (E(x, z_1) \wedge E(z_1, z_2) \wedge E(z_2, y)).$$

(iv) Alle Knoten im Graphen können durch einen Weg der Länge 3 verbunden werden:

$$\forall x\, \forall y\, \exists z_1\, \exists z_2\, (E(x, z_1) \wedge E(z_1, z_2) \wedge E(z_2, y)).$$

(v) Es gibt keinen Kreis der Länge 4:

$$\neg\, \exists x\, \exists z_1\, \exists z_2\, \exists z_3\, (E(x, z_1) \wedge E(z_1, z_2) \wedge E(z_2, z_3) \wedge E(z_3, x)).$$

Ohne Beweis geben wir den folgenden Satz an, der die Grenze der Ausdrucksfähigkeit der Logik erster Stufe markiert:

Satz 158. Es gibt keine σ_{Gr}-Sätze φ, ψ, so dass für alle Graphen $\mathfrak{G} = (V, E^{\mathfrak{G}})$ gilt:

(i) $\mathfrak{G} \models \varphi$ genau dann, wenn $\mathfrak{G}$ zusammenhängend ist.

(ii) $\mathfrak{G} \models \psi$ genau dann, wenn $\mathfrak{G}$ azyklisch (kreisfrei) ist.

Übungsaufgabe 159. Sei $k \geq 2$. Als *k-Clique* eines ungerichteten Graphen $\mathfrak{G} = (V, E^{\mathfrak{G}})$ bezeichnet man k Knoten $\{v_1, \ldots, v_k\} \subseteq V$, zwischen denen sämtliche Kanten vorhanden sind, d.h. $\{(v_i, v_j) \mid 1 \leq i, j \leq k \text{ und } i \neq j\} \subseteq E^G$. Geben Sie eine σ_{Gr}-Formel φ_k an, sodass für jedes σ_{Gr}-Modell $\mathfrak{G}$ gilt: $\mathfrak{G} \models \varphi_k$ genau dann, wenn $\mathfrak{G}$ eine k-Clique besitzt.

Übungsaufgabe 160. (i) Geben Sie eine σ_{Gr}-Formeln φ mit folgender Eigenschaft an: $\mathfrak{G} \models \varphi$ genau dann, wenn $\mathfrak{G}$ mindestens drei Knoten enthält.

(ii) Geben Sie eine σ_{Gr}-Formeln φ mit folgender Eigenschaft an: $\mathfrak{G} \models \varphi$ genau dann, wenn $\mathfrak{G}$ genau drei Knoten enthält.

(iii) Geben Sie eine σ_{Gr}-Formeln φ mit folgender Eigenschaft an: $\mathfrak{G} \models \varphi$ genau dann, wenn Graph $\mathfrak{G}$ enthält höchstens drei Kanten.

(iv) Geben Sie eine σ_{Gr}-Formeln φ mit folgender Eigenschaft an: $\mathfrak{G} \models \varphi$ genau dann, wenn $\mathfrak{G}$ mindestens zwei, aber höchstens vier Knoten enthält.

(v) Geben Sie eine σ_{Gr}-Formeln φ mit folgender Eigenschaft an: $\mathfrak{G} \models \varphi$ genau dann, wenn $\mathfrak{G}$ mindestens eine, aber höchstens drei Kanten enthält.

(vi) Geben Sie eine σ_{Gr}-Formeln φ mit folgender Eigenschaft an: $\mathfrak{G} \models \varphi$ genau dann, wenn Graph $\mathfrak{G}$ symmetrisch ist und die größte Clique in $\mathfrak{G}$ genau drei Knoten hat.

(vii) Geben Sie für jedes k eine eine σ_{Gr}-Formeln φ_k mit folgender Eigenschaft an: $\mathfrak{G} \models \varphi_k$ genau dann, wenn Graph $\mathfrak{G}$ ungerichtet ist und eine Clique der Größe k enthält.

(viii) Geben Sie eine σ_{Gr}-Formeln φ mit folgender Eigenschaft an: $\mathfrak{G} \models \varphi$ genau dann, wenn für die Kanten von $\mathfrak{G}$ gilt $E \subseteq E^{-1}$.

(ix) Geben Sie eine σ_{Gr}-Formeln φ mit folgender Eigenschaft an: $\mathfrak{G} \models \varphi$ genau dann, wenn jeder Knoten in $\mathfrak{G}$ mit jedem anderen Knoten direkt oder über einen Zwischenknoten verbunden ist.

Übungsaufgabe 161. Es sei $\sigma_{\mathrm{Gr}} = (E)$ die Signatur der Graphen. Gegeben sind die σ_{Gr}-Formeln

$$\varphi_1 := \exists x\ \exists y\ \exists z\ \big(x \neq y \wedge y \neq z \wedge x \neq z \wedge E(x,y) \wedge E(y,z) \wedge E(z,x)\big)$$
$$\varphi_2 := \big(\exists x_1\ \exists x_2\ x_1 \neq x_2\big) \to \big(\forall x_3\ \forall x_4\ E(x_3,x_4) \to \exists x_3\ \exists x_4\ E(x_3,x_4)\big)$$
$$\varphi_3 := \exists x\ \neg\forall y\ \big(\neg E(x,y) \vee E(x,y)\big).$$

(i) Was bedeutet φ_1 inhaltlich?

(ii) Sind φ_1, φ_2 bzw. φ_3 allgemeingültig, erfüllbar bzw. unerfüllbar? Geben Sie jeweils eine kurze Begründung an.

Übungsaufgabe 162. Es sei $\sigma_{\mathrm{Gr}} = (E)$ die Signatur der Graphen. Gegeben sind die Formeln

$$\varphi := \exists x_1\ \exists x_2\ \big(\neg x_1 = x_2 \wedge E(x_1,x_2) \wedge E(x_2,x_1)\big)$$

und

$$\psi := \exists x_1\ \exists x_2\ \exists x_3\ \exists x_4\ \big(E(x_1,x_2) \wedge E(x_2,x_3) \wedge E(x_3,x_4) \wedge E(x_4,x_1)\big).$$

Sind die folgenden Formeln allgemeingültig, erfüllbar bzw. unerfüllbar? Geben Sie eine kurze Begründung an.

(i) $\varphi \to \psi$

(ii) $\varphi \wedge \neg\psi$

(iii) $\neg\psi$

(iv) $\varphi \vee \psi$

10.2. Arithmetik

Als zweites Anwendungsfeld betrachten wir die Arithmetik in der Sprache $\sigma_{\mathrm{Ar}}^{<} = (<; +, \cdot; 0, 1)$ mit Relationssymbol $<$ für eine Ordnung, Funktionssymbolen $+$ und $\cdot$ für die Addition bzw. Multiplikation sowie Konstanten 0 und 1. Das Standardmodell ist hier die Struktur $\mathbb{N}$ der natürlichen Zahlen mit der üblichen Interpretation von $<, +, \cdot, 0, 1$. In $\sigma_{\mathrm{Ar}}^{<}$ formulieren wir folgende Sachverhalte:

(i) x teilt y:
$$\exists z\ x \cdot z = y.$$

(ii) Das Minuszeichen $-$ haben wir nicht in der Sprache $\sigma_{\mathrm{Ar}}^{<}$. Wir können es aber definieren, und zwar wird die Gleichung $x - y = z$ beschrieben durch die σ_{Ar}-Formel:
$$z + y = x.$$

(iii) Ebenso wird die Gleichung $x \equiv y \mod z$ beschrieben durch
$$\exists w\big((x + w = y \vee y + w = x) \wedge \exists v\ z \cdot v = w\big).$$

(iv) x ist Primzahl:

$$\varphi_{\text{prim}}(x) \;:=\; \neg x = 1 \wedge \forall y\,(\exists z\, y \cdot z = x \to y = 1 \vee y = x)$$

(v) Es gibt unendlich viele Primzahlen:

$$\forall x\, \exists y\, \varphi_<(x, y) \wedge \varphi_{\text{prim}}(y)$$

Übungsaufgabe 163. Sei $\sigma = (\leq)$ eine Signatur mit einem binären Relationssymbol $\leq$. Gegeben sind die σ-Formeln

$$\begin{aligned}\varphi_1 &:= \forall x\ \forall y\ (x \leq y \vee \neg(x \leq y))\\ \varphi_2 &:= \forall x\ \exists y\ x \leq y \wedge \forall y\ \exists x\ x \leq y\\ \varphi_3 &:= \exists x\ \forall y\ x \leq y.\end{aligned}$$

Es sei weiter $\theta_i = \bigwedge_{j=1}^{i} \varphi_j$ für $i = 1, 2, 3$.

(i) Sind θ_1, θ_2 bzw. θ_3 allgemeingültig, erfüllbar bzw. unerfüllbar? Geben Sie jeweils eine kurze Begründung an.

(ii) Wir betrachten nun nur Strukturen, die partielle Ordnungen sind. Welche der Formeln θ_1, θ_2 und θ_3 gilt (a) in allen partiellen Ordnungen, (b) in mindestens einer partiellen Ordnung, (c) in keiner partiellen Ordnung?

10.3. Zusammenhang zwischen Modallogik und Prädikatenlogik

Wir können die Modallogik ML wie folgt in die Prädikatenlogik übersetzen (man spricht auch von *Einbettung* von ML in die Prädikatenlogik):

Wir halten zunächst eine *endliche* Menge Var von vorkommenden modallogischen Variablen fest. Die prädikatenlogische Signatur enthält eine binäre Relation R, die die Kantenbeziehung zwischen Welten ausdrückt, sowie für alle $p \in \mathsf{Var}$ ein einstelliges Prädikat P_p:

$$\sigma_{\mathsf{ML}} = (R, (P_p)_{p \in \mathsf{Var}}).$$

Ein Modell $\mathcal{M} = (W, R, V)$ der Modallogik entspricht dann der σ_{ML}-Struktur $\mathfrak{M} = (W; R^{\mathfrak{M}}, (P_p^{\mathfrak{M}})_{p \in \mathsf{Var}})$ (wir lassen der Lesbarkeit halber die Indices $^{\mathfrak{M}}$ im Folgenden weg). Die Elemente des Universums von $\mathfrak{M}$ sind also die Welten von $\mathcal{M}$.

Für jede Formel $\varphi \in \mathsf{ML}$ definieren wir eine σ_{ML}-Formel $\varphi^\circ(x)$ so, dass

$$\mathcal{M}, w \vDash \varphi \quad\Longleftrightarrow\quad \mathfrak{M} \vDash \varphi^\circ(w). \tag{10.1}$$

Dies geschieht mit Hilfe folgender Regeln:

(1) $a^\circ := P_a(x)$

(2) $(\varphi \wedge \psi)^\circ := \varphi^\circ \wedge \psi^\circ$

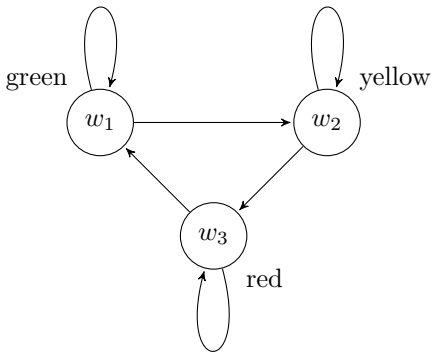
(3) $(\neg\varphi)^\circ := \neg\varphi^\circ$

(4) $(\Box\varphi)^\circ := \forall y\Big(R(x,y) \to \forall x\big(x = y \to \varphi^\circ(x)\big)\Big)$

Man beachte: φ° ist eine σ_{ML}-Formel, in der nur zwei Variablen vorkommen.

Übungsaufgabe 164. Beweisen Sie (10.1).

Beispiel 165. Wir betrachten folgendes Modell $\mathcal{M} = (W, R, V)$ der *Amerikanischen Ampel*.



Als Beispiel betrachten wir die Formel $\hat{\phi} := \Diamond\text{yellow}$. Nach Konvention ist ϕ eine Abkürzung für $\phi := \neg\Box\neg\text{yellow}$. Wir übersetzen nun ϕ nach obiger Vorschrift in eine σ_{ML}-Formel ϕ°:

$$\begin{aligned}
\phi^\circ \equiv (\neg\Box\neg\text{yellow})^\circ &\overset{(3)}{\equiv} \neg(\Box\neg\text{yellow})^\circ \\
&\overset{(4)}{\equiv} \neg\forall y\big(R(x,y) \to \forall x(x = y \to (\neg\text{yellow})^\circ)\big) \\
&\overset{(3)}{\equiv} \neg\forall y\big(R(x,y) \to \forall x(x = y \to \neg(\text{yellow})^\circ)\big) \\
&\overset{(1)}{\equiv} \neg\forall y\big(R(x,y) \to \forall x(x = y \to \neg P_{\text{yellow}}(x))\big)
\end{aligned}$$

Es gilt zum Beispiel $\mathcal{M}, w_1 \vDash \Diamond\text{yellow}$, also $\mathcal{M}, w_1 \vDash \neg\Box\neg\text{yellow}$.

In der zugehörigen σ_{ML}-Struktur $\mathfrak{M}$ gilt dementsprechend:

$$\begin{aligned}
&\mathfrak{M} \vDash \neg\forall y\big(R(x,y) \to \forall x(x = y \to \neg P_{\text{yellow}}(x))\big)(w_1) \\
&\mathfrak{M} \vDash \neg\forall y\big(R(w_1,y) \to \forall x(x = y \to \neg P_{\text{yellow}}(x))\big)
\end{aligned}$$

Kapitel 11

Axiomensysteme

Für eine Menge von σ-Sätzen Φ sei

$$\mathsf{Mod}(\Phi) = \{\mathcal{A} \mid \mathcal{A} \text{ ist } \sigma\text{-Struktur und } \mathcal{A} \vDash \Phi\}.$$

$\mathsf{Mod}(\Phi)$ ist die Klasse aller Modelle von Φ. Mod

Definition 166. Sei $\mathcal{C}$ eine Klasse von σ-Strukturen. Φ *axiomatisiert* $\mathcal{C}$ (Φ ist Axiomensystem für $\mathcal{C}$), falls $\mathcal{C} = \mathsf{Mod}(\Phi)$. Axiomensystem

Zu dieser zentralen Definition betrachten wir einige Beispiele:

Beispiel 167. (i) Sei $\sigma = (\circ, ^{-1}, e)$. Die Menge Φ_{Gruppe} enthält die folgenden drei Aussagen: Gruppe

$$\begin{aligned} \forall x\, \forall y\, \forall z \quad & (x \circ y) \circ z = x \circ (y \circ z) \\ \forall x \quad & x \circ e = x \\ \forall x \quad & x \circ x^{-1} = e \end{aligned}$$

Dann axiomatisiert Φ_{Gruppe} die Klasse aller Gruppen, d.h.

$$\mathsf{Mod}(\Phi_{\text{Gruppe}}) = \{G \mid G \text{ ist Gruppe}\}.$$

(ii) Die Klasse der endlichen Gruppen ist nicht axiomatisierbar. (Beweis: siehe Abschnitt 13.2.)

Beispiel 168. Wir betrachten die Signatur $\sigma^{<}_{\text{Ar}}$ der Arithmetik.

Die *minimale Arithmetik* Q ist durch folgendes Axiomensystem gegeben: minimale Arithmetik

(Q1) $\forall x\, \neg 0 = x + 1$

(Q2) $\forall x \forall y\, x + 1 = y + 1 \to x = y$

(Q3) $\forall x\, x + 0 = x$

(Q4) $\forall x \forall y\, x + y + 1 = (x + y) + 1$

(Q5) $\forall x\, x \cdot 0 = 0$

(Q6) $\forall x \forall y\, x \cdot (y + 1) = (x \cdot y) + x$

(Q7) $\forall x\, \neg x < 0$

(Q8) $\forall x \forall y\, x < y + 1 \leftrightarrow (x < y \vee x = y)$

(Q9) $\forall x \forall y\, x < y \vee x = y \vee y < x$

Sie formalisiert einfache Eigenschaften der üblichen Relationen und Funktionen auf den natürlichen Zahlen.

Peano-Arithematik

Die *Peano-Arithmetik* PA enthält zusätzlich folgendes Axiomenschema für jede Formel $\varphi(x, y_1, \ldots, y_n) \in \mathsf{Form}_{\sigma_{\mathrm{Ar}}^{<}}$, das das Prinzip der natürlichen Induktion formalisiert:

$$\begin{aligned} \forall y_1 \ldots \forall y_n\ & \big(\varphi(0, y_1, \ldots, y_k) \wedge \\ & \quad \forall x\, (\varphi(x, y_1, \ldots, y_k) \to \varphi(x + 1, y_1, \ldots, y_k)) \\ & \big) \\ & \to \forall x\, \varphi(x, y_1, \ldots, y_k) \end{aligned}$$

PA hat neben $\mathfrak{N}$ weitere Modelle, die sog. *Nichtstandardmodelle der Arithmetik*.

Tatsächlich gilt: Die natürlichen Zahlen sind nicht axiomatisierbar, d.h. es gibt keine Menge von Formeln $\Phi_{\mathfrak{N}}$ mit $\mathsf{Mod}(\Phi_{\mathfrak{N}}) = \{\mathfrak{N}\}$.

Peano-Axiomensystem

Aber: Die natürlichen Zahlen sind in der sog. Prädikatenllogik der zweiten Stufe formalisierbar durch das *Peano-Axiomensystem*, das zusätzlich zu Q folgende Formel der *Prädikatenlogik der zweiten Stufe* enthält:

$$\textbf{(P)} \quad \forall X \big(X(0) \wedge \forall x\, (X(x) \to X(x + 1))\big) \to \forall x\, X(x)$$

Hierbei ist X eine Variable zweiter Stufe. Ihre Belegungen sind Relationen/Prädikate über den natürlichen Zahlen.

Beispiel 169. Die Menge der endlichen kreisfreien Graphen ist zwar axiomatisierbar (siehe Übung), aber nicht endlich axiomatisierbar, d.h. es gibt keine endliche Menge Ψ von L_{Gr}-Sätzen mit

$$\mathsf{Mod}(\Psi) = \{\mathcal{G} \mid \mathcal{G} \text{ ist ein kreisfreier Graph}\}$$

(Beweis: siehe Abschnitt 13.2).

Übungsaufgabe 170. Geben Sie eine Formelmenge Φ über σ_{Gr} an, so dass $\mathsf{Mod}(\Phi)$ die Menge der kreisfreien Graphen ist.

Der folgende wichtige Satz drückt aus, dass die erststufige Logik nicht zwischen isomorphen Strukturen zu unterscheiden vermag:

Isomorphie-Satz

Satz 171. Sei φ ein σ-Satz und seien $\mathfrak{A}$ und $\mathfrak{B}$ isomorphe σ-Strukturen. Dann gilt

$$\mathfrak{A} \vDash \varphi \quad \Longleftrightarrow \quad \mathfrak{B} \vDash \varphi.$$

Übungsaufgabe 172. Beweisen Sie Satz 171.

Als Folgerung ergibt sich unmittelbar:

Korollar 173. Sei Φ eine Menge von σ-Sätzen. Dann ist $\mathsf{Mod}(\Phi)$ unter Isomorphie abgeschlossen.

Ist also eine Struktur $\mathfrak{A}$ in $\mathsf{Mod}(\Phi)$ enthalten, so sind auch alle isomorphen Kopien von $\mathfrak{A}$ in $\mathsf{Mod}(\Phi)$. Selbst wenn man isomorphe Strukturen miteinander identifiziert (wie dies in der Mathematik allgemein üblich ist), so sind die natürlichen Zahlen $\mathbb{N}$ trotzdem nicht axiomatisierbar (vgl. das entsprechende Beispiel in obiger Liste).

Übungsaufgabe 174. Es sei $\sigma_{\mathrm{G}} = (\circ; e)$ die Signatur der Gruppen. Eine Gruppe heißt *endlich*, wenn sie nur endlich viele Elemente besitzt. Und eine endliche Gruppe G heißt *zyklisch*, wenn es ein Element $x \in G$ gibt, sodass gilt: $G = \{x^k \mid k \in \mathbb{N} \setminus \{0\}\}$.

Geben Sie eine Formelmenge Φ über σ_{G} an, sodass für jede Gruppe $\mathfrak{G} = (G; \circ; e)$ gilt: $\mathfrak{G}$ ist endlich und zyklisch genau dann, wenn es eine Formel $\varphi \in \Phi$ mit $\mathfrak{G} \vDash \varphi$ gibt.

Übungsaufgabe 175. Wir betrachten die Sprache $\sigma_{\mathrm{Gr}} = (E)$ der Graphen. Gegeben seien die Formel

$$\psi_1(x, y) := E(x, y)$$

und für alle $k \geq 2$ die Formeln

$$\psi_k(x, y) := \exists x_1 \ldots \exists x_{k-1} \big(E(x, x_1) \wedge E(x_1, x_2) \wedge \ldots \wedge E(x_{k-2}, x_{k-1}) \wedge E(x_{k-1}, y) \big)$$

und

$$\varphi_k := \exists x \exists y \big(\psi_k(x, y) \wedge \neg\psi_1(x, y) \wedge \neg\psi_2(x, y) \wedge \ldots \wedge \neg\psi_{k-1}(x, y) \big)$$

sowie die Formelmenge $\Phi = \{\varphi_k \mid k \geq 2\}$.

(i) Welche Eigenschaft von Graphen beschreiben jeweils die Formeln ψ_k und φ_k? Was beschreibt die Formelmenge Φ?

(ii) Zeigen Sie, dass Φ erfüllbar ist!

(iii) Gibt es ein endliches Modell für Φ?

Kapitel 12

Wichtige Äquivalenzen und Normalformen

In diesem Abschnitt betrachten wir wichtige logische Äquivalenzen und Umformungsregeln für prädikatenlogische Formeln. Zunächst kann man sich relativ leicht davon überzeugen, dass sich alle aussagenlogischen Äquivalenzen direkt in die Prädikatenlogik übertragen. So gilt zum Beispiel das de Morgansche Gesetz für beliebige σ-Formeln φ und ψ:

$$\neg(\varphi \wedge \psi) \equiv \neg\varphi \vee \neg\psi.$$

Wir brauchen aber auch Äquivalenzen, die mit Quantoren umgehen können. Die wichtigsten Regeln zu Quantoren sind in dem nächsten Satz zusammengestellt.

Satz 176. Sei σ eine beliebige Signatur und seien φ und ψ σ-Formeln. Dann gilt:

(i) $\neg\forall x\,\varphi \;\equiv\; \exists x\,\neg\varphi$
$\neg\exists x\,\varphi \;\equiv\; \forall x\,\neg\varphi$

(ii) Falls $x \notin \mathsf{frei}(\psi)$, so gilt: $(\forall x\,\varphi) \wedge \psi \;\equiv\; \forall x\,(\varphi \wedge \psi)$

$(\forall x\,\varphi) \vee \psi \;\equiv\; \forall x\,(\varphi \vee \psi)$
$(\exists x\,\varphi) \wedge \psi \;\equiv\; \exists x\,(\varphi \wedge \psi)$
$(\exists x\,\varphi) \vee \psi \;\equiv\; \exists x\,(\varphi \vee \psi)$

(iii) $\forall x\,\varphi \wedge \forall x\,\psi \;\equiv\; \forall x\,(\varphi \wedge \psi)$
$\exists x\,\varphi \vee \exists x\,\psi \;\equiv\; \exists x\,(\varphi \vee \psi)$

(iv) $\forall x\,\forall y\,\varphi \;\equiv\; \forall y\,\forall x\,\varphi$
$\exists x\,\exists y\,\varphi \;\equiv\; \exists y\,\exists x\,\varphi$

Beweis Als Beispiel beweisen wir die erste Äquivalenz in Punkt 2. Sei

$$\mathfrak{I} \vDash (\forall x\,\varphi) \wedge \psi$$

wobei $x \notin \mathsf{frei}(\psi)$. Nach Definition ist dies äquivalent zu

$$\mathfrak{I} \vDash \forall x\varphi \quad \text{und} \quad \mathfrak{I} \vDash \psi.$$

Wiederum nach Definition ist der erste Teil äquivalent zu

$$\mathfrak{I}_x^a \vDash \varphi \quad \text{für alle } a \in A \tag{12.1}$$

und ebenso der zweite wegen des Koinzidenzlemmas (hier brauchen wir $x \notin \mathsf{frei}(\psi)$) zu

$$\mathfrak{I}_x^a \vDash \psi \quad \text{für alle } a \in A. \tag{12.2}$$

Dann ist (12.1) und (12.2) äquivalent zu

$$\text{für alle } a \in A\text{:} \quad \mathfrak{I}_x^a \vDash \varphi \text{ und } \mathfrak{I}_x^a \vDash \psi.$$

Nach Definition ist das äquivalent zu

$$\text{für alle } a \in A\text{:} \quad \mathfrak{I}_x^a \vDash \varphi \wedge \psi$$

und dies wiederum zu

$$\mathfrak{I} \vDash \forall x\,(\varphi \wedge \psi).$$

■

Bei den Äquivalenzen in den Punkten 3 und 4 muss man genau aufpassen. Es gilt nämlich:

$$\begin{aligned} \forall x\,\varphi \vee \forall x\,\psi &\not\equiv \forall x\,(\varphi \vee \psi) \\ \exists x\,\varphi \wedge \exists x\,\psi &\not\equiv \exists x\,(\varphi \wedge \psi) \\ \forall x\,\exists y\,\varphi &\not\equiv \exists y\,\forall x\,\varphi. \end{aligned}$$

Eine weitere nützliche Umformungsregel liefert das folgende intuitiv einsichtige Lemma, das wir hier ohne Beweis angeben:

gebundene Umbenennung

Lemma 177. Sei φ eine Formel und seien x, y zwei verschiedene Variablen, wobei y nicht in φ vorkommt. Dann gilt:

(i) $\forall x\,\varphi \equiv \forall y\,(\varphi[x/y])$

(ii) $\exists x\,\varphi \equiv \exists y\,(\varphi[x/y])$

Wir kommen nun zu dem angekündigten Normalformbegriff für prädikatenlogische Formeln.

pränexe Normalform

Definition 178. Eine σ-Formel ist in *pränexer Normalform*, falls sie die Form

$$Q_1 x_1 \ldots Q_n x_n \varphi$$

hat mit $Q_1, \ldots, Q_n \in \{\forall, \exists\}$, $x_1, \ldots, x_n \in \mathsf{Var}$ und quantorenfreier σ-Formel φ. φ nennt man auch die *Matrix* von $Q_1 x_1 \ldots Q_n x_n \varphi$.

Matrix

Satz 179. Jede Formel ist äquivalent zu einer Formel in pränexer Normalform.

Für den Beweis des Satzes benötigen wir eine Verallgemeinerung zweier Regeln aus Satz 176, die wir in folgendem Lemma festhalten:

Lemma 180. (i) Sei $\psi := Q_1 x_1 \dots Q_n x_n \, \theta$ mit $Q_1, \dots, Q_n \in \{\forall, \exists\}$ und sei

$$\bar{Q}_i = \begin{cases} \forall & \text{falls } Q_i = \exists \\ \exists & \text{falls } Q_i = \forall. \end{cases}$$

Dann gilt $\neg\psi \equiv \bar{Q}_1 x_1 \dots \bar{Q}_n x_n \, \neg\theta$.

(ii) Seien

$$\psi_1 := Q_1 x_1 \dots Q_k x_k \, \theta_1$$
$$\psi_2 := Q'_1 y_1 \dots Q'_l y_l \, \theta_2$$

mit $Q_1, \dots, Q_n, Q'_1 \dots, Q'_l \in \{\forall, \exists\}$ sowie $x_1, \dots, x_k \notin \mathsf{frei}(\psi_2)$ und $y_1, \dots, y_l \notin \mathsf{frei}(\psi_1)$. Dann gilt

$$\psi_1 \wedge \psi_2 \equiv Q_1 x_1 \dots Q_k x_k \, Q'_1 y_1 \dots Q'_l y_l \, \theta_1 \wedge \theta_2.$$

Beweis Der Beweis von Teil 1 erfolgt induktiv über n. Für Teil 2 ist eine simultane Induktion über k und l nötig. ■

Übungsaufgabe 181. Führen Sie den Beweis von Lemma 180 aus.

Nun können wir beweisen, dass jede Formel φ äquivalent zu einer Formel in pränexer Normalform ist.

Beweis von Satz 179. Der Beweis erfolgt per Induktion über den Formelaufbau von φ. Der Induktionsanfang für atomare Formeln ist einfach: diese sind bereits in pränexer Normalform.

Für den Induktionsschritt unterscheiden wir drei Fälle.

(i) Ist $\varphi := \neg\psi$, so existiert nach Induktionsannahme eine Formel ψ' in pränexer Normalform mit $\psi \equiv \psi'$. Nach Teil 1 von Lemma 180 ist $\neg\psi'$ zu einer Formel in pränexer Normalform äquivalent. Daher gilt das auch für φ.

(ii) Sei nun $\varphi := \psi_1 \wedge \psi_2$. Nach Induktionsannahme gibt es Formeln ψ'_1 und ψ'_2 in pränexer Normalform mit $\psi_1 \equiv \psi'_1$ und $\psi_2 \equiv \psi'_2$. Sei etwa

$$\psi'_1 := Q_1 x_1 \dots Q_k x_k \, \theta_1$$
$$\psi'_2 := Q'_1 y_1 \dots Q'_l y_l \, \theta_2$$

Nach Lemma 177 (gebundene Umbenennung) können wir o.B.d.A. annehmen, dass $x_1, \dots, x_k \notin \mathsf{frei}(\psi'_2)$ und $y_1, \dots, x_l \notin \mathsf{frei}(\psi'_1)$. Teil 2 von Lemma 180 liefert dann

$$\varphi \;\equiv\; \psi'_1 \wedge \psi'_2 \;\equiv\; Q_1 x_1 \dots Q_k x_k \, Q'_1 y_1 \dots Q'_l y_l \, \theta_1 \wedge \theta_2.$$

(iii) Für den letzten Fall sei nun $\varphi := \forall\psi$. Nach Induktionsannahme ist ψ zu einer Formel ψ' in pränexer Normalform äquivalent. Damit ist $\varphi \equiv \forall x \, \psi'$ auch in pränexer Normalform.

■

Zum Umformen in pränexe Normalform betrachten wir folgendes Beispiel in der Signatur σ_{Gr}:

$$\begin{aligned}
\varphi(y) &:= \forall x\, \neg(\exists y\, E(x,y) \to \exists x\, E(x,y)) \\
&\equiv \forall x\, \neg(\neg\exists y\, E(x,y) \vee \exists x\, E(x,y)) \\
&\equiv \forall x\, \neg(\forall y\, \neg E(x,y) \vee \exists x\, E(x,y)) \\
&\equiv \forall x\, \neg(\forall z\, \neg E(x,z) \vee \exists w\, E(w,y)) \\
&\equiv \forall x\, \neg\forall z\, \exists w\, (\neg E(x,z) \vee E(w,y)) \\
&\equiv \forall x\, \exists z\, \forall w\, \neg(\neg E(x,z) \vee E(w,y))
\end{aligned}$$

Übungsaufgabe 182. Bestimmen Sie für die folgenden Formeln jeweils äquivalente Formeln in pränexer Normalform.

(i) $\varphi := x + y = (x + z) \cdot x \wedge \neg\exists x\big(x < z \vee \forall z(\neg z = x \vee z \cdot z < x + y)\big)$.

(ii) $\varphi := \forall x \exists y P\big(x, f(y)\big) \wedge \forall y\big(Q(x,y) \vee R(x)\big)$.

(iii) $\varphi := \Big(\forall x \exists y P\Big(x, g\big(y, f(x)\big)\Big) \vee \neg Q(z)\Big) \vee \neg\forall x R(x,y)$.

(iv) $\varphi := \forall z \exists y\Big(P\big(x, g(y), z\big) \vee \neg\forall x Q(x)\Big) \wedge \neg\forall z \exists x \neg R\big(f(x,z), z\big)$.

(v) $\varphi := \Big(\big(\neg\forall x R(x,y) \vee \exists x \neg Q(f(x))\big) \vee \neg\forall\neg(x = y) \wedge \neg\exists y R\big(f(g(x)), g(y)\big)\Big)$.

Übungsaufgabe 183. Es sei $\sigma = (R)$ eine Signatur, die nur ein zweistelliges Relationssymbol R enthält. Gegeben seien die σ-Formeln

$$\varphi_1 := \forall x \forall y \forall z (xRy \wedge xRz \to x = y) \wedge \neg\forall x \exists y\, xRy$$

und

$$\varphi_2 := \forall x \forall y (xRy \vee yRx \vee x = y) \wedge \forall x \forall y (xRy \to \exists z (xRz \wedge zRy)).$$

(i) Schreiben Sie φ_1 und φ_2 in pränexer Normalform.

(ii) Bestimmen Sie für φ_1 und φ_2 allgemeingültig, erfüllbar bzw. unerfüllbar? Begründen Sie Ihre Antwort.

Kapitel 13

Folgern und Schließen

Wir gehen ähnlich wie beim Aussagenkalkül vor und verweisen zunächst auf die dort gemachten allgemeinen Erläuterungen und übertragen einige Definitionen auf die Prädikatenlogik.

Definition 184. Seien σ eine Signatur, $\varphi \in \mathsf{Form}_\sigma$, $\Phi \subseteq \mathsf{Form}_\sigma$.

- φ *folgt aus* Φ (kurz: $\Phi \models \varphi$), falls $\mathfrak{I} \models \Phi \Rightarrow \mathfrak{I} \models \varphi$ für alle $\mathfrak{I}$. $\models$
- $\Phi^{\models} = \{\, \varphi \mid \Phi \models \varphi \,\}$ ($^{\models}$ heißt *Folgerungsoperator*).

Wir halten einige einfache Eigenschaften der definierten Begriffe fest.

Satz 185. (i) φ ist genau dann wahr, wenn $\emptyset \models \varphi$.

(ii) $(\varphi_1 \to \varphi_2)$ ist genau dann wahr, wenn $\{\varphi_1\} \models \varphi_2$.

(iii) $\{\varphi_1, \varphi_2, \ldots, \varphi_m\}^{\models} = \{\varphi_1 \wedge \varphi_2 \wedge \ldots \wedge \varphi_m\}^{\models}$.

(iv) φ ist genau dann wahr, wenn $\forall x \varphi$ wahr ist.

Übungsaufgabe 186. Beweisen Sie Satz 185.

Um das System der Schlussregeln übersichtlich zu halten, schränken wir (wie auch in der Aussagenlogik) den formalen Kalkül ein: Wir verwenden $\to$, $\leftrightarrow$ und $\exists$ nicht. Das ist keine wirkliche Einschränkung wegen, da wir bereits gesehen haben, dass

$$\begin{aligned}(\varphi \to \varphi') &\equiv (\neg\varphi \vee \varphi')\\ (\varphi \leftrightarrow \varphi') &\equiv (\varphi \wedge \varphi') \vee (\neg\varphi \wedge \neg\varphi')\\ \exists x \varphi &\equiv \neg\forall x \neg\varphi\end{aligned}$$

Wir verwenden aber gelegentlich die linke Seite als Abkürzung für die rechte Seite.

13.1. Der Ableitungsbegriff in der Prädikatenlogik der 1. Stufe

Definition 187. Induktive Definition des *Schließens* oder *Ableitens* oder *Beweisens*, d.h. induktive Definition von $\Phi \vdash \varphi$.

Schließen
Ableiten
Beweisen

$\vdash$ **Induktionsanfang:** Falls $\varphi \in \Phi$, so ist $\Phi \vdash \varphi$.

Schlussregel **Induktionsschritt:** Es gelten die folgenden *Schlussregeln*:

- Regel der Fallunterscheidung: Falls $(\Phi \cup \{\varphi'\}) \vdash \varphi$ und $(\Phi \cup \{\neg\varphi'\}) \vdash \varphi$, dann auch $\Phi \vdash \varphi$.
- Regel des indirekten Beweises: Falls $(\Phi \cup \{\neg\varphi\}) \vdash \varphi'$ und $(\Phi \cup \{\neg\varphi\}) \vdash \neg\varphi'$, dann auch $\Phi \vdash \varphi$.
- Abtrennungsregel, modus ponens: Falls $\Phi \vdash (\varphi' \vee \varphi)$ und $\Phi \vdash \neg\varphi'$, dann auch $\Phi \vdash \varphi$. (Man beachte hier, dass $(\varphi' \vee \varphi) \equiv (\neg\varphi' \to \varphi)$; der modus ponens stellt also sozusagen die klassische Deduktionsregel schlechthin dar.)
- Einführung der Alternative: Falls $\Phi \vdash \varphi$, dann auch $\Phi \vdash (\varphi \vee \varphi')$ und $\Phi \vdash (\varphi' \vee \varphi)$ für beliebige $\varphi' \in \mathsf{Form}_{\mathrm{AL}}$.
- Einführung der Konjunktion: Falls $\Phi \vdash \varphi$ und $\Phi \vdash \varphi'$, dann auch $\Phi \vdash (\varphi \wedge \varphi')$.
- Auflösung der Konjunktion: Falls $\Phi \vdash (\varphi \wedge \varphi')$, dann auch $\Phi \vdash \varphi$ und $\Phi \vdash \varphi'$.
- Gleichheitsregel: Falls $\Phi \vdash \varphi(x/t)$, dann auch $\Phi \cup \{t = t'\} \vdash \varphi(x/t')$.
- Einführung von $\forall$: Falls $\Phi \vdash \varphi(x)$, dann auch $\Phi \vdash \forall x\varphi$.
- Auflösung von $\forall$: Falls $\Phi \vdash \forall x\varphi$, dann auch $\Phi \vdash \varphi(x/t)$, falls x nicht frei in t vorkommt.

$\Phi^{\vdash} =_{\mathrm{def}} \{\varphi : \Phi \vdash \varphi\}$ ($^{\vdash}$ heißt *Ableitungsoperator*).

Beispiel 188. Wir wenden uns folgender klassischen Schlussfolgerung zu:

$$\begin{array}{l} \text{Alle Menschen sind sterblich.} \\ \text{Sokrates ist ein Mensch.} \\ \hline \therefore \text{ Sokrates ist sterblich.} \end{array}$$

In der Prädikatenlogik könnte dies wie folgt formalisiert werden:

$$\begin{array}{l} \forall x(\mathsf{H}(x) \to \mathsf{M}(x)) \\ \mathsf{H}(\mathsf{Sokrates}) \\ \hline \therefore \mathsf{M}(\mathsf{Sokrates}) \end{array}$$

Die Menge der Prämissen ist also $\Phi = \{\forall x(\mathsf{H}(x) \to \mathsf{M}(x)), \mathsf{H}(\mathsf{Sokrates})\} \subseteq \Phi^{\vdash}$. Durch einmalige Anwendung der Regel der Auflösung von $\forall$ ergibt sich $(\mathsf{H}(\mathsf{Sokrates}) \to \mathsf{M}(\mathsf{Sokrates})) \in \Phi^{\vdash}$. Man beachte, dass diese Formel eine Abkürzung für $(\neg\mathsf{H}(\mathsf{Sokrates}) \vee \mathsf{M}(\mathsf{Sokrates}))$ darstellt. Eine Anwendung der Regel des modus ponens ergibt also $\mathsf{M}(\mathsf{Sokrates}) \in \Phi^{\vdash}$ und die Gültigkeit der Schlussfolgerung und nachgewiesen.

Ähnlich wie im Aussagenkalkül ergibt sich ein Endlichkeitssatz:

Endlichkeitssatz der Ableitung **Satz 189.** Ist $\varphi \in \Phi^{\vdash}$, so gibt es ein endliches $\Phi_0 \subseteq \Phi$ mit $\varphi \in \Phi_0^{\vdash}$.

Erneut ähnlich wie für den Aussagenkalkül vorgeführt ergeben sich Vollständigkeit und Korrektheit der Schlussregeln. Dieses Resultat geht auf den Logiker Kurt Gödel zurück.

Satz 190. Für jede Formelmenge Φ gilt $\Phi^{\models} = \Phi^{\vdash}$.

Korrektheits- und Vollständigkeitssatz der Prädikatenlogik der 1. Stufe

Damit haben wir sowohl für die Aussagenlogik als auch für die Prädikatenlogik die Äquivalenz der folgenden syntaktischen bzw. semantischen Begriffe gezeigt:

Syntax	**Semantik**
Ableitung/Beweis/Schluss $\Phi \vdash \varphi$	Folgerung/Implikation $\Phi \models \varphi$
Φ ist konsistent	Φ ist erfüllbar (Φ hat ein Modell)
Φ ist inkonsistent	Φ ist unerfüllbar (Φ hat kein Modell)

13.2. Grenzen der Formalisierbarkeit

In diesem Abschnitt werden wir interessante Folgerungen aus dem Vollständigkeitssatz angeben, die gewissermaßen die Grenzen der Formalisierbarkeit und damit der Anwendbarkeit der Prädikatenlogik aufzeigen.

Wir formulieren dazu den kennengelernten Endlichkeitssatz leicht um:

Satz 191.

Endlichkeitssatz, Kompaktheitssatz

(i) (für die Folgerungsbeziehung)
$\Phi \models \varphi$ genau dann, wenn es eine endliche Menge $\Phi_0 \subseteq \Phi$ gibt, sodass $\Phi_0 \models \varphi$.

(ii) (für Erfüllbarkeit)
Φ ist erfüllbar genau dann, wenn jede endliche Menge $\Phi_0 \subseteq \Phi$ erfüllbar ist.

Übungsaufgabe 192. Beweisen Sie Satz 191.

Satz 193. Sei Φ eine Formelmenge, sodass es für beliebige $n \in \mathbb{N}$ eine Interpretation $\mathfrak{I}_n$ gibt, sodass

- $\mathfrak{I}_n \models \Phi$
- $\mathfrak{I}_n$ hat eine Trägermenge der Kardinalität $\geq n$.

Dann gilt: Es gibt eine Interpretation $\mathfrak{I}$ mit unendlicher Trägermenge, sodass $\mathfrak{I} \models \Phi$.

Der Satz sagt also aus, dass jede Formelmenge, die beliebig große endliche Modelle besitzt, auch ein unendliches Modell besitzt.

Beweis Sei

$$\varphi_{\geq n} := \exists x_1 \exists x_2 \exists x_3 \ldots \exists x_n \bigwedge_{1 \leq i < j \leq n} x_i \neq x_j.$$

Jedes Modell $\mathfrak{I}$ mit $\mathfrak{I} \models \varphi_{\geq n}$ besitzt mindestens n Elemente.

Wir setzen

$$\Psi := \Phi \cup \{\varphi_{\geq n} \mid 2 \leq n\}.$$

Sei zunächst $\Psi_0 \subseteq \Psi$ endlich. Dann gibt es $n_0 \in \mathbb{N}$, sodass

$$\Psi_0 \subseteq \Phi \cup \{\varphi_{\geq n} \mid 2 \leq n \leq n_0\}.$$

Also gilt: $\mathfrak{I}_{n_0} \vDash \Psi_0$. Somit ist jede endliche Teilmenge von Ψ und daher nach dem Endlichkeitssatz Ψ erfüllbar. Sei also $\mathfrak{I} \vDash \Psi$. Offensichtlich hat nach Definition von Ψ aber $\mathfrak{I}$ eine unendliche Trägermenge und – da $\Phi \subseteq \Psi$ – es folgt $M \vDash \Phi$. ■

Zur Erinnerung: Für eine Menge Φ von Sätzen (über festgehaltener Signatur σ) ist

$$\mathsf{Mod}(\Phi) = \{A \mid A \text{ ist } \sigma\text{-Struktur und } A \vDash \Phi\}$$

die Klasse aller Modelle von Φ.

Definition 194. Sei $\mathcal{K}$ eine Menge von σ-Strukturen. Wir sagen:

(i) $\mathcal{K}$ ist *axiomatisierbar*, falls es ein Φ gibt, sodass $\mathcal{K} = \mathsf{Mod}(\Phi)$.

(ii) $\mathcal{K}$ ist *endlich axiomatisierbar*, falls es ein endliches Φ gibt, sodass $\mathcal{K} = \mathsf{Mod}(\Phi)$.

Man beachte, dass im zweiten Fall sogar gilt, dass $\mathcal{K} = \mathsf{Mod}(\{\varphi\})$ für eine einzelne Formel φ. Man definiere einfach φ als Konjunktion aller Formeln aus Φ.

Beispiel 195. (i) Die Klasse der Gruppen ist endlich axiomatisierbar. Wir haben oben eine Menge Φ_{Gruppe} angegeben.

(ii) Die Klasse der endlichen Gruppen ist nicht axiomatisierbar: Angenommen, die Klasse der endlichen Gruppen wäre gleich $\mathsf{Mod}(\Phi)$ für eine Formelmenge Φ. Dann hätte Φ beliebig große endliche Modelle, aber kein unendliches Modell: Widerspruch zu Satz 193.

(iii) Analog: Die Klasse der endlichen Körper ist nicht axiomatisierbar.

(iv) Allgemein: Die Klasse der endlichen σ-Strukturen ist nicht axiomatisierbar.

Beispiel 196. Wir betrachten die Signatur σ_{Ar} der Arithmetik.

(i) Die Klasse der Körper (als σ_{Ar}-Strukturen) ist endlich axiomatisierbar. Wir geben eine entsprechende Formelmenge $\Phi_{\text{Kö}}$ an:

$$\begin{gathered}
\forall x\, \forall y\, \forall z\ (x + y) + z = x + (y + z), \\
\forall x\, \forall y\, \forall z\ (x \cdot y) \cdot z = x \cdot (y \cdot z), \\
\forall x\, \forall y\ x + y = y + x, \\
\forall x\, \forall y\ x \cdot y = y \cdot x, \\
\forall x\ x + 0 = x, \\
\forall x\ x \cdot 1 = x, \\
\forall x\, \exists y\ x + y = 0, \\
\forall x\ (\neg\, x = 0 \to \exists y\ x \cdot y = 1), \\
\neg\, 0 = 1, \\
\forall x\, \forall y\, \forall z\ x \cdot (y + z) = (x \cdot y) + (x \cdot z)
\end{gathered}$$

(ii) Sei p eine Primzahl. Ein Körper hat *Charakteristik* p, falls er die Formel

$$\xi_p := \underbrace{1+1+\cdots+1}_{p\text{-mal}} = 0$$

erfüllt. Ein Körper hat Charakteristik 0, falls er für kein $p > 1$ Charakteristik p hat. $\mathbb{Z}_p$, die Menge der ganzen Zahlen modulo p, hat Charakteristik p. $\mathbb{R}$ hat Charakteristik 0.

Die Klasse der Körper der Charakteristik 0 ist axiomatisierbar durch

$$\Phi_{\text{Kö}} \cup \{\neg\xi_p \mid p > 1\}.$$

Die Klasse der Körper der Charakteristik $p > 1$ ist endlich axiomatisierbar.

(iii) Die Klasse der Körper der Charakteristik 0 ist nicht endlich axiomatisierbar. Beweis durch Widerspruch: Sei die Klasse der Körper der Charakteristik p endlich axiomatisierbar. Dann gibt es einen Satz φ, der genau in allen Körpern der Charakteristik p gilt, also

$$\Phi_{\text{Kö}} \cup \{\neg\xi_p \mid p > 1\} \vDash \varphi.$$

Nach dem Endlichkeitssatz gibt es eine Zahl $n_0 \in \mathbb{N}$, sodass

$$\Phi_{\text{Kö}} \cup \{\neg\mathcal{X}_p \mid 1 < p \leq n_0\} \vDash \varphi.$$

Also gilt φ in Körpern der Charakteristik $> n_0$. Widerspruch.

Beispiel 197. (i) Die Klasse der azyklischen Graphen ist axiomatisierbar über σ_{Gr}. Sei dazu

$$\chi_k := \neg\exists x_1\, \exists x_2\, \exists x_3 \ldots \exists x_k\, \big(E(x_1, x_2) \\ \wedge E(x_2, x_3) \wedge \ldots \wedge E(x_{k-1}, x_k) \wedge E(x_k, x_1)\big) \quad \text{für } k \geq 1.$$

Dann ist $\mathsf{Mod}(\{\chi_k\})$ die Menge der Graphen ohne einen Kreis der Länge k und $\mathsf{Mod}(\{\chi_k \mid k \geq 1\})$ die Menge der kreisfreien Graphen.

(ii) Die Menge der azyklischen Graphen ist nicht endlich axiomatisierbar. Beweis durch Widerspruch: Sei $\mathsf{Mod}(\{\varphi\})$ die Menge der kreisfreien Graphen. Dann gilt: $\{\chi_k \mid k \geq 1\} \vDash \varphi$. Nach dem Endlichkeitssatz gibt es $n_0 \in \mathbb{N}$, sodass $\{\chi_k \mid 1 \leq k \leq n_0\} \vDash \varphi$. Also gilt φ in allen Graphen, die keine Kreise der Länge $\leq n_0$ haben. Widerspruch!

Beispiel 198. Sei PA die oben kennengelernte Peano-Arithmetik. Weiterhin sei

$$\zeta_k = \exists x\, \underbrace{1+1+\cdots+1}_{k \text{ mal}} < x$$

und $\mathsf{PA}^\omega = \mathsf{PA} \cup \{\neg\zeta_k \mid k > 1\}$.

Jede endliche Menge $\Phi_0 \subseteq \mathsf{PA}^\omega$ ist erfüllbar, also ist nach dem Kompaktheitssatz auch PA^ω erfüllbar. Sei $\mathfrak{N}^\omega \vDash \mathsf{PA}^\omega$.

Dann gilt: Alle Aussagen, die in $\mathfrak{N}$ gelten, gelten auch in $\mathfrak{N}^\omega$. Aber $\mathfrak{N}$ und $\mathfrak{N}^\omega$ sind nicht isomorph, da letztere eine „unendlich große Zahl“ enthält.

Daher heißt $\mathfrak{N}^\omega$ auch Nichtstandardmodell der Arithmetik.

Kapitel A

Elementare Begriffe und Schreibweisen

A.1. Elementbeziehung und Enthaltenseinsrelation (Inklusion)

- $a \in M \Leftrightarrow_{\text{def}} a$ ist ein Element der Menge M
- $a \notin M \Leftrightarrow_{\text{def}} a$ ist kein Element der Menge M
- $M \subseteq N \Leftrightarrow_{\text{def}}$ aus $a \in M$ folgt $a \in N$ (M ist Teilmenge von N)
- $M \not\subseteq N \Leftrightarrow_{\text{def}}$ es gilt nicht $M \subseteq N$ (M ist keine Teilmenge von N)
- $M \subset N \Leftrightarrow_{\text{def}} M \subset N$ und $M \neq N$ (M ist echte Teilmenge von N)

A.2. Möglichkeiten der Definition spezieller Mengen

- Durch Angabe der Elemente:

  $\{a_1, a_2, \ldots, a_n\}$ ist die Menge, die aus den Elementen $a_1, a_2, \ldots, a_n$ besteht.

  Beispiele:

  - $\{0, 1\}$
  - $\{2, 3, 5, 7, 11, 13, 17, 19, 31, 37\}$
  - $\mathbb{N} =_{\text{def}} \{0, 1, 2, 3, 4, 5, \ldots\}$ (Menge der natürlichen Zahlen)
- Durch eine Eigenschaft E:

  $\{a \mid E(a)\}$ ist die Menge aller Elemente, die die Eigenschaft E besitzen.

  Beispiele:

  - $\{n \mid n \in \mathbb{N}$ und durch 3 teilbar$\}$
  - $\{n \mid n \in \mathbb{N}$ und n Primzahl und $n \leq 50\}$

  - $\emptyset =_{\text{def}} \{a \mid a \neq a\}$ (leere Menge)

- Durch eine induktive Definition:

  Eine Menge D wird aus einer Menge B durch gewisse Operationen erzeugt: Sei dazu gegeben eine Teilmenge $B \subseteq A$ einer Grundmenge A sowie Operationen $O_i : A^{s_i} \to A$ $(i = 1, \ldots, k)$. Dann wird D wie folgt definiert:

(IA)	(Induktionsanfang)	Ist $a \in B$, so ist $a \in D$.
(IS)	(Induktionsschritt)	Für $i = 1, \ldots, k$: Sind $a_1, \ldots, a_{s_i} \in D$, so ist $O_i(a_1, \ldots, a_{s_i}) \in D$.

  Beispiele:

  - D enthält die Zahl 0; ist $n \in D$, so ist auch $n + 1 \in D$ (natürliche Zahlen).
  - D enthält die Zahl 0; ist $n \in D$, so ist auch $n + 2 \in D$ (gerade Zahlen).

A.3. Operationen auf Mengen

$A \cap B$	$=_{\text{def}}$	$\{a \mid a \in A \text{ und } a \in B\}$	Durchschnitt von A und B
$A \cup B$	$=_{\text{def}}$	$\{a \mid a \in A \text{ oder } a \in B\}$	Vereinigung von A und B
$A \setminus B$	$=_{\text{def}}$	$\{a \mid a \in A \text{ und } a \notin B\}$	Differenz von A und B
$\overline{A}$	$=_{\text{def}}$	$M \setminus A$	Komplement von A relativ zu einer festen Grundmenge M
$\mathcal{P}(A)$	$=_{\text{def}}$	$\{B \mid B \subseteq A\}$	Potenzmenge von A

A.4. Gesetze für Mengenoperationen

$A \cap B$	$=$	$B \cap A$	Kommutativgesetz für den Durchschnitt
$A \cup B$	$=$	$B \cup A$	Kommutativgesetz für die Vereinigung
$A \cap (B \cap C)$	$=$	$(A \cap B) \cap C$	Assoziativgesetz für den Durchschnitt
$A \cup (B \cup C)$	$=$	$(A \cup B) \cup C$	Assoziativgesetz für die Vereinigung
$A \cap (B \cup C)$	$=$	$(A \cap B) \cup (A \cap C)$	Distributivgesetz
$A \cup (B \cap C)$	$=$	$(A \cup B) \cap (A \cup C)$	Distributivgesetz
$A \cap A = A \cup A$	$=$	A	Duplizitätsgesetz
$A \cap (B \cup C)$	$=$	A	Absorptionsgesetz
$A \cup (B \cap C)$	$=$	A	Absorptionsgesetz
$\overline{A \cap B}$	$=$	$(\overline{A} \cup \overline{B})$	de-Morgansche Regel
$\overline{A \cup B}$	$=$	$(\overline{A} \cap \overline{B})$	de-Morgansche Regel
$\overline{\overline{A}}$	$=$	A	Gesetz des doppelten Komplements

A.5. Tupel (Vektoren) und Kreuzprodukt

- $(a_1, a_2, \ldots, a_n) =_{\text{def}}$ Folge der Elemente $a_1, a_2, \ldots, a_n$ in dieser festgelegten Reihenfolge (n-Tupel, n-stelliger Vektor)
- $A_1 \times A_2 \times \ldots \times A_n =_{\text{def}} \{(a_1, a_2, \ldots, a_n) \mid a_1 \in A_1, a_2 \in A_2, \ldots, a_n \in A_n\}$ (Kreuzprodukt der Mengen $A_1, A_2, \ldots A_n$)
- $A^n =_{\text{def}} \underbrace{A \times A \times \ldots \times A}_{n\text{-mal}}$ (n-faches Kreuzprodukt der Menge A)

A.6. Anzahl

$|A| =_{\text{def}}$ Anzahl der Elemente der endlichen Menge A

Für endliche Mengen A gilt:

- $|A^n| = |A|^n$
- $|\mathcal{P}(A)| = 2^{|A|}$
- $|A| + |B| = |A \cup B| + |A \cap B|$
- $|A| = |A \setminus B| + |A \cap B|$

A.7. Induktion

Es geht darum, nachzuweisen, dass für jedes Element a einer Menge D eine bestimmte Eigenschaft E gilt. Wir schreiben kurz $E(a)$ für: a besitzt die Eigenschaft E.

1. Variante

Es sei D durch die Operation $O_i : A^{s_i} \to A$ $(i = 1, \ldots, k)$ aus der Menge $B \subseteq A$ erzeugt, d.h.

(IA)	(Induktionsanfang)	Ist $a \in B$, so ist $a \in D$.
(IS)	(Induktionsschritt)	Für $i = 1, \ldots, k$: Sind $a_1, \ldots, a_{s_i} \in D$, so ist $O_i(a_1, \ldots, a_{s_i}) \in D$.

Induktionsprinzip zum Nachweis von $E(a)$ für alle $a \in D$:

Wenn

(IA)	(Induktionsanfang)	$E(a)$ für alle $a \in B$ und
(IS)	(Induktionsschritt)	aus $E(a_1), \ldots, E(a_{s_i})$ folgt $E(O_i(a_1, \ldots, a_{s_i}))$ für $i = 1, \ldots, k$ und $a_1, \ldots, a_{s_i} \in A$,

so gilt $E(a)$ für jedes $a \in D$.

2. Variante

Es gibt eine totale Funktion $\beta : D \to \mathbb{N}$.

Induktionsprinzip zum Nachweis von $E(a)$ für alle $a \in D$:

Wenn

(IA)	(Induktionsanfang)	$E(a)$ für alle $a \in B$ mit $\beta(a) \leq n_0$ und
(IS)	(Induktionsschritt)	für alle $a \in D$ mit $\beta(a) > n_0$ gilt: aus $E(b)$ für alle $b \in D$ mit $\beta(b) < \beta(a)$ folgt $E(a)$,

so gilt $E(a)$ für jedes $a \in D$.

Ist $D = \mathbb{N}$ und β die Identitätsfunktion, so erhält man den häufig verwendeten Spezialfall des *Induktionsprinzips* zum Nachweis von $E(n)$ für alle $a \in \mathbb{N}$:

Gibt es ein $n_0 \in \mathbb{N}$ mit

(IA)	(Induktionsanfang)	$E(n)$ für alle $n \leq n_0$ und
(IS)	(Induktionsschritt)	für alle $n > n_0$ gilt: aus $E(m)$ für alle $m < n$ folgt $E(n)$,

so gilt $E(n)$ für alle $n \in \mathbb{N}$.

A.8. Griechisches Alphabet

A	α	Alpha	I	ι	Iota	P	ρ	Rho
B	β	Beta	K	κ	Kappa	Σ	σ	Sigma
Γ	γ	Gamma	Λ	λ	Lambda	T	τ	Tau
Δ	δ	Delta	M	μ	My	Y	υ	Ypsilon
E	ε	Epsilon	N	ν	Ny	Φ	φ	Phi
Z	ζ	Zeta	Ξ	ξ	Xi	X	χ	Chi
H	η	Eta	O	o	Omikron	Ψ	ψ	Psi
Θ	θ	Theta	Π	π	Pi	Ω	ω	Omega

Literaturverzeichnis

Haupt-Literaturquellen zur Vorlesung:

[1] Ebbinghaus, Heinz-Dieter, Flum, Jörg, Thomas, Wolfgang, *Einführung in die mathematische Logik*, Spektrum Akademischer Verlag, Heidelberg, 5. Auflage, 2007.
Ein klassisches, gutes, aber schwieriges Lehrbuch zur Prädikatenlogik. Deckt den Stoff der Vorlesung zur Prädikatenlogik ab.

[2] van Dalen, Dirk, *Logic and Structure*, Springer Verlag, London, 5. Auflage, 2013.
Ein gutes Lehrbuch, das eingeführte Konzepte ausführlich erklärt und motiviert. Es enthält den größten Teil des Stoffs der Vorlesung (Ausnahmen: Resolution, modale Logik).

[3] Rautenberg, Wolfgang, *Einführung in die mathematische Logik*, Vieweg + Teubner, Wiesbaden, 3. Aufl. Wiesbaden 2008.
Ein klassisches Lehrbuch im deutschsprachigen Raum. Es deckt den größten Teil der Vorlesung ab, aber die verwendete Notation weicht zum Teil stark von unserer ab.

[4] Schöning, Uwe, *Logik für Informatiker*, Spektrum Akademischer Verlag, Heidelberg, 5. Aufl., 2000.
Ein auf Resolution und Logikprogrammierung ausgerichtetes, gut lesbares Lehrbuch.

Klassiker:

Die folgenden ohne Einschränkung zu empfehlenden Bücher sind klassische Lehrbücher zur Logik. Sie gehen über den Stoff der Vorlesung weit hinaus.

[5] Enderton, Herbert B., *A Mathematical Introduction to Logic*, Academic Press, San Diego, 3. Aufl., 2013.

[6] Shoenfield, Joseph R., *Mathematical Logic*, A. K. Peters, Wellesley, Massachusetts, 2001.

[7] Hinman, Peter G., *Fundamentals of Mathematical Logic*, Wellesley, Massachusetts, 2005

Weitere neuere Lehrbücher:

[8] Ben-Ari, Mordechai, *Mathematical Logic for Computer Science*, Springer Verlag, London, 3. Aufl., 2012.

[9] Halbach, Volker, *The Logic Manual*, Oxford University Press, Oxford, 2010.

[10] Smith, Peter, *An Introduction to Formal Logic*, Cambridge University Press, Cambridge, 2003.

[11] Sider, Theodore, *Logic for Philosophy*, Oxford University Press, Oxford, 2010.

[12] Kreuzer, Martin, Kühling, Stefan, *Logik für Informatiker*, Addison-Wesley, München, 2006.

[13] Martin Ziegler, *Mathematische Logik*, Birkhäuser, Basel, 2010.

[14] Jürgen Dassow, *Logik für Informatiker*, Teubner, Stuttgart, 2005.

Index

$\square$, 12
$\models$, 10, 80, 99
$\vdash$, 39, 100
$\models$, 81
Σ_{AL}, 6
σ, 66
σ-Formel, 75
σ-Interpretation, 79
σ-Term, 75
σ_{Ar}, 67
$\sigma^{<}_{\mathrm{Ar}}$, 67
σ_{Gr}, 67
σ_{ML}, 88

Ableiten, 38, 39, 99
Ableitungsbaum, 8
Ableitungskalkül, 39
Ableitungsoperator, 100
äquivalent, 10, 81
    erfüllbarkeits-, 15
    semantisch, 10
Äquivalenzen, 11
Äquivalenzrelation, 71
Äquivalenzstruktur, 71
Allabschluss, 82
allgemeingültig, 16, 81
Alphabet, 6
Antisymmetrie, 67
Arithmetik
    minimale, 91
    Peano, 92
Atom, 5
atomare Formel, 5, 76
Aussage, 76
    elementar, 5
Aussagenlogik, 5
    Semantik, 9
    Syntax, 5
Automat, 68, 69
Axiomensystem, 91

BDD, 20
Belegung, 9, 79
    erfüllende, 10
    für eine Formel, 10
    minimale, 25
Beweis
    Resolutions-, 27
    Widerlegungs-, 27
Beweisen, 38, 39, 99
bijektiv, 65
Bisimulation, 56
Boole'sche Algebra, 68
Buchstabe, 6

Cook, 16

Davis, Martin, 27
Davis-Putnam-Algorithmus, 33
DNF, 12
DPLL-Algorithmus, 34

EBNF, 6
Einbettung, 70
endlicher Automat, 68, 69
Endlichkeitssatz, 17, 101
    der Ableitung, 39, 100
Entscheidungsdiagramm, 20
Erfüllbarkeit, 16
    für Hornformeln, 24
    für modallogische Formeln, 50
erfüllbar, 80
Erfüllbarkeitsäquivalenz, 15
exklusives ODER, 19
Extension, 50

Faktor-Struktur, 72
Folgern, 37, 38, 81

Folgerungsoperator, 38, 99
Form, 5
$\mathsf{Form}_{\mathrm{AL}}$, 6
Formel, 5
    abgeschlossen, 61
    allgemeingültige, 81
    atomare, 5, 76
    aussagenlogische, 5
    erfüllbare, 80
    Prim-, 76
    quantifizierte Boole'sche, 61
    quantorenfreie, 76
Form_{σ}, 76
freie Variable, 76
Funktion, 65
funktional vollständig, 14

Gödel, Kurt, 47, 101
Gültigkeit, 80
Gatter, 19
gebundene Variable, 76
Grundmenge, 66
Gruppe, 91

Halbordnung, 67
Henkin, Leon, 45
Homomorphiesatz, 73
Homomorphismus, 55, 69
    stark, 70
Horn, Alfred, 23
Hornformel, 23
HORNSAT, 24

injektiv, 65
Interpretation, 79
Irreflexivität, 67
Isomorphie-Satz, 92
Isomorphismus, 55, 70

k-Klausel, 15
Kalkül, 39
Karp, Richard, 16
kartesisches Produkt, 65
Klammerregeln, 6
Klausel, 12
    k-Klausel, 15
    Horn-, 23
    leere, 12
KNF, 12
    k-KNF, 15
    3-KNF, 15
Koinzidenzlemma, 82
Kompaktheitssatz, 101
kompatibel, 71
Kongruenzrelation, 71
Konklusion, 37
Konnektor, 5
Konsistenz, 37, 44
Korrektheitssatz, 38, 42
Kripke, Saul, 49
Kripke-Rahmen, 49

lineare Ordnung, 67
Literal, 12
Logik
    Aussagen-, 5
    deontische, 50
    modale, 49
    Modale Systeme, 54
    temporale, 50

möglicherweise, 49
Markierungsalgorithmus, 24
Matrix, 96
McCluskey, Edward J., 21
mehrsortige Signatur, 68
mehrsortige Struktur, 69
minimale erfüllende Belegung, 25
Minimierung, 21
Mod, 91
Modalität, 49
Modallogik
    Semantik, 49
    Syntax, 49
Modallogische Systeme, 54
Modell, 10, 80
    eingeschränktes, 55
Morphismus
    beschränkt, 56

$\mathfrak{N}$, 67
$\mathfrak{N}^{<}$, 67
NAND, 20
Nichtstandardmodell, 92, 103
$\mathfrak{N}^{\omega}$, 67
NOR, 20
Normalform
    disjunktive, 12
    konjunktive, 12
    pränexe, 96

notwendigerweise, 49
NP-vollständig, 16

Ordnung, 67

partielle Ordnung, 67
Peano-Arithmetik, 92
Peano-Axiomensystem, 92
Prädikatenlogik, 65
    der zweiten Stufe, 92
    Semantik, 79
    Syntax, 75
Prämisse, 37
pränexe Normalform, 96
Primformel, 76
Primterm, 75
Putnam, Hilary, 27

QBF, 61
quantorenfreie Formel, 76
Quine, Willard Van Orman, 21
Quotientenstruktur, 72

Rahmen, 49
    reflexiver, 54
    symmetrischer, 54
    transitiver, 53
Reflexivität, 67, 71
Relation, 65
$\mathsf{res}(\Gamma)$, 28
Resolution, 27
    Endlichkeit, 29
Resolutionsableitung, 27
Resolutionswiderlegung, 27
Resolvente, 27
Robinson, J. A., 27

SAT, 16
    3-SAT, 17
    HORNSAT, 24
Satz, 76
Satz von Henkin, 45
Schaltkreis, 8
Schließen, 37
Schließen, 38, 39, 99
Schlussfolgerung, 37
Schlussregel, 39, 100
Semantik
    aussagenlogischer Formeln, 9
    der Aussagenlogik, 9
    der Modallogik, 49
    der Prädikatenlogik, 79
    kompositionale, 10
    quantifizierter Boole'scher Formeln, 61
semantisches Tableau, 47
Signatur, 66
    der Arithmetik, 67
    der Graphen, 67
    der Halbordnungen, 67
    der Mengen, 67
    mehrsortig, 68
Sokrates, 100
Sprache, 6
Standard-Arithmetik, 67
Stelligkeit, 65
strikte Halbordnung, 67
Struktur, 66
    σ-Struktur, 66
    mehrsortig, 69
    Quotienten-, 72
$\mathsf{sub}(\varphi)$, 7
Substitution, 77
surjektiv, 65
Symmetrie, 71
Syntax
    der Aussagenlogik, 5
    der Modallogik, 49
    der Prädikatenlogik, 75

Tarski, Alfred, 10, 79
TAUT, 16
Tautologie, 16, 50, 81
Teilformel, 7
Teilmodell, 55
    erzeugtes, 55
    punktgeneriert, 55
Teilrahmen, 55
Term, 75
    Prim-, 75
totale Ordnung, 67
Träger, 66
Transitivität, 67, 71

Universum, 66, 69

$\mathsf{Var}(\varphi)$, 7
Variable
    aussagenlogische, 5
    freie, 76

gebundene, 76
Vergleichbarkeit, 67
verträglich, 71
vollständig, 44
coNP-, 16
funktional, 14
NP-, 16
PSPACE-, 61
widerlegungs-, 29
Vollständigkeit
einer Formelmenge, 44
Vollständigkeitssatz, 39, 47, 101
für Resolution, 29

Wahrheitswert, 9
Widerlegungsbeweis, 27
widerlegungsvollständig, 29
Widerspruchsfreiheit, 44
Wort, 6

$\mathfrak{Z}$, 67