Logik
(und Formale Systeme)

Heribert Vollmer

Institut für Theoretische Informatik
Leibniz Universität Hannover

Vorlesung im Sommersemester 2015

Inhalt

Die Sprache der Aussagenlogik

Das Alphabet Σ_{AL} der Aussagenlogik besteht aus

1. der Menge der aussagenlogischen Variablen
   $\mathsf{Var} = \{p_1, p_2, p_3 \ldots\}$,
2. den Konstanten $0, 1$,
3. den Konnektoren
   - $\wedge$ (Konjunktion)
   - $\vee$ (Disjunktion)
   - $\neg$ (Negation)
   - $\rightarrow$ (Implikation)
   - $\leftrightarrow$ (Äquivalenz oder Biimplikation)
4. und den Klammersymbolen $($, $)$.

Formal ist also

$$\Sigma_{AL} = \mathsf{Var} \cup \{0, 1, \wedge, \vee, \neg, \rightarrow, \leftrightarrow, (,)\} \ .$$

Die Sprache der Aussagenlogik

Über dem Alphabet Σ_{AL} definieren wir jetzt induktiv die Menge der aussagenlogischen Formeln Form_{AL}. Formal ist Form_{AL} eine Sprache über Σ_{AL}, d.h. $\mathsf{Form}_{AL} \subseteq \Sigma_{AL}^*$.

Definition (Syntax der Aussagenlogik)

Die Menge der aussagenlogischen Formeln Form_{AL} ist induktiv definiert als:

1. Jede Variable $p \in \mathsf{Var}$ und Konstante $0, 1$ ist in Form_{AL}.
2. Seien $\varphi, \psi \in \mathsf{Form}_{AL}$. Dann ist auch
   - $\neg\varphi \in \mathsf{Form}_{AL}$
   - $(\varphi \vee \psi) \in \mathsf{Form}_{AL}$
   - $(\varphi \wedge \psi) \in \mathsf{Form}_{AL}$
   - $(\varphi \rightarrow \psi) \in \mathsf{Form}_{AL}$
   - $(\varphi \leftrightarrow \psi) \in \mathsf{Form}_{AL}$.

Konventionen

Wir nutzen die folgenden Klammersparungsregeln:

- Äußere Klammern werden weggelassen.
- $\neg$ bindet am stärksten,
- $\wedge$ bindet stärker als $\vee$ und
- $\vee$ wiederum bindet stärker als $\rightarrow$ und $\leftrightarrow$.
- Für mehrfache Konnektoren desselben Typs werden die Klammern von rechts nach links gesetzt, z.B. $p \rightarrow q \rightarrow r$ entspricht $(p \rightarrow (q \rightarrow r))$.

Zusätzlich verwenden wir noch die folgenden Abkürzungen:

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

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

Induktive Definitionen - Beispiel

Definition

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

1. $\mathsf{sub}(0) = \{0\}$ und $\mathsf{sub}(1) = \{1\}$.
2. Für $p \in \mathsf{Var}$ ist $\mathsf{sub}(p) = \{p\}$.
3. Für $\varphi \in \mathsf{Form}_{\mathsf{AL}}$ ist $\mathsf{sub}(\neg\varphi) = \{\neg\varphi\} \cup \mathsf{sub}(\varphi)$.
4. Für $\varphi, \psi \in \mathsf{Form}_{\mathsf{AL}}$ und $\circ \in \{\wedge, \vee, \rightarrow, \leftrightarrow\}$ ist

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

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

Syntax von $\mathsf{Form}_{\mathsf{AL}}$ – Kurzform in EBNF

$$\phi ::= p \mid 0 \mid 1 \mid \neg\phi \mid (\phi \wedge \phi) \mid (\phi \vee \phi) \mid (\phi \rightarrow \phi) \mid (\phi \leftrightarrow \phi),$$

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

Die Semantik der Aussagenlogik

Definition

Eine aussagenlogische Belegung $\mathfrak{I}$ ist eine Abbildung

$$\mathfrak{I} : \mathsf{Var} \rightarrow \{0, 1\}.$$

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

$$\hat{\mathfrak{I}} : \mathsf{Form}_{\mathsf{AL}} \rightarrow \{0, 1\}$$

erweitert werden vermöge von:

1. $\hat{\mathfrak{I}}(p) = \mathfrak{I}(p)$ für $p \in \mathsf{Var}$.
2. $\hat{\mathfrak{I}}(0) = 0$ und $\hat{\mathfrak{I}}(1) = 1$.
3. $\hat{\mathfrak{I}}(\neg\varphi) = 1$, falls $\hat{\mathfrak{I}}(\varphi) = 0$, für $\varphi \in \mathsf{Form}_{\mathsf{AL}}$.
4. $\hat{\mathfrak{I}}(\varphi \wedge \psi) = 1$, falls $\hat{\mathfrak{I}}(\varphi) = 1$ und $\hat{\mathfrak{I}}(\psi) = 1$, für $\varphi, \psi \in \mathsf{Form}_{\mathsf{AL}}$.
5. $\hat{\mathfrak{I}}(\varphi \vee \psi) = 1$, falls $\hat{\mathfrak{I}}(\varphi) = 1$ oder $\hat{\mathfrak{I}}(\psi) = 1$, für $\varphi, \psi \in \mathsf{Form}_{\mathsf{AL}}$.

Definition von $\hat{\mathfrak{I}}$ (Forts.)

Für $\varphi, \psi \in \mathsf{Form}_{\mathsf{AL}}$ ergibt sich für die Konnektoren $\circ \in \{\rightarrow, \leftrightarrow\}$ die Definition von $\hat{\mathfrak{I}}(\varphi \circ \psi)$ aus nachfolgender Tabelle.

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

Notation

Wir unterscheiden zur Vereinfachung der Notation oft nicht zwischen $\mathfrak{I}$ und $\hat{\mathfrak{I}}$, falls sich dadurch keine Zweideutigkeiten ergeben können.

Erfüllende Belegungen

- 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.
- $\mathfrak{I}$ ist eine erfüllende Belegung für eine Formel φ, falls $\hat{\mathfrak{I}}(\varphi) = 1$.
- Notation: $\mathfrak{I} \models \varphi$.
- $\mathfrak{I}$ heißt Modell für φ.
- Ist Φ eine Formelmenge, so schreiben wir $\mathfrak{I} \models \Phi$, falls $\mathfrak{I} \models \varphi$ für alle $\varphi \in \Phi$ gilt.

SAT und TAUT

- Die Menge aller erfüllbaren Formeln wird bezeichnet mit

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

- Eine Formel φ ist eine Tautologie, falls sie durch alle Belegungen erfüllt wird.
- Notation: $\models \varphi$.
- Die Menge aller Tautologien ist $\mathsf{TAUT} = \{\varphi \in \mathsf{Form}_{\mathsf{AL}} \mid \models \varphi\}$.
- Cook 71, Karp 72, Levin 73: SAT ist NP-vollständig und TAUT ist coNP-vollständig.

Ein erster Algorithmus für SAT

Wahrheitstafelmethode

Eingabe: φ
Für alle Belegungen $\mathfrak{I}$ für φ DO
  IF $\mathfrak{I} \models \varphi$ THEN Ausgabe(erfüllbar)
Ausgabe(unerfüllbar)

Rechtfertigung: Koinzidenzlemma

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 $\mathfrak{I}$ und $\mathfrak{I}'$ zwei Belegungen mit $\mathfrak{I}(p) = \mathfrak{I}'(p)$ für alle $p \in \mathsf{Var}(\varphi)$, so gilt $\mathfrak{I}(\varphi) = \mathfrak{I}'(\varphi)$.

SAT und TAUT

Bemerkung

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

Beweis.

$$\begin{aligned} \varphi \notin \mathsf{SAT} &\iff \text{Für alle Belegungen } \mathfrak{I} \text{ gilt } \mathfrak{I} \not\models \varphi \\ &\iff \text{Für alle Belegungen } \mathfrak{I} \text{ gilt } \mathfrak{I} \models \neg\varphi. \\ &\iff \neg\varphi \in \mathsf{TAUT} \end{aligned}$$

□

Äquivalenz

Definition

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

  $$\Phi \models \Psi \ ,$$

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

Wichtige Äquivalenzen

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

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

Wozu Äquivalenzen gut sind

Äquivalenzsatz (Ersetzbarkeitstheorem)

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

Beweis.

Induktiv über den Aufbau von σ (siehe Übungen). $\square$

Normalformen – Definition

- Literale sind negierte oder unnegierte Variablen.
- 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}.
- 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}.

Äquivalente KNF und DNF

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

Beweis.
Ergibt sich zusammen mit dem Äquivalenzsatz aus dem nachfolgenden Algorithmus. $\square$

Algorithmus zum Umformen in KNF

Eingabe: eine Formel φ

1: Ersetze in φ jedes Vorkommen einer Teilformel der Form

$$\begin{array}{lll} \psi \rightarrow \theta & \text{durch} & \neg\psi \vee \theta \\ \psi \leftrightarrow \theta & \text{durch} & (\psi \wedge \theta) \vee (\neg\psi \wedge \neg\theta) \end{array}$$

bis keine derartige Teilformel mehr vorkommt.

2: Ersetze jedes Vorkommen einer Teilformel der Form

$$\begin{array}{lll} \neg\neg\psi & \text{durch} & \psi \\ \neg(\psi \wedge \theta) & \text{durch} & \neg\psi \vee \neg\theta \\ \neg(\psi \vee \theta) & \text{durch} & \neg\psi \wedge \neg\theta \end{array}$$

bis keine derartige Teilformel mehr vorkommt.

3: Ersetze jedes Vorkommen einer Teilformel der Form

$$\begin{array}{lll} \psi \vee (\sigma \wedge \theta) & \text{durch} & (\psi \vee \sigma) \wedge (\psi \vee \theta) \\ (\sigma \wedge \theta) \vee \psi & \text{durch} & (\sigma \vee \psi) \wedge (\theta \vee \psi) \end{array}$$

bis keine derartige Teilformel mehr vorkommt.

Größe der KNF

Bemerkung

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.

Korollar

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

Der Endlichkeitssatz

Satz

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

Erfüllbarkeitstests

Erfüllbarkeitsproblem SAT

Eingabe:	Eine aussagenlogische Formel φ
Frage:	Ist φ erfüllbar?

Brute-Force-Algorithmus

Werte φ unter allen Belegungen der Variablen von F aus.

Dieser Algorithmus ist nicht effizient.

- Sei φ eine Formel in 100 Variablen.
- $\Rightarrow 2^{100}$ Belegungen
- Angenommen, ein Rechner kann eine Milliarde Belegungen pro Sekunde testen.
- $\Rightarrow$ Rechenzeit $> 10^{13}$ Jahre

SAT ist NP-vollständig.

Hornformeln

Definition

Eine Hornformel ist eine Formel in konjunktiver Normalform, die höchstens ein positives Literal pro Klausel enthält.

Beispiel

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

- erstmals betrachtet von Alfred Horn (1918–2001)
- Anwendung: logisches Programmieren (PROLOG)

Alternative Sichtweise auf Hornformeln

Jede Hornklausel ist äquivalent zu einer Implikation der Form

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

Beispiel

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

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	$1 \to p_2$

Effizienter Algorithmus für HORNSAT

HORNSAT

Eingabe:	Eine Hornformel Formel F
Frage:	Ist F erfüllbar?

Satz

- Für erfüllbare Hornformeln lässt sich eine erfüllende Belegung in Polynomialzeit konstruieren.
- HORNSATist also effizient lösbar.

Der Algorithmus

Idee

Konstruiere bei Eingabe F eine "minimale" erfüllende Belegung b für F (falls existent).

Eingabe: Hornformel φ

$\mathfrak{I}(p) := \begin{cases} 1 & \text{falls } 1 \to p \text{ in } \varphi \text{ vorkommt} \\ 0 & \text{sonst} \end{cases}$

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**

  $\mathfrak{I}(q) := 1$

end while

if $\mathfrak{I}$ erfüllt φ **then return** Erfüllbar

else return Unerfüllbar

end if

Der Algorithmus im Beispiel

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: $\quad\mathfrak{I}(q) := 1$
4: **end while**
5: **if** $\mathfrak{I}$ erfüllt φ **then return** Erfüllbar
6: **else return** Unerfüllbar
7: **end if**

$\varphi = (p_1 \wedge p_2 \to p_3) \wedge (p_2 \to p_1) \wedge (p_2 \wedge p_4 \to 0) \wedge (1 \to p_2)$

Programmschritt	$\mathfrak{I}(p_1)$	$\mathfrak{I}(p_2)$	$\mathfrak{I}(p_3)$	$\mathfrak{I}(p_4)$
Zeile 1	0	1	0	0
While-Schleife, 1. Durchlauf	1	1	0	0
2. Durchlauf	1	1	1	0

Ausgabe „Erfüllbar“

Korrektheit des Algorithmus

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: $\quad\mathfrak{I}(q) := 1$
4: **end while**
5: **if** $\mathfrak{I}$ erfüllt φ **then return** Erfüllbar
6: **else return** Unerfüllbar
7: **end if**

Behauptung

1. Die Ausgabe “Erfüllbar” in Zeile 4 ist korrekt. (Klar)
2. Die Ausgabe “Unerfüllbar” in Zeile 5 ist korrekt.

Laufzeit des Algorithmus

Eingabe: Hornformel φ

$\mathfrak{I}(p) := \begin{cases} 1 & \text{falls } 1 \to p \text{ in } \varphi \text{ vorkommt} \\ 0 & \text{sonst} \end{cases}$
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**
    $\mathfrak{I}(q) := 1$
end while
if $\mathfrak{I}$ erfüllt φ **then return** Erfüllbar
else return Unerfüllbar
end if

- Sei F eine Formel der Größe $n \Rightarrow F$ hat $\leq n$ Variablen.
- In jedem Durchlauf der While-Schleife wird eine Variable von 0 auf 1 gesetzt $\Rightarrow \leq n$ Durchläufe der While-Schleife
- Also: Laufzeit $\mathcal{O}(n^2)$

Weitere Resultate für HORNSAT

Satz (Dowling, Gallier 84)

HORNSAT ist in Linearzeit lösbar.

Satz

HORNSAT ist vollständig für die Komplexitätsklasse P.

Konjunktive Normalformen – Klauselschreibweise

- Eine Klausel ist eine endliche Menge von Literalen.
- 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

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

Erfüllbarkeit von Klauseln – Definition

- Eine Klausel C wird durch eine Belegung $\mathfrak{I}$ erfüllt, falls $\mathfrak{I}$ mindestens ein Literal aus C erfüllt.
  Schreibweise: $\mathfrak{I} \models C$
- Eine Klauselmenge $\Gamma = \{C_1, \ldots, C_n\}$ wird durch $\mathfrak{I}$ erfüllt, falls $\mathfrak{I} \models C_i$ für $i = 1, \ldots, n$.
  Schreibweise: $\mathfrak{I} \models \Gamma$
- Die leere Klausel wird mit $\square$ bezeichnet und ist unerfüllbar.

Resolution

- Eingeführt von Blake 1937 sowie Davis und Putnam 1960 und Robinson 1965
- Resolutionsbeweise operieren mit Klauseln.
- Resolutionsbeweise sind Widerlegungsbeweise.

Definition

Seien C und D Klauseln mit $p \in C$ und $\neg p \in D$.
Die Resolutionsregel angewendet auf C und D liefert die Klausel $(C \setminus \{p\}) \cup (D \setminus \{\neg p\})$ (sog. Resolvente).

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

Beachte: $\{C,D\} \models (C \setminus \{p\}) \cup (D \setminus \{\neg p\})$.

Resolutionsableitungen

Definition

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

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

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

1. $C_i \in \Gamma$ oder
2. es existieren $1 \leq j_1 \leq j_2 < i$ mit

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

Schreibweise: $\Gamma \vdash_{\mathsf{Res}} C$.

Resolutionswiderlegungen

Definition

- Beachte: Falls $\Gamma \vdash_{\mathsf{Res}} C$, dann $\Gamma \vDash C$.
- Falls $\Gamma \vdash_{\mathsf{Res}} \Box$, dann $\Gamma \vDash \Box$; also ist Γ unerfüllbar.
- Eine Resolutionswiderlegung von Γ ist eine Resolutionsableitung von $\Box$ aus Γ.

Beispiel für eine Resolutionswiderlegung

$$\Gamma = \{\{p, q\}, \{p, \neg q\}, \{\neg p, q\}, \{\neg p, \neg q\}\}$$

Eine Resolutionswiderlegung von Γ ist:

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

Endlichkeit der Resolution

Definition

Sei Γ eine Menge von Klauseln.

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

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

Also: $\Gamma \vdash_{\mathsf{Res}} C$ gdw. $C \in \mathsf{res}^*(\Gamma)$.

Satz

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 $\mathsf{res}^*(\Gamma) = \mathsf{res}^n(\Gamma)$.

Vollständigkeit

- Semantischer Begriff: Folgern $\models$
- Syntaktischer Begriff: Ableiten $\vdash$,
  beispielsweise: $\vdash_{\mathsf{Res}}$
- Das Ableiten soll das Folgern nachbilden/imitieren.
- $\vdash$ wird also so definiert, dass $\Phi \vdash \varphi \implies \Phi \models \varphi$
  ("Korrektheit")
  ($\Phi \subseteq \mathsf{Form}_{\mathsf{AL}}$, $\varphi \in \mathsf{Form}_{\mathsf{AL}}$).
- Ziel bei der Definition von $\vdash$ ist ein Vollständigkeitssatz:

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

Vollständigkeit von Resolution

- Resolution ist ein Widerlegungskalkül, d.h. wir zeigen $\varphi \in \mathsf{TAUT}$ mittels $\neg\varphi \vdash_{\mathsf{Res}} \Box$.
- Der Resolutionskalkül ist nicht vollständig:
  Beispiel: $p \models p \vee q$, aber $p \nvdash_{\mathsf{Res}} p \vee q$.
- Resolution ist jedoch widerlegungsvollständig.

Vollständigkeitssatz für Resolution

Sei Γ eine endliche Klauselmenge. Dann ist Γ genau dann unerfüllbar, wenn es eine Resolutionswiderlegung von Γ gibt.
Symbolisch: $\Gamma \models \Box \iff \Gamma \vdash_{\mathsf{Res}} \Box$

Ein Tautologietest

Der Resolutionskalkül liefert einen Tautologietest, der die Struktur der gegebenen Formel ausnutzt (anders als die Wahrheitstafelmethode):

$\varphi \in \mathsf{TAUT}$ mittels $\neg\varphi \vdash_{\mathsf{Res}} \Box$.

Eingabe: eine Formel φ
1: Berechne eine KNF Γ für $\neg\phi$.
2: Berechne $\mathsf{res}^*(\Gamma)$.
3: Prüfe, ob $\Box \in \mathsf{res}^*(\Gamma)$.
4: Falls ja: Ausgabe "φ ist eine Tautologie".

Hoffnung: Bessere Effizienz.

Harte Instanzen für Resolution

Satz

Es gibt eine Folge Γ_n von Klauselmengen mit folgenden Eigenschaften:

1. Γ_n ist eine unerfüllbare Klauselmenge in n Variablen.
2. Die Anzahl der Klauseln in Γ_n ist polynomiell in n.
3. Die minimale Resolutionswiderlegung von Γ_n enthält exponentiell viele Klauseln.

Auch Resolution erfordert exponentielle Zeitkomplexität.

Noch ein Problem

Resolution ist ein Widerlegungskalkül, d.h. wir zeigen $\varphi \in \mathsf{TAUT}$ mittels $\neg\varphi \vdash_{\mathsf{Res}} \square$.

Problem

$\neg\varphi$ muss in KNF vorliegen.

- Dies ist gegeben für Formeln φ in DNF:
  Für diese erhalten wir nach den de Morgan'schen Regeln eine KNF-Formel für $\neg\varphi$.
- Allgemein bieten sich folgende Möglichkeiten an:
  1. $\neg\varphi$ wird in eine KNF-Formel Γ umgeformt.
     Nachteil: die Größe von Γ kann exponentiell in der Größe von φ sein.
  2. Wir formen $\neg\varphi$ in eine erfüllbarkeitsäquivalente Klauselmenge $\Gamma_{\neg\varphi}$ um.

Die zweite Möglichkeit im Detail

- Wir formen eine beliebige Formel χ in eine erfüllbarkeitsäquivalente Klauselmenge Γ_χ um.
- Für jede Teilformel ψ von χ führen wir eine neue Erweiterungsvariable q_ψ ein.
- Für eine Formel ψ definieren wir die Klauselmenge Δ_ψ:

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

- Nun setzen wir

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

Die zweite Möglichkeit im Detail

Satz

χ und Γ_χ sind erfüllbarkeitsäquivalent,
d.h. χ ist erfüllbar genau dann, wenn Γ_χ erfüllbar ist.

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.

Beispiel

- $\chi = (x \wedge y) \vee \neg z$
- 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	$\{q_y, \neg y\}, \{\neg q_y, y\}$

$\Gamma_\chi =$ alle obigen Klauseln zusammen mit $\{q_{(x\wedge y)\vee\neg z}\}$

Schlussfolgerungen

Prämissen:

1. Wenn ich den Schalter drücke, geht das Licht an.
2. Ich drücke den Schalter.

Konklusion:

1. Das Licht geht an.

Schlussfolgerungen

Schreibweise der Logiker:

Wenn ich den Schalter drücke, geht das Licht an.	$\phi \rightarrow \psi$
Ich drücke den Schalter.	ϕ
$\therefore$ Das Licht geht an.	$\therefore \psi$

In unserer Schreibweise:

$\{\varphi \rightarrow \psi, \psi\} \models \psi$

Schlussfolgerungen

Wie überprüft man die Gültigkeit von Schlussfolgerungen?

Konklusion negieren	$\neg\psi$
Zu Prämisse hinzufügen	$(\phi \rightarrow \psi) \wedge \phi \wedge \neg\psi$
Auf Erfüllbarkeit überprüfen ("Konsistenz")	mit Wahrheitstafel

Geht das effizienter?
Kann man menschliche Schlussfolgerungen mechanisch (algorithmisch) imitieren?

$\rightsquigarrow$ Maschinelles Beweisen (Grundlage der KI)

Folgern und Schließen

- Semantischer Begriff: Folgern $\models$
- Syntaktischer Begriff: Schließen/Ableiten/Beweisen $\vdash$
- Das Schließen soll das Folgern nachbilden/imitieren.
- $\vdash$ wird also so definiert, dass $\Phi \vdash \varphi \implies \Phi \models \varphi$ ("Korrektheit").
- Ziel bei der Definition von $\vdash$ ist ein Vollständigkeitssatz:

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

  ($\Phi \subseteq \mathsf{Form}_{\mathsf{AL}}$, $\varphi \in \mathsf{Form}_{\mathsf{AL}}$).

Ein weiteres Beispiel

Schlussfolgerung:

$$\begin{array}{ll} & \text{Alle Menschen sind sterblich.} \\ & \text{Sokrates ist ein Mensch.} \\ \hline \therefore & \text{Sokrates ist sterblich.} \end{array}$$

Mit den Mitteln der Aussagenlogik nicht sinnvoll formalisierbar.

In der Prädikatenlogik (später):

$$\begin{array}{ll} & \forall x\big(\mathsf{H}(x) \to \mathsf{M}(x)\big) \\ & \mathsf{H}(\mathsf{Sokrates}) \\ \hline \therefore & \mathsf{M}(\mathsf{Sokrates}) \end{array}$$

Schlussregeln

Wir betrachten nur Formeln mit den Operatoren $\neg, \wedge, \vee$.

- $\varphi \in \Phi \Longrightarrow \Phi \vdash \varphi$
- Fallunterscheidung:
  $(\Phi \cup \{\varphi'\}) \vdash \varphi,\ (\Phi \cup \{\neg\varphi'\}) \vdash \varphi \Longrightarrow \Phi \vdash \varphi$
- Indirekter Beweis:
  $(\Phi \cup \{\neg\varphi\}) \vdash \varphi',\ (\Phi \cup \{\neg\varphi\}) \vdash \neg\varphi' \Longrightarrow \Phi \vdash \varphi$
- Abtrennungsregel, modus ponens:
  $\Phi \vdash (\neg\varphi' \vee \varphi), \Phi \vdash \varphi' \Longrightarrow \Phi \vdash \varphi$
  (Beache: $(\neg\varphi' \vee \varphi) \equiv (\varphi' \to \varphi)$.)
- Einführung der Alternative:
  $\Phi \vdash \varphi \Longrightarrow \Phi \vdash (\varphi \vee \varphi'),\ \Phi \vdash (\varphi' \vee \varphi)$ für beliebiges φ'
- Einführung der Konjunktion:
  $\Phi \vdash \varphi,\ \Phi \vdash \varphi' \Longrightarrow \Phi \vdash (\varphi \wedge \varphi')$
- Auflösung der Konjunktion:
  $\Phi \vdash (\varphi \wedge \varphi') \Longrightarrow \Phi \vdash \varphi,\ \Phi \vdash \varphi'$

Endlichkeitssatz der Ableitung

Satz

Falls $\Phi \vdash \varphi$, so gibt es ein endliches $\Phi_0 \subseteq \Phi$ sodass $\Phi_0 \vdash \varphi$.

Korrektheitssatz

Satz
Falls $\Phi \vdash \varphi$, so ist $\Phi \models \varphi$.

Notation

$\Phi^{\models} = \{\varphi \mid \Phi \models \varphi\}$ ($^{\models}$ heißt Folgerungsoperator).
$\Phi^{\vdash} = \{\varphi \mid \Phi \vdash \varphi\}$ ($^{\vdash}$ heißt Ableitungsoperator).

Notation

$$\Phi_0^{\vdash} = \Phi$$

$$\Phi_{n+1}^{\vdash} = \Phi_n \cup \left\{\begin{array}{l}\text{Formeln, die Anwendung einer Schlussregel aus Formeln}\\ \text{aus } \Phi_n^{\vdash} \text{ abgeleitet werden können}\end{array}\right\}$$

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

Klar: $\Phi = \Phi_0^{\vdash} \subseteq \Phi_1^{\vdash} \subseteq \Phi_2^{\vdash} \subseteq \cdots \subseteq \Phi^{\vdash} = \bigcup_{n \geq 0} \Phi_n^{\vdash}$.

Für den Beweis des Korrektheitssatzes ist zu zeigen:
Für alle Formelmengen Φ und alle $n \in \mathbb{N}$ gilt: $\Phi_n^{\vdash} \subseteq \Phi^{\models}$.

Widerspruchsfreiheit/Konsistenz

Definition

Φ heißt konsistent (widerspruchsfrei), falls es kein φ gibt mit $\Phi \vdash \varphi$ und $\Phi \vdash \neg\varphi$.

Satz

1. Ist Φ nicht konsistent, so gilt $\Phi^{\vdash} = \mathsf{Form}_{\mathsf{AL}}$.
2. Ist Φ konsistent, so ist $\Phi \cup \{\varphi\}$ oder $\Phi \cup \{\neg\varphi\}$ konsistent.

Vollständigkeit

Definition

Φ heißt vollständig, falls für jedes φ gilt: $\varphi \in \Phi$ oder $\neg\varphi \in \Phi$.

Satz

Sei Φ konsistent und vollständig. Dann gilt:

1. $\Phi = \Phi^{\vdash}$.
2. $\neg\varphi \in \Phi \Leftrightarrow \varphi \notin \Phi$.
3. $(\varphi_1 \vee \varphi_2) \in \Phi \Leftrightarrow \varphi_1 \in \Phi$ oder $\varphi_2 \in \Phi$.
4. $(\varphi_1 \wedge \varphi_2) \in \Phi \Leftrightarrow \varphi_1 \in \Phi$ und $\varphi_2 \in \Phi$.

Der Vollständigkeitssatz

Satz von Henkin

Φ genau dann konsistent, wenn Φ ein Modell hat.

Vollständigkeitssatz

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

Folgerung

$\Phi^{\models} = \Phi^{\vdash}$.

Was haben wir gesehen?

Syntax	**Semantik**
Ableitung/Beweis/Schluss	Folgerung/Implikation
$\Phi \vdash \varphi$	$\Phi \vDash \varphi$
Φ ist konsistent	Φ ist erfüllbar (Φ hat ein Modell)
Φ ist inkonsistent	Φ ist unerfüllbar (Φ hat kein Modell)

Nichtklassische Logiken

- Neben der klassischen Aussagenlogik gibt es eine Vielzahl nichtklassischer Logiken.
- Oftmals erweitern diese die klassische Aussagenlogik um weitere Ausdrucksmöglichkeiten.
- Wir betrachten als wichtiges Beispiel modale Logiken.

Philosophische Modallogik

Gottfried Wilhelm Leibniz: Idee der möglichen Welt,

d. h. Vorstellung, wie die Welt beschaffen sein könnte (die Gesetze der Logik achtend).

„Wir leben in der besten aller möglichen Welten“ (Theodizee, 1710).

Saul A. Kripke (Naming and Necessity, 1980):
„Mögliche Welten werden festgelegt und nicht durch mächtige Teleskope entdeckt.“

Rudolf Carnap (Meaning and Necessity, 1947):

- Eine Aussage ist notwendigerweise wahr, wenn sie in allen möglichen Welten wahr ist.
- Eine Aussage ist möglicherweise wahr, wenn sie in wenigstens einer möglichen Welten wahr ist.

Die Sprache der Modallogik

- Zusätzlich zu der Sprache der Aussagenlogik enthält die modale Sprache den unären Konnektor $\Box$ („notwendigerweise“).
- Wir benutzen zusätzlich den Konnektor $\Diamond$ („möglicherweise“), den wir als Abkürzung für $\neg\Box\neg$ definieren.

Syntax in EBNF:

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

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

Semantik der Modallogik

Definition
Eine Kripke-Struktur (oder Kripke-Rahmen) ist ein Paar $K = (W, R)$ mit

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

Definition
Ein Modell der Modallogik ist ein Paar (K, V) mit

- $K = (W, R)$ ist eine Kripke-Struktur,
- $V : \mathsf{Var} \mapsto \mathcal{P}(W)$ ist eine Abbildung, die jeder aussagenlogischen Variablen p eine Menge $V(p)$ von Welten zuordnet.
  ($\mathcal{P}(W)$ ist die Potenzmenge von W).

Modale Erfüllbarkeit (Kripke-Semantik)

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

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

Semantik von $\Diamond$

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

Satz

Sei φ eine modale Formel.
Sei $M = (W, R, V)$ ein Modell und $w \in W$ eine Welt.
Dann gilt

$$M, w \models \Diamond\varphi$$

falls es eine Welt $v \in W$ gibt mit $(w, v) \in R$ und

$$M, v \models \varphi.$$

Interpretationen

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

  $\Box\varphi$ „φ ist notwendigerweise wahr“
  $\Diamond\varphi$ „φ ist möglicherweise wahr“
- Deontische Deutung:

  $\Box\varphi$ „φ ist geboten“
  $\Diamond\varphi$ „φ ist erlaubt“
- Temporale Deutung:

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

Modales SAT und TAUT

Definition

- Eine modale Formel φ ist erfüllbar wenn es ein Modell $M = (W, R, V)$ und eine Welt $w \in W$ mit $M, w \models \varphi$ gibt.
- Dual dazu ist φ eine modale Tautologie falls für jedes Modell $M = (W, R, V)$ und jedes $w \in W$ gilt $M, w \models \varphi$.

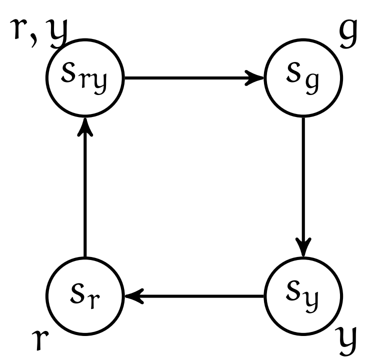
Beispiele

- Jede klassische Tautologie ist auch eine modale Tautologie.
- $\Box(\varphi_1 \to \varphi_2) \to (\Box\varphi_1 \to \Box\varphi_2)$ ist eine modale Tautologie.

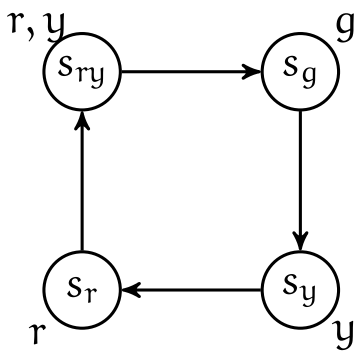
Beispiel Ampelschaltung

- Zustände $W = \{s_r, s_{ry}, s_g, s_y\}$
- Relation $R = \{(s_{ry}, s_g), (s_g, s_y), (s_y, s_r), (s_r, s_{ry})\}$
- Variablen $\Phi = \{r, g, y\}$ (Rot, Grün, Gelb) beschreiben, in welchen Zuständen eine Lampe aktiviert ist

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

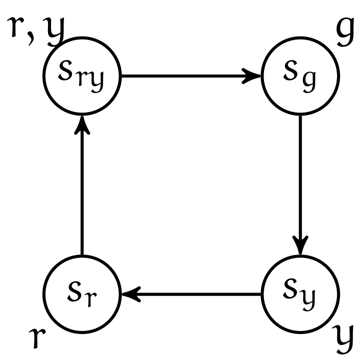


Beispiel Ampelschaltung



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

Beispiel Ampelschaltung

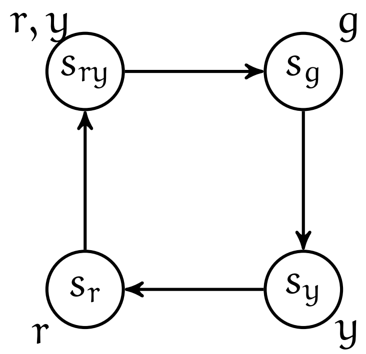


- Für alle $w \in W$ gilt

$$M, w \vDash (r \wedge y \wedge \neg g) \rightarrow \neg\Diamond(r \wedge \neg y \wedge \neg g)$$

- Inhaltlich: nach Rot-Gelb folgt niemals Rot
- kurz: $M \vDash (r \wedge y \wedge \neg g) \rightarrow \neg\Diamond(r \wedge \neg y \wedge \neg g)$

Beispiel Ampelschaltung



- Für alle Formeln φ, alle Belegungen V' und alle Welten $w \in W$ gilt

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

- Inhaltlich: der Rahmen ist ein Viererzyklus
- kurz: $(W, R) \models \varphi \leftrightarrow \Diamond\Diamond\Diamond\Diamond\varphi$

Charakterisierung von Rahmeneigenschaften

(Zur Erinnerung: Rahmen = Kripke-Struktur)

Satz

Ein Rahmen $F = (W, R)$ ist transitiv genau dann, wenn für alle Formeln φ gilt

$$F \models \Diamond\Diamond\varphi \rightarrow \Diamond\varphi.$$

Satz

Ein Rahmen $F = (W, R)$ ist reflexiv genau dann, wenn für alle Formeln φ gilt

$$F \models \varphi \rightarrow \Diamond\varphi.$$

Satz

Ein Rahmen $F = (W, R)$ ist symmetrisch genau dann, wenn für alle Formeln φ gilt

$$F \models \varphi \rightarrow \Box\Diamond\varphi.$$

Beispiele modaler Logiken

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

(nach Clarence Irving Lewis, 1912)

Beispiele für prädikatenlogische Formeln

Aus der Philosophie:

Alle Menschen sind sterblich.	$\forall x(H(x) \to M(x))$
Sokrates ist ein Mensch.	$H(\mathsf{Sokrates})$
∴ Sokrates ist sterblich.	$\therefore M(\mathsf{Sokrates})$

Aus der Mathematik:

$\forall x_1 \forall x_2 (f(x,y) > 0 \to (x > 0 \wedge y > 0))$

Semantik der Prädikatenlogik:

Wir müssen die Bedeutung der nichtlogischen Symbole H, M, Sokrates, f, $>$, 0 festlegen.

$\rightsquigarrow$ Begriff der Struktur

Mathematische Strukturen

Informale Definition

- Eine Struktur (oder: Algebra) $\mathfrak{A}$ ist eine nichtleere Menge A zusammen mit ausgezeichneten Funktionen, Relationen und Konstanten aus A.
- A heißt Träger oder Grundmenge.
- 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, so schreiben wir

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

Beispiele

- die Struktur der natürlichen Zahlen $\mathfrak{N} = (\mathbb{N}; <; +, \cdot; 0, 1)$
- beliebige Gruppen $\mathfrak{G} = (G; \circ; 1)$
- Abelsche Gruppen $\mathfrak{A} = (G; +; 0)$.
- gerichtete Graphen $\mathfrak{G} = (V; E)$

Signaturen

Formale Definition

Eine Signatur σ ist ein Tupel

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

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, $i \in I_c$.

Achtung

Eine Signatur σ enthält lediglich Zeichen, die sog. nichtlogischen Symbole. Diese werden erst in einer σ-Struktur $\mathfrak{A}$ interpretiert.

Beispiele für Signaturen

- Signatur der Arithmetik

$$\sigma_{Ar} = (<; +, \cdot; 0, 1),$$

mit
  - einem zweistelligen Relationssymbol $<$
  - zwei zweistelligen Funktionssymbolen $+, \cdot$
  - zwei Konstantensymbolen $0, 1$
- Signatur der Gruppen

$$\sigma_G = (\circ; 1),$$

mit
  - einem zweistelligen Funktionssymbol $\circ$
  - einem Konstantensymbol 1
- Signatur der gerichteten Graphen

$$\sigma_{Gr} = (E),$$

mit einem zweistelligen Relationssymbol E

Strukturen

Definition

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

- einem Träger $A \neq \emptyset$,
- 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 $i \in I_c$.

Beispiele für Strukturen über $\sigma_{\mathrm{Ar}} = (<; +, \cdot; 0, 1)$

- Struktur der natürlichen Zahlen

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

  wobei
  - den Träger $\mathbb{N}$
  - eine zweistellige Relation: $<^{\mathfrak{N}}$
  - zwei zweistellige Funktionen: $+^{\mathfrak{N}}, \cdot^{\mathfrak{N}}$
  - zwei Konstanten: $0^{\mathfrak{N}}, 1^{\mathfrak{N}}$
- Struktur der ganzen Zahlen

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

  wobei
  - den Träger $\mathbb{Z}$
  - eine zweistellige Relation: $<^{\mathfrak{Z}}$
  - zwei zweistellige Funktionen: $+^{\mathfrak{Z}}, \cdot^{\mathfrak{Z}}$
  - zwei Konstanten: $0^{\mathfrak{Z}}, 1^{\mathfrak{Z}}$

Konvention

Vereinfachte Schreibweise

Statt

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

schreiben wir auch

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

Achtung

Trotzdem sind

- die Signatur $\sigma_{\mathbb{N}} = (<; +, \cdot; 0, 1)$ und
- die Struktur $\mathfrak{N} = (\mathbb{N}; <; +, \cdot; 0, 1)$

zwei verschiedene Dinge.

Die Syntax der Prädikatenlogik

Sei eine Signatur σ vorgegeben.
Wir definieren die Sprache der σ-Formeln Form_σ wie folgt:
Der zugrundeliegende Zeichenvorrat enthält die folgenden Symbole:

1. eine abzählbar unendliche Menge

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

   von Variablen,
2. die Relations-, Funktions- und Konstantensymbole aus σ, die nichtlogischen Symbole,
3. die logischen Symbole $\wedge$, $\neg$, $\forall$, $=$ und
4. Klammern (,).

Terme

Über diesem Alphabet definieren wir induktiv Terme und Formeln. Wir beginnen mit den Termen:

Definition

Die Menge der σ-Terme ist induktiv definiert als:

1. Jede Variable ist eine σ-Term.
2. Jede Konstante aus σ ist ein Term.
3. Sind $t_1, \dots, t_n$ σ-Terme und ist f ein n-stelliges Funktionssymbol aus σ, 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 $x + y$ (d. h. wir verwenden Infix-Notation soweit üblich).

Induktion über den Termaufbau

Induktiv über den Termaufbau können wir Beweise führen oder Eigenschaften definieren wie in folgendem Beispiel:

Definition

Die Menge der Variablen $\mathsf{Var}(t)$ in einem Term t ist induktiv definiert als

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

Formeln

Die Menge der σ-Formeln ist induktiv definiert als:

1. Sind s und t σ-Terme, so ist $s = t$ eine σ-Formel.
2. Sind $t_1, \dots, t_n$ σ-Terme und ist R ein n-stelliges Relationssymbol aus σ, so ist $R(t_1, \dots, t_n)$ eine σ-Formel.
3. Sind φ und ψ σ-Formeln, so auch
   - $(\varphi \wedge \psi)$,
   - $\neg\varphi$ und
   - $\forall x \varphi$ mit $x \in \mathsf{Var}$.

Formeln gemäß 1 und 2 heißen Primformeln (oder atomare Formeln).

Abkürzungen

Wir verwenden die Abkürzungen:

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

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

Beispiel

$\forall x \exists y \; x + y = 0$ ist eine σ_{Ar}-Formel.

Freie und gebundene Variablen

Definition

- 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 $\mathrm{gbd}\varphi$.
- 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

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

Aussagen

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

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

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

Substitutionen

Definition
Sei φ eine Formel, t ein Term und x eine Variable. Induktiv über den Term- und Formelaufbau definieren wir die Substitution von x durch t in φ wie folgt:

1. für eine Variable y:
$$y[x/t] := \begin{cases} t & \text{falls } x = y \\ y & \text{sonst} \end{cases}$$
2. für ein Konstantensymbol c:
$$c[x/t] := c$$
3. für ein n-stelliges Funktionssymbol f:
$$f(t_1, \ldots, t_n)[x/t] \; := \; f(t_1[x/t], \ldots, t_n[x/t])$$

Definition Substitution (Forts.)

4. für Terme t_1, t_2:

$$(t_1 = t_2)[x/t] \;:=\; t_1[x/t] = t_2[x/t]$$

5. für ein n-stelliges Relationssymbol R:

$$R(t_1, \dots, t_n)[x/t] \;:=\; R(t_1[x/t], \dots, t_n[x/t])$$

6. für Formeln φ, ψ:

$$(\varphi \wedge \psi)[x/t] := \varphi[x/t] \wedge \psi[x/t]$$

7. $(\neg\varphi)[x/t] \;:=\; \neg(\varphi[x/t])$
8. $(\forall y \varphi)[x/t] \;:=\; \begin{cases} \forall y \varphi & \text{falls } x = y \\ \forall y(\varphi[x/t]) & \text{sonst.} \end{cases}$

Die Semantik der Prädikatenlogik

Ziel:

Termen und Formeln semantische Interpretationen zuordnen, und zwar:

- Termen werden wir Elemente im Träger zuweisen.
- Formeln werden wir Wahrheitswerte $0, 1$ zuweisen.

Was fehlt noch?

Ist $\varphi \in \mathrm{Form}_\sigma$ und $\mathfrak{A}$ eine σ-Struktur, so ist die Bedeutung aller nichtlogischen Symbolen in φ klar.

Kleines Problem: Freie Variablen.

Beispiel: $\varphi(x) \triangleq \forall y\,((1 < y \wedge y < x) \to \forall z \neg x = y \cdot z)$.
Gilt φ in $\mathfrak{N}$?

Interpretationen

Definition

Sei σ eine Signatur. Ein σ-Interpretation ist ein Paar $\mathfrak{I} = (\mathfrak{A}, \beta)$ bestehend aus einer σ-Struktur $\mathfrak{A}$ und einer Belegung

$$\beta : \mathsf{Var} \to A$$

mit

$$\beta : x \mapsto x^{\beta} \quad .$$

Schreibweise:

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 $x^{\mathfrak{I}}$.

Auswertung von Termen

Definition

Der Wert eines Terms in einer Interpretation $\mathfrak{I} = (\mathfrak{A}, \beta)$ ist induktiv definiert als:

1. Induktionsanfang:
   - $x^{\mathfrak{I}} = \beta(x)$ für Variablen x
   - $c^{\mathfrak{I}} = c^{\mathfrak{A}}$ für Konstanten
2. Induktionsschritt:
   - $(f(t_1, \ldots, t_n))^{\mathfrak{I}} = f^{\mathfrak{I}}(t_1^{\mathfrak{I}}, \ldots, t_n^{\mathfrak{I}})$ für n-stellige Funktionssymbole aus σ.

Kompositionale Semantik nach Tarski

Definition (Induktive Wahrheitsdefinition)

Sei $\mathfrak{I} = (\mathfrak{A}, \beta)$ eine σ-Interpretation. Wir definieren induktiv über den Formelaufbau die Gültigkeit einer σ-Formel φ in Interpretation $\mathfrak{I}$:

1. $\mathfrak{I} \models s = t \iff s^{\mathfrak{I}} = t^{\mathfrak{I}}$ für σ-Terme s, t.
2. $\mathfrak{I} \models R(t_1, \ldots, t_n) \iff R^{\mathfrak{I}}(t_1^{\mathfrak{I}}, \ldots, t_n^{\mathfrak{I}})$ für n-stellige Relationen R und Terme $t_1, \ldots, t_n$.
3. $\mathfrak{I} \models \varphi \wedge \psi \iff \mathfrak{I} \models \varphi$ und $\mathfrak{I} \models \psi$ für σ-Formeln φ und ψ.
4. $\mathfrak{I} \models \neg\varphi \iff \mathfrak{I} \not\models \varphi$ (nicht $\mathfrak{I} \models \varphi$) für eine σ-Formel φ.
5. $\mathfrak{I} \models \forall x \varphi$ *... nicht so einfach!*

Gepatchte Interpretationen

Definition

Sei $\mathfrak{I} = (\mathfrak{A}, \beta)$ eine Interpretation, sei $x \in \mathsf{Var}$ und $a \in A$ ein Element aus dem Träger von $\mathfrak{A}$.
Wir definieren β_x^a als

$$\beta_x^a(y) = \begin{cases} a & \text{falls } y = x, \\ \beta(y) & \text{sonst.} \end{cases}$$

Ferner sei $\mathfrak{I}_x^a = (\mathfrak{A}, \beta_x^a)$.

Also:

Interpretation $\mathfrak{I}_x^a$ verhält sich wie $\mathfrak{I}$, nur die Variable x wird anders interpretiert.

Kompositionale Semantik nach Tarski

Definition (Induktive Wahrheitsdefinition)

Sei $\mathfrak{I} = (\mathfrak{A}, \beta)$ eine σ-Interpretation. Wir definieren induktiv über den Formelaufbau die Gültigkeit einer σ-Formel φ in Interpretation $\mathfrak{I}$:

1. $\mathfrak{I} \models s = t \iff s^{\mathfrak{I}} = t^{\mathfrak{I}}$ für σ-Terme s, t.
2. $\mathfrak{I} \models R(t_1, \ldots, t_n) \iff R^{\mathfrak{I}}(t_1^{\mathfrak{I}}, \ldots, t_n^{\mathfrak{I}})$ für n-stellige Relationen R und Terme $t_1, \ldots, t_n$.
3. $\mathfrak{I} \models \varphi \wedge \psi \iff \mathfrak{I} \models \varphi$ und $\mathfrak{I} \models \psi$ für σ-Formeln φ und ψ.
4. $\mathfrak{I} \models \neg\varphi \iff \mathfrak{I} \not\models \varphi$ (nicht $\mathfrak{I} \models \varphi$) für eine σ-Formel φ.
5. $\mathfrak{I} \models \forall x \varphi \iff \mathfrak{I}_x^a \models \varphi$ für alle $a \in A$ für eine σ-Formel φ und $x \in \mathrm{Var}$.

Beispiel

Beispiel

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

$$\begin{aligned} \mathfrak{I} &\models x_1 + x_2 = x_3 \\ \mathfrak{I} &\not\models x_7 < x_4 \\ \mathfrak{I} &\models \forall x_7 \, (x_7 = 0 \vee 0 < x_7) \end{aligned}$$

Was gilt für die Abkürzungen?

Satz

1. $\mathfrak{I} \models \varphi \vee \psi \quad \Longleftrightarrow \quad \mathfrak{I} \models \varphi$ oder $\mathfrak{I} \models \psi$.
2. $\mathfrak{I} \models \varphi \to \psi \quad \Longleftrightarrow \quad$ wenn $\mathfrak{I} \models \varphi$, dann $\mathfrak{I} \models \psi$.
3. $\mathfrak{I} \models \varphi \leftrightarrow \psi \quad \Longleftrightarrow \quad \mathfrak{I} \models \varphi$ genau dann, wenn $\mathfrak{I} \models \psi$.
4. $\mathfrak{I} \models \exists x \varphi \quad \Longleftrightarrow \quad$ es gibt ein $a \in A$ mit $\mathfrak{I}^a_a \models \varphi$.

Beweis von 4.

Satz

4. $\mathfrak{I} \models \exists x \varphi \quad \Longleftrightarrow \quad$ es gibt ein $a \in A$ mit $\mathfrak{I}^a_x \models \varphi$.

Beweis.

Nach Definition von $\models$ gelten die folgenden Äquivalenzen:

$$\begin{aligned}
\mathfrak{I} \models \exists x \varphi &\iff \mathfrak{I} \models \neg \forall x \, \neg \varphi \\
&\iff \mathfrak{I} \not\models \forall x \, \neg \varphi \\
&\iff \text{nicht für alle } a \in A \text{ gilt } \mathfrak{I}^a_x \models \neg \varphi \\
&\iff \text{es gibt ein } a \in A \text{ mit } \mathfrak{I}^a_x \not\models \neg \varphi \\
&\iff \text{es gibt ein } a \in A \text{ mit } \mathfrak{I}^a_x \models \varphi.
\end{aligned}$$

□

Erfüllbarkeit

Definition

1. Sei Φ eine Formelmenge. Dann schreiben wir $\mathfrak{I} \models \Phi$, falls $\mathfrak{I} \models \varphi$ für alle $\varphi \in \Phi$.
2. Eine σ-Formel φ heißt erfüllbar, falls es eine σ-Interpretation $\mathfrak{I}$ mit $\mathfrak{I} \models \varphi$ gibt.
   So eine Interpretation nennen wir auch ein σ-Modell für φ.
3. φ heißt allgemeingültig (oder logisch gültig, Tautologie), falls $\mathfrak{I} \models \varphi$ für alle σ-Interpretationen $\mathfrak{I}$.

Implikation und Äquivalenz

Definition

1. Für eine Menge Φ von σ-Formeln und eine Formel φ schreiben wir
   $$\Phi \models \varphi,$$
   falls für alle σ-Interpretationen $\mathfrak{I}$ mit $\mathfrak{I} \models \Phi$ auch $\mathfrak{I} \models \varphi$ gilt.
2. Für Formeln φ und ψ schreiben wir $\varphi \models \psi$, falls $\{\varphi\} \models \psi$.
3. Zwei σ-Formeln φ und ψ heißen logisch äquivalent, falls für alle σ-Interpretationen $\mathfrak{I}$ gilt:
   $$\mathfrak{I} \models \varphi \quad \Longleftrightarrow \quad \mathfrak{I} \models \psi,$$
   mit anderen Worten: $\varphi \models \psi$ und $\psi \models \varphi$.
   Wie in der Aussagenlogik benutzen wir die Notation $\varphi \equiv \psi$.

Beispiele

Sei wieder $\sigma = (<; +, \cdot; 0, 1)$. Dann gilt:

1. $\exists x\, x = 0$ ist allgemeingültig.
2. $0 = 1$ ist erfüllbar, aber nicht allgemeingültig.
3. $x < 1 + 1$ ist erfüllbar, aber nicht allgemeingültig.
4. $\exists z\, (x < y + z \wedge z = 1) \quad \equiv \quad x < y + 1$.
5. $\forall x \forall y\, (x < y \to \exists z\, (x < z \wedge z < y))$ ist erfüllbar, aber nicht allgemeingültig.

Das Lokalitätsprinzip

Der Wahrheitswert einer Formel hängt nur von der Bedeutung der vorkommenden Symbole ab.

Koinzidenzlemma

Sei $V \subseteq \mathsf{Var}$ und $\varphi \in \mathsf{Form}_\sigma$ mit $\mathsf{frei}(\varphi) \subseteq V$. Seien weiter $\mathfrak{I}_1 = (\mathfrak{A}_1, \beta_1)$ und $\mathfrak{I}_2 = (\mathfrak{A}_2, \beta_2)$ zwei σ-Modelle mit

- $\beta_1(x) = \beta_2(x)$ für alle $x \in V$ und
- $s^{\mathfrak{I}_1} = s^{\mathfrak{I}_2}$ für alle nichtlogischen Symbole $s \in \sigma$, die auch in φ vorkommen.

Dann gilt

$$\mathfrak{I}_1 \models \varphi \quad \Longleftrightarrow \quad \mathfrak{I}_2 \models \varphi \ .$$

Konvention

Gilt für alle Belegungen $\beta : \mathit{Var} \to A$, dass

$$(\mathfrak{A}, \beta) \models \varphi \quad ,$$

so schreiben wir einfach

$$\mathfrak{A} \models \varphi \quad .$$

Speziell:

Ist φ eine Aussage ($\mathsf{frei}(\varphi) = \emptyset$) und $(\mathfrak{A}, \beta) \models \varphi$, so ist also $\mathfrak{A} \models \varphi$.

Ein Beispiel

Für alle Strukturen $\mathfrak{A}$ mit $|A| \geq 2$ gilt

$$\mathfrak{A} \models \forall x \exists y \, \neg x = y \quad .$$

Beweis.

Sei $a \in A$ beliebig.
Sei weiter β eine beliebige Belegung und $\mathfrak{I} = (\mathfrak{A}, \beta)$.
Nach Voraussetzung existiert ein $b \in A$ mit $a \neq b$. Also gilt:

$$(\mathfrak{I}_x^a)_y^b \models \neg x = y$$

und damit

$$\mathfrak{I}_x^a \models \exists y \, \neg x = y \quad .$$

Weil a beliebig gewählt war, erhalten wir

$$\mathfrak{I} \models \forall x \exists y \, \neg x = y \quad .$$

$\square$

Wichtige Äquivalenzen und Normalformen

- Alle aussagenlogischen Äquivalenzen übertragen sich direkt in die Prädikatenlogik. So gilt zum Beispiel das de Morgansche Gesetz für beliebige σ-Formeln φ und ψ:

$$\neg(\varphi \wedge \psi) \equiv \neg\varphi \vee \neg\psi \ .$$

- Daneben gibt es auch Äquivalenzen, die mit Quantoren umgehen.

Wichtige Quantorenregeln

Sie σ eine beliebige Signatur und seien φ und ψ σ-Formeln.

1. $\neg\forall x\, \varphi \equiv \exists x\, \neg\varphi$
   $\neg\exists x\, \varphi \equiv \forall x\, \neg\varphi$
2. Falls $x \notin \mathsf{frei}(\psi)$, so gilt:
   $$\begin{aligned}(\forall x\, \varphi) \wedge \psi &\equiv \forall x\, (\varphi \wedge \psi)\\ (\forall x\, \varphi) \vee \psi &\equiv \forall x\, (\varphi \vee \psi)\\ (\exists x\, \varphi) \wedge \psi &\equiv \exists x\, (\varphi \wedge \psi)\\ (\exists x\, \varphi) \vee \psi &\equiv \exists x\, (\varphi \vee \psi)\end{aligned}$$
3. $\forall x\, \varphi \wedge \forall x\, \psi \equiv \forall x\, (\varphi \wedge \psi)$
   $\exists x\, \varphi \vee \exists x\, \psi \equiv \exists x\, (\varphi \vee \psi)$
4. $\forall x\, \forall y\, \varphi \equiv \forall y\, \forall x\, \varphi$
   $\exists x\, \exists y\, \varphi \equiv \exists y\, \exists x\, \varphi$

Ein Beweis

2. Falls $x \notin \mathsf{frei}(\psi)$, so gilt $\quad (\forall x\, \varphi) \wedge \psi \;\equiv\; \forall x\, (\varphi \wedge \psi)$

Beweis.
Sei $\mathfrak{I} \models (\forall x\, \varphi) \wedge \psi$ wobei $x \notin \mathsf{frei}(\psi)$.
Nach Definition ist dies äquivalent zu

$$\mathfrak{I} \models \forall x \varphi \quad \text{und} \quad \mathfrak{I} \models \psi \ .$$

Wiederum nach Definition ist der erste Teil äquivalent zu

$$\mathfrak{I}^a_x \models \varphi \quad \text{für alle } a \in A \tag{1}$$

und ebenso der zweite wegen des Koinzidenzlemmas (hier brauchen wir $x \notin \mathsf{frei}(\psi)$) zu

$$\mathfrak{I}^a_x \models \psi \quad \text{für alle } a \in A. \tag{2}$$

Beweis (Forts.)

Dann ist (1) und (2) äquivalent zu

$$\text{für alle } a \in A\text{:} \quad \mathfrak{I}^a_x \models \varphi \;\; \text{und} \;\; \mathfrak{I}^a_x \models \psi \ .$$

Nach Definition ist das äquivalent zu

$$\text{für alle } a \in A\text{:} \quad \mathfrak{I}^a_x \models \varphi \wedge \psi$$

und dies wiederum zu

$$\mathfrak{I} \models \forall x\, (\varphi \wedge \psi) \ .$$

$\square$

Vorsicht beim Umformen

Bei den Äquivalenzen in den Punkten 3 und 4 muss man genau aufpassen. Es gilt nämlich:

$$\begin{aligned} \forall x\, \varphi \lor \forall x\, \psi &\not\equiv \forall x\, (\varphi \lor \psi) \\ \exists x\, \varphi \land \exists x\, \psi &\not\equiv \exists x\, (\varphi \land \psi) \\ \forall x\, \exists y\, \varphi &\not\equiv \exists y\, \forall x\, \varphi\ . \end{aligned}$$

Pränexe Normalform

Definition

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 quantorenfreiem φ.

Die Formel φ nennt man auch die Matrix von $Q_1 x_1 \ldots Q_n x_n\, \varphi$.

Pränexe Normalform

Satz

Jede Formel ist äquivalent zu einer Formel in pränexer Normalform.

Zwei Umformungsregeln für den Beweis

1. Sei $\psi \triangleq Q_1 x_1 \dots Q_n x_n \, \theta$ mit $Q_1, \dots, Q_n \in \{\forall, \exists\}$ und sei

$$\overline{Q}_i = \begin{cases} \forall & \text{falls } Q_i = \exists \\ \exists & \text{falls } Q_i = \forall. \end{cases}$$

   Dann gilt $\quad \neg\psi \equiv \overline{Q}_1 x_1 \dots \overline{Q}_n x_n \, \neg\theta$.

2. Seien
   $\psi_1 \triangleq Q_1 x_1 \dots Q_k x_k \, \theta_1$ und
   $\psi_2 \triangleq Q'_1 y_1 \dots Q'_l y_l \, \theta_2$
   mit $Q_1, \dots, Q_k, Q'_1 \dots, Q'_l \in \{\forall, \exists\}$ sowie
   $x_1, \dots, x_k \notin \mathrm{frei}(\psi_2)$ und $y_1, \dots, y_l \notin \mathrm{frei}(\psi_1)$,
   $\circ \in \{\bigwedge, \bigvee\}$. Dann gilt

$$\psi_1 \circ \psi_2 \equiv Q_1 x_1 \dots Q_k x_k \, Q'_1 y_1 \dots Q'_l y_l \, \theta_1 \circ \theta_2 \ .$$

Gebundene Umbenennung

Wie stellt man die Voraussetzung für die Anwendung der vorherigen Umformungsregel her?

Lemma

Sei φ eine Formel und seien x, y zwei verschiedene Variablen, wobei y nicht in φ vorkommt. Dann gilt:

1. $\forall x\, \varphi \;\equiv\; \forall y\, (\varphi[x/y])$
2. $\exists x\, \varphi \;\equiv\; \exists y\, (\varphi[x/y])$

Beispiel zum Umformen in pränexe Normalform

Wir betrachten die Formel $\varphi(y)$ in der Signatur σ_{Gr}.

$$
\begin{aligned}
\varphi(y) \;&\triangleq\; \forall x\, \neg(\exists y\, E(x,y) \rightarrow \exists x\, E(x,y)) \\
&\equiv\; \forall x\, \neg(\neg\exists y\, E(x,y) \vee \exists x\, E(x,y)) \\
&\equiv\; \forall x\, \neg(\forall y\, \neg E(x,y) \vee \exists x\, E(x,y)) \\
&\equiv\; \forall x\, \neg(\forall z\, \neg E(x,z) \vee \exists w\, E(w,y)) \\
&\equiv\; \forall x\, \neg\forall z\, \exists w\, (\neg E(x,z) \vee E(w,y)) \\
&\equiv\; \forall x\, \exists z\, \forall w\, \neg(\neg E(x,z) \vee E(w,y)) \\
&\equiv\; \forall x\, \exists z\, \forall w\, (E(x,z) \wedge \neg E(w,y))
\end{aligned}
$$

Prädikatenlogisches Formalisieren

In diesem Abschnitt betrachten wir in einigen Beispielen, wie sich mathematische Sachverhalte in der Prädikatenlogik erster Stufe ausdrücken lassen.

- Im ersten Beispiel betrachten wir gerichtete Graphen.
- Als Signatur wählen wir

  $$\sigma_G = (E),$$

  d.h. die Sprache der Graphen enthält nur ein einziges Relationssymbol E für die Kantenbeziehung.
- σ_G-Strukturen

  $$\mathfrak{G} = (V; E^{\mathfrak{G}})$$

  enthalten als Träger eine Menge V von Knoten
  und eine binäre Relation auf V (die Interpretation $E^{\mathfrak{G}}$), die die Kantenbeziehung angibt.

Formalisieren in Graphen

Wir formulieren folgende Sachverhalte über Graphen als σ_G-Formeln:

1. $\mathfrak{G}$ ist ungerichtet:

   $$\varphi_{\text{ung}} \triangleq \forall x \, \forall y \, E(x, y) \to E(y, x)$$

   Dann gilt:

   $$\mathfrak{G} \models \varphi_{\text{ung}} \quad \text{genau dann, wenn } \mathfrak{G} \text{ ungerichtet ist.}$$

2. $\mathfrak{G}$ ist schleifenfrei:

   $$\forall x \, \neg E(x, x)$$

Formalisieren in Graphen

3. Zwischen den Knoten x und y existiert ein Weg der Länge 3:

$$\varphi_{\text{dist3}}(x,y) \triangleq \exists z_1\, \exists z_2\, (E(x, z_1) \land E(z_1, z_2) \land E(z_2, y))$$

4. Alle Knoten im Graphen können durch einen Weg der Länge 3 verbunden werden:

$$\begin{aligned} \varphi_{\text{dia3}} \quad &\triangleq \quad \forall x\, \forall y\, \exists z_1\, \exists z_2\, (E(x, z_1) \land E(z_1, z_2) \land E(z_2, y)) \\ &\triangleq \quad \forall x\, \forall y\, \varphi_{\text{dist3}}(x,y) \end{aligned}$$

5. Es gibt keinen Kreis der Länge 4:

$$\neg\, \exists x\, \exists z_1\, \exists z_2\, \exists z_3\, (E(x, z_1) \land E(z_1, z_2) \land E(z_2, z_3) \land E(z_3, x))$$

Grenzen der Ausdrucksfähigkeit

Ohne Beweis geben wir den folgenden Satz an, der die Grenze der Ausdrucksfähigkeit der Logik erster Stufe markiert:

Satz

Es gibt keine σ_G-Sätze φ, ψ, so dass für alle Graphen $\mathfrak{G} = (V, E^{\mathfrak{G}})$ gilt:

1. $\mathfrak{G} \models \varphi$ genau dann, wenn $\mathfrak{G}$ zusammenhängend ist.
2. $\mathfrak{G} \models \psi$ genau dann, wenn $\mathfrak{G}$ azyklisch (kreisfrei) ist.

Zweites Beispiel: Arithmetik

- Als zweites Beispiel betrachten wir die Arithmetik in der Sprache
$$\sigma_{\mathrm{Ar}} = (<; +, \cdot; 0, 1).$$
- Das Standardmodell ist die Struktur $\mathfrak{N}$ der natürlichen Zahlen mit der üblichen Interpretation von $<, +, \cdot, 0, 1$.

Arithmetisches Formalisieren

In σ_{Ar} formulieren wir folgende Sachverhalte:

1. x teilt y:
$$\varphi_{\text{teilt}}(x,y) \triangleq \exists z\ x \cdot z = y$$
2. Das Minuszeichen $-$ haben wir nicht in der Sprache σ_{Ar}. Wir können es aber definieren, und zwar wird die Gleichung $x - y = z$ beschrieben durch die σ_{Ar}-Formel:
$$y + z = x$$
3. Ebenso wird die Gleichung $x \equiv y \pmod z$ beschrieben durch
$$\exists w\,((x + w = y \vee y + w = x) \wedge \exists v\ z \cdot v = w)$$

Arithmetisches Formalisieren

4. x ist Primzahl:

$$\varphi_{\text{prim}}(x) \triangleq \neg x = 1 \wedge \forall y\, (\exists z\, y \cdot z = x \rightarrow y = 1 \vee y = x)$$

5. Es gibt unendlich viele Primzahlen:

$$\forall x\, \exists y\, (x < y \wedge \varphi_{\text{prim}}(y))$$

Arithmetisches Formalisieren

6. x ist Summe zweier Primzahlen:

$$\varphi_{\text{s2prim}}(x) \triangleq \exists z_1 \exists z_2\, (\varphi_{\text{prim}}(z_1) \wedge \varphi_{\text{prim}}(z_2) \wedge x = z_1 + z_2)$$

7. Es gibt eine gerade Zahl größer als 2, die sich als Summe zweier Primzahlen schreiben lässt:

$$\exists x((x > 1 + 1 \wedge \varphi_{\text{teilt}}(1 + 1, x)) \wedge \varphi_{\text{s2prim}}(x))$$

8. Jede gerade Zahl größer als 2 lässt sich als Summe zweier Primzahlen schreiben:

$$\varphi_{\text{Goldbach}} \triangleq \forall x((x > 1 + 1 \wedge \varphi_{\text{teilt}}(1 + 1, x)) \rightarrow \varphi_{\text{s2prim}}(x))$$

(Goldbach'sche Vermutung)

Relativierte Quantoren

Angenommen, wir wollen in den Aussagen

$$\exists x\, \psi(x) \qquad \text{und} \qquad \forall x\, \psi(x)$$

den Geltungsbereich der Quantoren einschränken auf solche x, die deine Eigenschaft ϕ haben:

- „Es gibt ein x mit der Eigenschaft ϕ, für das ψ gilt."
  Intuitiv: $(\exists x, \phi(x))\, \psi(x)$
  Formal: $\exists x\, (\phi(x) \wedge \psi(x))$
- „Für alle x mit der Eigenschaft ϕ gilt ψ."
  Intuitiv: $(\forall x, \phi(x))\, \psi(x)$
  Formal: $\forall x\, (\phi(x) \to \psi(x))$

Axiomensysteme

Für eine Menge von σ-Sätzen Φ sei

$$\mathsf{Mod}(\Phi) = \{\, \mathfrak{A} \mid \mathfrak{A} \text{ ist } \sigma\text{-Struktur und } \mathfrak{A} \models \Phi \,\}$$

$\mathsf{Mod}(\Phi)$ ist die Klasse aller Modelle von Φ.

Definition

Sei $\mathcal{C}$ eine Klasse von σ-Strukturen.
Φ axiomatisiert $\mathcal{C}$ (Φ ist Axiomensystem für $\mathcal{C}$), falls

$$\mathcal{C} = \mathsf{Mod}(\Phi) \ .$$

Beispiele für Axiomensysteme

1. Sei $\sigma = (\circ, ^{-1}; e)$.
   Die Menge Φ_{Gruppe} enthält die folgenden drei Aussagen:

$$\begin{aligned} \forall x\, \forall y\, \forall z \quad & (x \circ y) \circ z = x \circ (y \circ z) \\ \forall x \quad & x \circ e = x \\ \forall x \quad & x \circ x^{-1} = e \end{aligned}$$

   Dann axiomatisiert Φ_{Gruppe} die Klasse aller Gruppen, d.h.

$$\mathsf{Mod}(\Phi_{\text{Gruppe}}) = \{\, G \mid G \text{ ist Gruppe} \,\}.$$

Beispiele für Axiomensysteme

2. Die minimale Arithmetik Q ist durch folgendes Axiomensystem gegeben:

   (Q1) $\forall x\, \neg 0 = x + 1$

   (Q2) $\forall x \forall y\, x + 1 = y + 1 \rightarrow x = y$

   (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) $\forall x \forall y\, x < y \vee x = y \vee y < x$

Beispiele für Axiomensysteme

3. Die Peano-Arithmetik PA enthält zusätzlich folgendes Axiomenschema für jede Formel $\varphi(x, y_1, \ldots, y_n) \in \mathsf{Form}_{\sigma_{Ar}}$:

$$\forall y_1 \ldots \forall y_n \; \Big(\varphi(0, y_1, \ldots, y_k) \wedge \forall x \, (\varphi(x, y_1, \ldots, y_k) \to \varphi(x+1, y_1, \ldots, y_k)) \Big) \to \forall x \, \varphi(x, y_1, \ldots, y_k)$$

   PA hat neben $\mathfrak{N}$ weitere Modelle, die sog. Nichtstandardmodelle der Arithmetik.

4. Die natürlichen Zahlen sind nicht axiomatisierbar, d.h. es gibt keine Menge von Formeln $\Phi_{\mathbb{N}}$ mit $\mathsf{Mod}(\Phi_{\mathbb{N}}) = \{\mathfrak{N}\}$.

Beispiele für Axiomensysteme

5. Die Peano-Axiomensystem enthält zusätzlich zu Q folgende Formel der Prädikatenlogik der zweiten Stufe:

$$\textbf{(P)} \quad \forall X \big(X(0) \wedge \forall x \, (X(x) \to X(x+1)) \big) \to \forall x \, X(x)$$

   X: Variable zweiter Stufe, steht für Prädikate oder Mengen

   Satz von Dedekind
   Jede Struktur, die (Q1) – (Q9) und (P) erfüllt, ist isomorph zu $\mathfrak{N}$.

Beispiele für Axiomensysteme

6. Die Menge der endlichen kreisfreien Graphen ist zwar axiomatisierbar, aber nicht endlich axiomatisierbar, d.h. es gibt keine endliche Menge Ψ von σ_G-Sätzen mit

$$\mathrm{Mod}(\Psi) = \{\, \mathfrak{G} \mid \mathfrak{G} \text{ ist ein kreisfreier Graph} \,\}.$$

Abschluss unter Isomorphie

Satz

Sei φ ein σ-Satz und seien $\mathfrak{A}$ und $\mathfrak{B}$ isomorphe σ-Strukturen. Dann gilt

$$\mathfrak{A} \models \varphi \quad \Longleftrightarrow \quad \mathfrak{B} \models \varphi \ .$$

Als Folgerung ergibt sich unmittelbar:

Korollar

Sei Φ eine Menge von σ-Sätzen. Dann ist $\mathrm{Mod}(\Phi)$ unter Isomorphie abgeschlossen.

Definition

Seien σ eine Signatur, $\varphi, \varphi' \in \mathsf{Form}_\sigma$, $\Phi \subseteq \mathsf{Form}_\sigma$ und $\mathfrak{I}$ eine σ-Interpretation.

- $\mathfrak{I} \models \Phi$ falls $\mathfrak{I} \models \varphi$ für alle $\varphi \in \Phi$.
- φ heißt gültig in $\mathfrak{I}$ falls $\mathfrak{I} \models \varphi$.
- φ heißt wahr oder allgemeingültig, falls $\mathfrak{I} \models \varphi$ für alle $\mathfrak{I}$.
- φ heißt erfüllbar, falls es ein $\mathfrak{I}$ gibt mit $\mathfrak{I} \models \varphi$.
- φ folgt aus Φ (kurz: $\Phi \models \varphi$), falls $\mathfrak{I} \models \Phi \Rightarrow \mathfrak{I} \models \varphi$ für alle $\mathfrak{I}$.
- $\Phi^{\models} = \{\, \varphi \mid \Phi \models \varphi \,\}$ (Folgerungsoperator).

Eigenschaften

1. φ ist wahr $\quad\Leftrightarrow\quad \emptyset \models \varphi$.
2. $(\varphi_1 \to \varphi_2)$ ist wahr $\quad\Leftrightarrow\quad \{\varphi_1\} \models \varphi_2$.
3. $\{\varphi_1, \varphi_2, \ldots, \varphi_m\}^{\models} = \{\varphi_1 \wedge \varphi_2 \wedge \ldots \wedge \varphi_m\}^{\models}$.
4. φ ist wahr $\quad\Leftrightarrow\quad \forall x \varphi$ ist wahr.

Schlussregeln für die Prädikatenlogik

Wir betrachten nur Formeln über $\neg, \wedge, \vee, \forall$.

Wir erweitern den Kalkül des natürlichen Schließens, den wir für die Aussagenlogik kennengelernt haben, um drei weitere Regeln, die sich mit der Gleichheit und dem Existenzquantor befassen.

Schlussregeln I: Aussagenlogischer Teil

- $\varphi \in \Phi \Longrightarrow \Phi \vdash \varphi$
- Fallunterscheidung:
  $(\Phi \cup \{\varphi'\}) \vdash \varphi,\ (\Phi \cup \{\neg\varphi'\}) \vdash \varphi \Longrightarrow \Phi \vdash \varphi$
- Indirekter Beweis:
  $(\Phi \cup \{\neg\varphi\}) \vdash \varphi',\ (\Phi \cup \{\neg\varphi\}) \vdash \neg\varphi' \Longrightarrow \Phi \vdash \varphi$
- Abtrennungsregel, modus ponens:
  $\Phi \vdash (\neg\varphi' \vee \varphi), \Phi \vdash \varphi' \Longrightarrow \Phi \vdash \varphi$
  (Beache: $(\neg\varphi' \vee \varphi) \equiv (\varphi' \to \varphi)$.)
- Einführung der Alternative:
  $\Phi \vdash \varphi \Longrightarrow \Phi \vdash (\varphi \vee \varphi'), \Phi \vdash (\varphi' \vee \varphi)$ für beliebiges φ'
- Einführung der Konjunktion:
  $\Phi \vdash \varphi,\ \Phi \vdash \varphi' \Longrightarrow \Phi \vdash (\varphi \wedge \varphi')$
- Auflösung der Konjunktion:
  $\Phi \vdash (\varphi \wedge \varphi') \Longrightarrow \Phi \vdash \varphi,\ \Phi \vdash \varphi'$

Schlussregeln II: Prädikatenlogischer Teil

- Gleichheitsregel:
  $\Phi \vdash \varphi(x/t) \Longrightarrow \Phi \cup \{t = t'\} \vdash \varphi(x/t')$
- Einführung von $\forall$:
  $\Phi \vdash \varphi(x) \Longrightarrow \Phi \vdash \forall x \varphi$
- Auflösung von $\forall$:
  $\Phi \vdash \forall x \varphi \Longrightarrow \Phi \vdash \varphi(x/t)$,
  falls x nicht frei in t vorkommt.

$\Phi^{\vdash} = \{\, \varphi \mid \Phi \vdash \varphi \,\}$ (Ableitungsoperator).

Schlussfolgerungen

Schlussfolgerung:

$$\begin{array}{l} \quad\text{Alle Menschen sind sterblich.} \\ \quad\text{Sokrates ist ein Mensch.} \\ \hline \therefore \text{Sokrates ist sterblich.} \end{array}$$

Formalisierung:

$$\begin{array}{l} \quad\forall x(H(x) \to M(x)) \\ \quad H(\text{Sokrates}) \\ \hline \therefore M(\text{Sokrates}) \end{array}$$

mittels: Auflösung von $\forall$, modus ponens.

Endlichkeits-, Korrektheits- und Vollständigkeitssatz

Satz

Falls $\Phi \vdash \varphi$, so gibt es ein endliches $\Phi_0 \subseteq \Phi$ sodass $\Phi_0 \vdash \varphi$.

Satz (Kurt Gödel, 1929)

$\Phi \vdash \varphi$ gdw. $\Phi \models \varphi$.

Syntax	**Semantik**
Ableitung/Beweis/Schluss	Folgerung/Implikation
$\Phi \vdash \varphi$	$\Phi \models \varphi$
Φ ist konsistent	Φ ist erfüllbar (Φ hat ein Modell)
Φ ist inkonsistent	Φ ist unerfüllbar (Φ hat kein Modell)

Kompaktheitssatz

Aus dem Endlichkeitssatz und dem Vollständigkeitssatz ergibt sich direkt ein Endlichkeitssatz für die Folgerungsbeziehung:

Satz

$\Phi \models \varphi$ genau dann, wenn es eine endliche Menge $\Phi_0 \subseteq \Phi$ gibt, sodass $\Phi_0 \models \varphi$.

Anwendung des Endlichkeitssatzes

Satz

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 $\mathfrak{I} \models \Phi$.

Axiomensysteme mit beliebig großen Modellen haben auch ein unendliches Modell.

Beweis

Sei $\varphi_{\geq n} \triangleq \exists x_1 \exists x_2 \exists x_3 \ldots \exists x_n \bigwedge_{1 \leq i < j \leq n} x_i \neq x_j$.
Jedes Modell $\mathfrak{I}$ mit $\mathfrak{I} \models \varphi_{\geq n}$ besitzt mindestens n Elemente.

Sei $\Psi = \Phi \cup \{ \varphi_{\geq n} \mid 2 \leq n \}$.
Sei $\Psi_0 \subseteq \Psi$ endlich. Dann

$$\Psi_0 \subseteq \Phi \cup \{ \varphi_{\geq n} \mid 2 \leq n \leq n_0 \} \text{ für ein } n_0 \in \mathbb{N}.$$

Es folgt: $\mathfrak{I}_{n_0} \models \Psi_0$.
Also ist jede endliche Teilmenge von Ψ erfüllbar.

Endlichkeitssatz: Ψ ist erfüllbar. Sei $\mathfrak{I} \models \Psi$.

Es gilt:

- $\mathfrak{I} \models \Phi$.
- $\mathfrak{I}$ besitzt eine unendliche Trägermenge.

Folgerungen

1. Die Klasse der endlichen Gruppen ist nicht axiomatisierbar.

2. Die Klasse der endlichen σ-Strukturen ist nicht axiomatisierbar.

Folgerungen

Die Klasse der azyklischen Graphen ist axiomatisierbar über σ_{Gr} wie folgt:

$$\chi_k \triangleq \neg\exists x_1\, \exists x_2\, \exists x_3 \ldots \exists x_k\, \big(E(x_1, x_2) \wedge E(x_2, x_3) \wedge \ldots \wedge E(x_{k-1}, x_k) \wedge E(x_k, x_1)\big) \quad \text{für } k \geq 1.$$

Dann gilt:

- $\mathrm{Mod}(\{\chi_k\})$ ist die Menge der Graphen ohne einen Kreis der Länge k.
- $\mathrm{Mod}(\{\chi_k \mid k \geq 1\})$ ist die Menge der kreisfreien Graphen.

Folgerungen

Satz

Die Menge der azyklischen Graphen ist nicht endlich axiomatisierbar.

Beweis durch Widerspruch:

Sei $\mathrm{Mod}(\{\varphi\})$ die Menge der kreisfreien Graphen.

Dann gilt: $\{\chi_k \mid k \geq 1\} \models \varphi$.

Endlichkeitssatz:
Es gibt ein $n_0 \in \mathbb{N}$, sodass $\{\chi_k \mid 1 \leq k \leq n_0\} \models \varphi$.

Also gilt φ in allen Graphen, die keine Kreise der Länge $\leq n_0$ haben. Widerspruch!

Inhalt