Logik

Jürgen Koslowski und Roland Meyer

TU Braunschweig

SoSe 2018

Inhalt

4. Grundlagen der Prädikatenlogik
   - Syntax
   - Semantik
   - Substitution
   - Normalformen
   - Herbrand-Theorie
   - Semi-Entscheidbarkeit der Allgemeingültigkeit
   - Untere Schranke für Allgemeingültigkeit
   - Kompaktheitssatz der Prädikatenlogik erster Stufe
5. Deduktiver Aufbau der Prädikatenlogik
   - Logische Folgerung
   - Das deduktive System $\mathcal{F}$
   - Theorien erster Stufe
   - Axiomatisierung
6. Algorithmischer Aufbau der Prädikatenlogik
   - Semantische Tableaus
   - Unifikation
   - Resolution

Einleitung

Methoden zur Lösung von Problemen mit Hilfe von Rechnern
Formalisierung

- **Logik:** Lehre vom folgerichtigen Schließen bzw. Lehre von formalen Beziehungen zwischen Denkinhalten

Zentrale Fragen: Wahrheit und Beweisbarkeit von Aussagen $\rightsquigarrow$ **Mathematische Logik**.

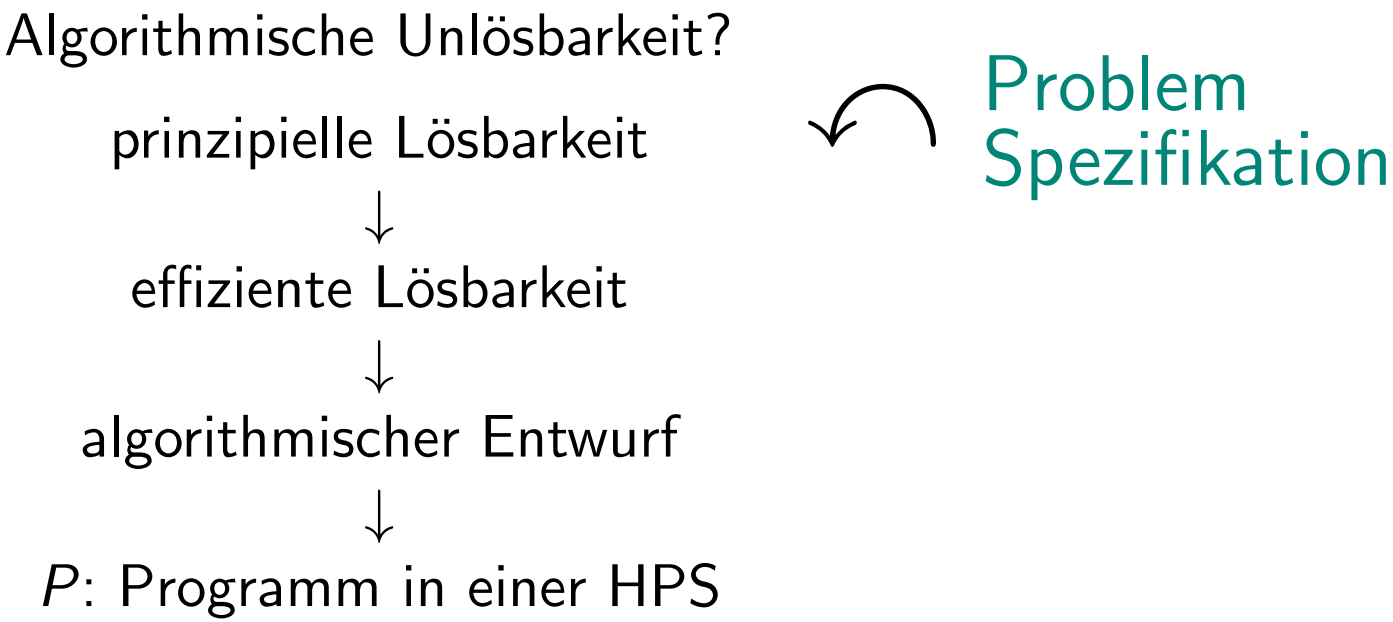
- **Logik in der Informatik**:
  - **Aussagenlogik**: Boolesche Algebra. Logische Schaltkreise (Kontrollsystemen), Schaltungen, Optimierung. SAT ist überall.
  - **Prädikatenlogik**: Schließen über Dateninhalte (KI, IS, SE).
  - **Modal- und Temporallogik**: Spezifikation und Verifikation (Hardware, seit 2000 Software).

Logik in der Informatik

1. Semantik von Programmiersprachen (Hoarscher Kalkül).
2. Spezifikation von funktionalen Eigenschaften.
3. Verifikationsprozess bei der SW-Entwicklung.
   Beweise von Programmeigenschaften.
4. Repräsentation von Daten. (Predicate Abstraction).
5. Spezielle Programmiersprachen (PROLOG)

- **Automatisierung des logischen Schließens**
  1. Automatisches Beweisen (Verfahren,...)
  2. Grundlagen von Informationsystemen (Verarbeitung von Wissen, Reasoning,. . .)

Berechnungsmodelle/Programmiersprachen



Syntaktische und semantische Verifikation

- **Syntaxanalyse**
  Sprachen Chomski-Hierarchie
  Kontextfreie Sprachen
  Grammatiken/Erzeugungsprozesse

- **Programmverifikation**
  Tut P auch, was erwartet wird?
  (Anforderungs)spezifikation und (Programm)verifikation.

Typische Ausdrücke

- $(x+1)(y-2)/5$ Terme als Bezeichner von Objekten.
- $3+2=5$ Gleichungen als spezielle Formeln.
- „29 ist (k)eine Primzahl “ Aussage.
- „$3+2=5$ und 29 ist keine Primzahl “ Aussage.
- „wenn 29 keine Primzahl ist, dann ist $0=1$ “ Aussage.
- „jede gerade Zahl, die größer als 2 ist, ist die Summe zweier Primzahlen “ Aussage.
- $2 \leq x$ und $(\forall y \in \mathbb{N})$
  $((2 \leq y \text{ und } y+1 \leq x) \rightarrow \text{nicht}(\exists z \in \mathbb{N}) y * z = x)$ Aussage.

Typische Ausdrücke (Fort.)

- $(\forall X \subseteq \mathbb{N})(0 \in X \wedge (\forall x \in \mathbb{N})(x \in X \to x + 1 \in X) \to X = \mathbb{N})$
  Induktionsprinzip.
- $(\forall X \subseteq \mathbb{N})(X \neq \emptyset \to X$ hat ein kleinstes Element)
  Jede nichtleere Menge natürlicher Zahlen enthält ein minimales Element.

Zweiwertige Logik Jede Aussage ist entweder **wahr** oder **falsch**.

- Es gibt auch andere Möglichkeiten (Mehrwertige Logik).
- Prädikatenlogik erster Stufe (PL1): Nur Eigenschaften von Elementen und Quantifizierung von Elementvariablen erlaubt.

Teil I

Aussagenlogik

Aussagenlogik

- Aufbau von Aussagen $\rightsquigarrow$ **Syntax**

- Bedeutung von Aussagen $\rightsquigarrow$ **Semantik** **wahr** (1), **falsch** (0)

Syntax der Aussagenlogik

Definition 1.1 (Syntax)

Betrachte das Alphabet $\Sigma = V \cup O \cup K$ mit

$V = \{p_1, p_2, ...\}$ einer abzählbaren Menge von **Aussagevariablen**,

$O = \{\neg/_1, \wedge/_2, \vee/_2, \rightarrow/_2, \leftrightarrow/_2\}$ **Verknüpfungen** mit Stelligkeiten (Junktoren),

$K = \{(,)\}$ **Klammern** (Hilfssymbole).

Die Menge der **Aussageformen** (Formeln der Aussagenlogik) $F \subseteq \Sigma^*$ ist induktiv definiert durch:

1. $V \subseteq F$ Menge der **atomaren Aussagen**
2. Falls $A, B \in F$ dann $(\neg A), (A \wedge B), (A \vee B), (A \rightarrow B), (A \leftrightarrow B) \in F$.

Induktive Definition nutzen implizit den Hüllenoperator: F ist die kleinste Menge, die V enthält und 2. erfüllt. Dieser Zusatz wird oft weggelassen.

Vereinbarungen: Abkürzungen und Prioritäten

- Beispiele für aussagenlogische Formeln sind

$$p_1, p_{101}, (((p_1 \to p_2) \wedge (\neg p_2)) \to (\neg p_1)), (p_1 \vee (\neg p_1))$$

- Äußere Klammern weglassen.
- Zur besseren Lesbarkeit: Prioritäten: $\neg, \wedge, \vee, \to, \leftrightarrow$

$$\begin{array}{rll} A \wedge B \to C & \text{steht für} & ((A \wedge B) \to C) \\ A \vee B \wedge C & \text{steht für} & (A \vee (B \wedge C)) \\ \neg A \vee B \wedge C & \text{steht für} & ((\neg A) \vee (B \wedge C)) \\ A \vee B \vee C & \text{steht für} & ((A \vee B) \vee C) \quad \text{(Linksklammerung).} \end{array}$$

Semantik

Definition 1.2 (Bewertung)

Eine **Bewertung** der aussagenlogischen Formeln ist eine Funktion $\varphi : F \to \mathbb{B} := \{0, 1\}$, so dass Folgendes gilt

$$\varphi(\neg A) = 1 - \varphi(A)$$
$$\varphi(A \vee B) = max(\varphi(A), \varphi(B))$$
$$\varphi(A \wedge B) = min(\varphi(A), \varphi(B))$$
$$\varphi(A \to B) = \begin{cases} 0 & \text{falls } \varphi(A) = 1 \text{ und } \varphi(B) = 0 \\ 1 & \text{sonst} \end{cases}$$
$$\varphi(A \leftrightarrow B) = \begin{cases} 0 & \text{falls } \varphi(A) \neq \varphi(B) \\ 1 & \text{falls } \varphi(A) = \varphi(B) \end{cases}$$

Belegungen und Bewertungen (Fort.)

- Sprechweise: A ist **falsch** unter φ, falls $\varphi(A) = 0$
  A ist **wahr** unter φ oder φ **erfüllt** A, falls $\varphi(A) = 1$.
- Darstellung von Bewertungen durch **Wahrheitstafeln**:

A	$\neg A$
1	0
0	1

A	B	$A \vee B$	$A \wedge B$	$A \to B$	$A \leftrightarrow B$
0	0	0	0	1	1
0	1	1	0	1	0
1	0	1	0	0	0
1	1	1	1	1	1

Belegungen und Bewertungen (Fort.)

- Eine **Belegung** der Variablen V ist eine Funktion $\psi : V \to \mathbb{B}$.
- Jede Bewertung induziert eine eindeutige Belegung: $\psi(p_i) := \varphi(p_i)$.

Lemma 1.3

Jede Belegung $\psi : V \to \mathbb{B}$ lasst sich auf genau eine Weise zu einer Bewertung $\varphi : F \to \mathbb{B}$ fortsetzen. Insbesondere wird jede Bewertung durch die Werte auf V eindeutig festgelegt.

Belegungen und Bewertungen (Fort.)

Folgerung 1.4

Die Bewertung einer Aussageform $A \in F$ hängt nur von den Werten der in ihr vorkommenden Aussagevariablen aus V ab. Das heißt, will man $\varphi(A)$ berechnen, genügt es, die Werte $\varphi(p)$ zu kennen für alle Aussagevariablen p, die in A vorkommen.

- Beispiel: Sei $\varphi(p) = 1, \varphi(q) = 1, \varphi(r) = 0$. Dann kann $\varphi(A)$ iterativ berechnet werden:

$$A \equiv \underbrace{\underbrace{\underbrace{(\underbrace{p}_{1} \to \underbrace{(\underbrace{q}_{1} \to \underbrace{r}_{0})}_{0}}_{0} \to \underbrace{(\underbrace{\underbrace{p}_{1} \wedge \underbrace{q}_{1}}_{1} \to \underbrace{r}_{0})}_{0})}_{1}}_{1}$$

Also gilt $\varphi(A) = 1$.

Belegungen und Bewertungen (Fort.)

Welche Werte nimmt $\varphi(A)$ an, wenn φ **alle** Belegungen durchläuft?

A definiert eine Boolesche Funktion $f_A : \mathbb{B}^n \to \mathbb{B}$.

- Ist etwa $\varphi(A) = 1$ für alle Belegungen φ?
- Es genügt, die **endlich** vielen Belegungen der Variablen, die in A vorkommen, zu prüfen.
- Kommen n Variablen in A vor, so gibt es 2^n verschiedene Belegungen.
- Beispiel: Für die drei Variablen p, q und r aus A im obigen Beispiel gibt es 8 Belegungen, die betrachtet werden müssen.

Belegungen und Bewertungen (Fort.)

p	q	r	$q \to r$	$p \wedge q$	$p \to (q \to r)$	$(p \wedge q) \to r$	A
0	0	0	1	0	1	1	1
0	0	1	1	0	1	1	1
0	1	0	0	0	1	1	1
0	1	1	1	0	1	1	1
1	0	0	1	0	1	1	1
1	0	1	1	0	1	1	1
1	1	0	0	1	0	0	1
1	1	1	1	1	1	1	1

A ist wahr unabhängig von den Werten von p, q, r, d.h. für jede Bewertung φ. Weitere solche Formeln sind etwa:
$(A \to (B \to A))$, $(A \to (B \to C)) \to ((A \to B) \to (A \to C))$ oder
$((\neg A \to \neg B) \to (B \to A))$.

Wichtige Begriffe

Definition 1.5

Sei $A \in F$, $\Sigma \subseteq F$.

1.(a) A heißt **Tautologie** (**allgemeingültig**), falls $\varphi(A) = 1$ für jede Bewertung φ gilt. (Schreibweise $\models A$)

(b) A ist **erfüllbar**, falls es eine Bewertung φ gibt, mit $\varphi(A) = 1$.

(c) A ist **widerspruchsvoll**, falls $\varphi(A) = 0$ für jede Bewertung φ.

(d) TAUT $:= \{A \in F \mid A \text{ ist Tautologie}\}$ die **Menge der Tautologien**.

(e) SAT $:= \{A \in F \mid A \text{ ist erfüllbar}\}$ die **Menge der erfüllbaren Formeln**. Beachte TAUT $\subseteq$ SAT.

Definition (Fort.)

2.(a) Σ ist **erfüllbar**, falls es eine Bewertung φ gibt mit $\varphi(A) = 1$ für alle $A \in \Sigma$. (φ **erfüllt** Σ)

(b) **Semantischer Folgerungsbegriff:** A ist **logische Folgerung** von Σ, falls $\varphi(A) = 1$ für jede Bewertung φ, die Σ erfüllt.

Man schreibt $\Sigma \models A$. Auch $A_1, \ldots, A_n \models A$, falls $\Sigma = \{A_1, \ldots, A_n\}$.

(c) Die Menge Folg(Σ) der Folgerungen aus Σ ist definiert durch:

$$\text{Folg}(\Sigma) := \{A \in F \mid \Sigma \models A\}.$$

Beispiele

Beispiel 1.6

1. $(p \vee (\neg p))$, $((p \to q) \vee (q \to r))$, $p \to (q \to p)$, $(p \to p)$, $(p \to \neg\neg p)$ und A aus Folgerung 1.4 sind Tautologien.
2. $(p \wedge (\neg p))$ ist widerspruchsvoll.
3. $(p \wedge q)$ ist erfüllbar aber weder Tautologie noch Widerspruch.
4. Sei $\Sigma = \{p\}$ und $A = p \vee q$. Dann gilt $\Sigma \models A$, denn falls $\varphi(p) = 1$, dann auch $\varphi(p \vee q) = 1$. Jede Bewertung, die Σ erfüllt, erfüllt also auch A.

Folgerungen

Lemma 1.7

(a) *A allgemeingültig gdw $\neg A$ widerspruchsvoll.*

(b) *Es gilt $\emptyset \models A$ genau dann, wenn A Tautologie ist: $Folg(\emptyset) = TAUT$.*

(c) *Ist Σ nicht erfüllbar, dann gilt $\Sigma \models A$ für alle $A \in F$: $Folg(\Sigma) = F$. Insbesondere $\Sigma \models A$ und $\Sigma \models \neg A$ für ein $A \in F$.*

(d) *Sei $\Sigma \subseteq \Sigma'$. Ist Σ' erfüllbar, dann ist auch Σ erfüllbar.*

(e) *Es gilt $\Sigma \subseteq Folg(\Sigma)$ und $Folg(Folg(\Sigma)) = Folg(\Sigma)$.*

(f) *Falls $\Sigma \subseteq \Sigma'$, dann gilt $Folg(\Sigma) \subseteq Folg(\Sigma')$.*

(g) *$\Sigma \models A$ gilt genau dann, wenn $\Sigma \cup \{\neg A\}$ nicht erfüllbar.*

(h) *Ist Σ endlich, dann ist es entscheidbar, ob Σ erfüllbar ist, und die Menge $Folg(\Sigma)$ ist entscheidbar.*

(i) *Die Mengen $TAUT, SAT$ sind entscheidbar.*

Deduktionstheorem und Modus-Ponens Regel

Lemma 1.8

a) **Deduktionstheorem** *(semantische Version):*

$$\Sigma, A \models B \qquad \text{gdw} \qquad \Sigma \models (A \to B).$$

(Σ, A ist Kurzschreibweise für $\Sigma \cup \{A\}$)

b) **Modus-Ponens-Regel**: *Es gilt*

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

Insbesondere ist B Tautologie, falls A und $(A \to B)$ Tautologien.

Kompaktheitssatz der Aussagenlogik

Satz 1.9 (Kompaktheitssatz, Beweis nach Lindenbaum)

$\Sigma \subseteq F$ ist erfüllbar genau dann, wenn jede **endliche** *Teilmenge von Σ erfüllbar ist.*
$\Sigma \subseteq F$ ist unerfüllbar genau dann, wenn es eine unerfüllbare **endliche** *Teilmenge von Σ gibt.*

Korollar 1.10

Es gilt $\Sigma \models A$ genau dann, wenn es eine **endliche** *Teilmenge $\Sigma_0 \subseteq \Sigma$ gibt mit $\Sigma_0 \models A$.*

- Der zweite Teil des Satzes ist die Grundlage für Beweisverfahren für $\Sigma \models A$. Dies ist der Fall, wenn $\Sigma \cup \{\neg A\}$ unerfüllbar ist.
- Widerspruchbeweise versuchen systematisch eine **endliche** Menge $\Sigma_0 \subseteq \Sigma$ zu finden, so dass $\Sigma_0 \cup \{\neg A\}$ unerfüllbar ist.

Anwendungen Kompaktheitssatz

Beispiel 1.11

Sei $\Sigma \subseteq F$. Gibt es zu jeder Bewertung φ ein $A \in \Sigma$ mit $\varphi(A) = 1$, so gibt es $A_1, ..., A_n \in \Sigma$ $(n > 0)$ mit $\models A_1 \vee ... \vee A_n$.

- Betrachte die Menge $\Sigma' = \{\neg A \in F \mid A \in \Sigma\}$.
  Nach Voraussetzung ist sie unerfüllbar.
  Also gibt es endliche nichtleere Teilmenge $\{\neg A_1, ..., \neg A_n\}$ von Σ', die unerfüllbar ist.
  Also gibt es für jede Bewertung φ ein i mit $\varphi(\neg A_i) = 0$.
  Also $\varphi(A_i) = 1$ und somit $\varphi(A_1 \vee ... \vee A_n) = 1$.

Logische Äquivalenz

Definition 1.12 (Logische Äquivalenz)

Formeln $A, B \in F$ heißen **logisch äquivalent**, $A \models\models B$, falls für jede Bewertung φ gilt: $\varphi(A) = \varphi(B)$.

Beispiele logisch äquivalenter Formeln:

(Involution) $A \models\models \neg(\neg A)$

(Idempotenz) $A \models\models A \wedge A$ $\qquad$ $A \models\models A \vee A$

(Kommutativität) $A \wedge B \models\models B \wedge A$ $\qquad$ $A \vee B \models\models B \vee A$

(Assoziativität) $A \wedge (B \wedge C) \models\models (A \wedge B) \wedge C$

$A \vee (B \vee C) \models\models (A \vee B) \vee C$

(Distributivität) $A \wedge (B \vee C) \models\models (A \wedge B) \vee (A \wedge C)$

$A \vee (B \wedge C) \models\models (A \vee B) \wedge (A \vee C)$

Logische Äquivalenz (Fort.)

(De Morgan) $\neg(A \wedge B) \models\!\!\models \neg A \vee \neg B$ $\quad$ $\neg(A \vee B) \models\!\!\models \neg A \wedge \neg B$

$A \to B \models\!\!\models \neg A \vee B$ $\quad$ $A \leftrightarrow B \models\!\!\models (A \to B) \wedge (B \to A)$

$A \wedge B \models\!\!\models \neg(A \to \neg B)$ $\quad$ $A \vee B \models\!\!\models \neg A \to B$

Lemma 1.13

Logische Äquivalenz $\models\!\!\models \subseteq F \times F$ ist eine Äquivalenzrelation, also reflexiv, symmetrisch und transitiv.

Es ist sogar eine Kongruenz: ersetzt man in einer Formel A eine Teilformel B durch $C \models\!\!\models B$, so erhält man $A' \models\!\!\models A$.

Logische Äquivalenz (Fort.)

Lemma 1.14

Folgende Aussagen sind äquivalent:

$$\models A \leftrightarrow B \qquad\qquad A \models\!\models B$$

$$A \models B \text{ und } B \models A \qquad\qquad Folg(A) = Folg(B)$$

Lemma 1.15

Zu jeder Formel $A \in F$ gibt es $B, C, D \in F$ mit

1. $A \models\!\models B$, *B enthält nur $\rightarrow$ und $\neg$ als Verknüpfungen*
2. $A \models\!\models C$, *C enthält nur $\wedge$ und $\neg$ als Verknüpfungen*
3. $A \models\!\models D$, *D enthält nur $\vee$ und $\neg$ als Verknüpfungen*

Folgt aus obigen Äquivalenzen.

Logische Äquivalenz (Fort.)

Definition 1.16 (Vollständige Operatorenmengen)

Eine Menge $OP \subseteq \{\neg, \vee, \wedge, \rightarrow, \leftrightarrow\}$ heißt **vollständig**, falls es zu jedem $A \in F$ eine logisch äquivalente Formel $B \in F(OP)$ gibt. Dabei ist $F(OP)$ die Menge der Formeln mit Verknüpfungen in OP.

- Vollständige Operatorenmengen für die Aussagenlogik sind z.B.: $\{\neg, \rightarrow\}, \{\neg, \vee\}, \{\neg, \wedge\}, \{\neg, \vee, \wedge\}, \{\text{false}, \rightarrow\}$.

  Dabei ist false eine Konstante mit $\varphi(\text{false}) = 0$ für jede Bewertung φ. Offenbar gilt $\neg A \models\!\models (A \rightarrow \text{false})$.
- **Normalformen**: DNF (Disjunktive Normalform), KNF (Konjunktive Normalform), KDNF, KKNF (Kanonische Formen).

Boolsche Funktionen

Jede Formel $A(p_1, \ldots, p_n)$ stellt eine Boolsche Funktion $f_A : \mathbb{B}^n \to \mathbb{B}$ dar, nämlich mit der Definition $f_A(b_1, ..., b_n) := \varphi_{\overline{b}}(A)$, wobei $\varphi_{\overline{b}}(p_i) := b_i$.

- Man kann zeigen, dass sich jede Boolsche Funktion $f : \mathbb{B}^n \to \mathbb{B}$ $(n > 0)$ in obiger Form durch eine Formel $A(p_1, \ldots, p_n)$ darstellen lässt, sofern die Operatormenge vollständig ist.
- Die Boolsche Algebra hat als übliche Operatormenge **true, false, not, or, and**.
- Für andere Operatormengen, die etwa **nand, nor** enthalten, siehe Digitale Logik. Dort werden nand, nor-Gatter bevorzugt, da sie nur zwei Transistoren benötigen.

Boolsche Funktionen: Beispiel

Ein Patientenüberwachungssystem erhält gewisse Daten über den Zustand eines Patienten: Temperatur, Blutdruck, Pulsrate. Die Schwellenwerte für die Daten seien wie folgt festgelegt:

Zustände Ein/Ausgaben	Bedeutung
A	Temperatur außerhalb 36-39°C.
B	Blutdruck außerhalb 80-160 mm.
C	Pulsrate außerhalb 60-120 Schläge pro Minute.
O	Alarmaktivierung ist notwendig.

Boolsche Funktionen: Beispiel (Fort.)

Die Anforderungen, d.h. bei welchen Kombinationen der Werte der Zustände eine Alarmaktivierung notwendig ist, werden durch den Medizin-Experten festgelegt. Sie seien in folgender Tabelle fixiert:

I/O - **Tabelle**

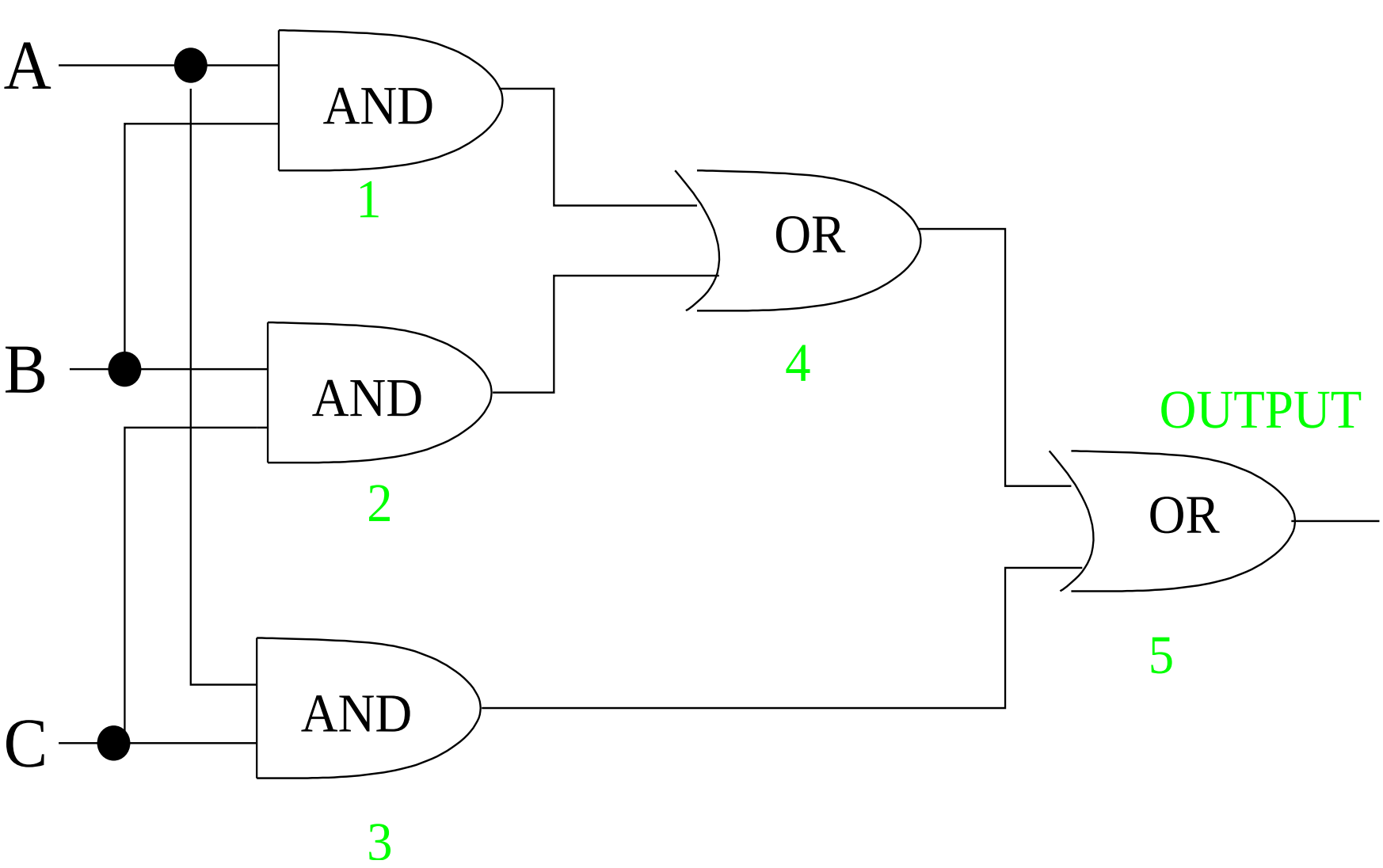
A	B	C	O
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

Logischer Entwurf: Betrachte die Zeilen in denen O den Wert 1 hat und stelle eine KDNF auf:

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

Boolsche Funktionen: Beispiel (Fort.)

Als eine Realisierung könnte man das folgende Schaltnetz nehmen:



Deduktiver Aufbau der Aussagenlogik

Dieser Abschnitt beschäftigt sich mit einem axiomatischen Aufbau der Aussagenlogik mittels eines **Deduktiven Systems** oder **Kalküls**.

Eine syntaktisch korrekte Formel in einem Deduktiven System wird Theorem genannt, wenn sie durch rein mechanische Anwendungen der Regeln des Systems aus den Axiomen des Systems abgeleitet werden kann.

Man kann deduktive Systeme angeben, in denen aussagenlogische Formeln genau dann Theoreme sind, wenn sie auch Tautologien sind.

Deduktive Systeme

Definition 2.1 (Deduktives System)

Ein **Deduktives System** $\mathcal{F}(\mathrm{Ax}, R)$ besteht aus

- einem Alphabet Δ,
- einer Menge von Formeln $F \subseteq \Delta^\star$,
- einer Menge von Axiomen $\mathrm{Ax} \subseteq F$ und
- einer Menge R von Regeln der Form $\frac{A_1, \ldots, A_n}{A}$ mit $n > 0$ und $A_1, ..., A_n, A \in F$.

Die Mengen F, Ax und R sind typischerweise **entscheidbar**.

Deduktive Systeme (Fort.)

Definition 2.2

Die Menge $T = T(\mathcal{F})$ der **Theoreme** ist induktiv definiert durch:

1. $Ax \subseteq T$ alle Axiome sind Theoreme
2. Sind $A_1, \ldots, A_n \in T$ und ist $\frac{A_1, \ldots, A_n}{A}$ in R, dann ist $A \in T$.

Schreibe $A \in T(\mathcal{F})$ als $\vdash_{\mathcal{F}} A$ oder $\vdash A$ und sage A ist in $\mathcal{F}$ herleitbar.

Deduktiver Folgerungsbegriff: Sei $\Sigma \subseteq F$, $A \in F$. Dann ist A in $\mathcal{F}$ aus Σ herleitbar, kurz $\Sigma \vdash_{\mathcal{F}(Ax,R)} A$, falls $\vdash_{\mathcal{F}(Ax \cup \Sigma,R)} A$ gilt. Auch:

$$Folg_{\mathcal{F}}(\Sigma) := \{A \in F \mid \Sigma \vdash_{\mathcal{F}(Ax,R)} A\}.$$

Σ heißt **konsistent**, falls für keine Formel $A \in F$ gilt: $\Sigma \vdash A$ und $\Sigma \vdash \neg A$. Gibt es eine solche Formel, so heißt Σ **inkonsistent**.

Beweise

Bemerkung 2.3

Formel A ist in $\mathcal{F}$ herleitbar, falls es eine endliche Folge von Formeln $B_1, \ldots, B_n$ gibt mit $A \equiv B_n$ und für $1 \leq i \leq n$ gilt:

$$B_i \in Ax \text{ oder es gibt } i_1, \ldots, i_l < i \text{ und } \frac{B_{i_1} \ldots B_{i_l}}{B_i} \in R.$$

Folge $B_1, \ldots, B_n$ heißt auch **Beweis** *für A in $\mathcal{F}$.*

Eine endliche Folge $B_1, \ldots, B_n$ heißt **abgekürzter Beweis** *für $\Sigma \vdash B_n$, falls für $1 \leq j \leq n$ gilt:*

$$\Sigma \vdash B_j \text{ oder es gibt } j_1, \ldots, j_r < j \text{ mit } B_{j_1}, \ldots, B_{j_r} \vdash B_j.$$

Lemma 2.4

1. $\vdash A$ *gilt gdw. es einen Beweis für A gibt.*
2. *Es gibt einen Beweis für $\Sigma \vdash A$ gdw. es einen abgekürzten Beweis für $\Sigma \vdash A$ gibt.*

Bemerkung

Bemerkung 2.5

- *Eigenschaften der Elemente von T werden durch strukturelle Induktion bewiesen.*
- *Die Menge der Beweise*

$$\text{Bew} := \{B_1, \ldots, B_n \in F^+ \mid B_1, \ldots, B_n \text{ ist Beweis}\}$$

  ist entscheidbar.
- *Mit der vorherigen Bemerkung ist die Menge T der Theoreme rekursiv aufzählbar.*
- *Ist Σ entscheidbar, dann gelten die Aussagen entsprechend. Insbesondere ist $Folg_{\mathcal{F}}(\Sigma)$ aufzählbar.*

Bemerkung (Fort.)

Lemma 2.6

- *Gilt* $\Sigma \vdash A$, *so folgt aus der Definition des Ableitungsbegriffs, dass es eine endliche Teilmenge* $\Sigma_0 \subseteq \Sigma$ *gibt mit* $\Sigma_0 \vdash A$.
  (Dies entspricht dem Kompaktheitssatz für $\models$*.)*
- *Ist* Σ *inkonsistent, dann gibt es eine endliche Teilmenge* $\Sigma_0 \subseteq \Sigma$, *die inkonsistent ist.*
- *Ist* $\Sigma \subseteq \Gamma$, *dann gilt* $Folg_{\mathcal{F}}(\Sigma) \subseteq Folg_{\mathcal{F}}(\Gamma)$.
- *Aus* $\Sigma \vdash A$ *und* $\Gamma \vdash B$ *für alle* $B \in \Sigma$ *folgt* $\Gamma \vdash A$.
  Ist also $\Sigma \subseteq Folg_{\mathcal{F}}(\Gamma)$, *dann gilt* $Folg_{\mathcal{F}}(\Sigma) \subseteq Folg_{\mathcal{F}}(\Gamma)$.
  (Beweise lassen sich also zusammensetzen.)
- *Gilt* $\Sigma \vdash A$, *so ist* $\Sigma \cup \{\neg A\}$ *inkonsistent.*
  (Gilt auch die Umkehrung?)
- *Es gilt* $T(\mathcal{F}) \subseteq Folg_{\mathcal{F}}(\Sigma)$ *für jede Menge* Σ.

Schemata

Gibt es ein deduktives System $\mathcal{F}_0$, so dass

$$\vdash_{\mathcal{F}_0} A \quad \text{gdw.} \quad \models A?$$

Hierzu werden Ax und R häufig endlich beschrieben durch Schemata.
Beispielsweise beschreibt das Schema $A \to (B \to A)$ die Menge

$$\{A_0 \to (B_0 \to A_0) \quad | \quad A_0, B_0 \in F\}.$$

Das Schema $\dfrac{A, A \to B}{B}$ beschreibt die Menge von Regeln

$$\{\frac{A_0, A_0 \to B_0}{B_0} \quad | \quad A_0, B_0 \in F\}.$$

Das deduktive System $\mathcal{F}_0$

Eingeführt von Stephen Cole Kleene (1909 — 1994).

Definition 2.7 (Das deduktive System $\mathcal{F}_0$)

Das deduktive System $\mathcal{F}_0$ für die Aussagenlogik besteht aus der Formelmenge F_0 der Formeln in $V, \neg, \rightarrow, ($ und $)$. Die Axiomenmenge Ax wird durch folgende Axiomenschemata beschrieben:

Ax1: $A \rightarrow (B \rightarrow A)$

Ax2: $(A \rightarrow (B \rightarrow C)) \rightarrow ((A \rightarrow B) \rightarrow (A \rightarrow C))$

Ax3: $(\neg A \rightarrow \neg B) \rightarrow (B \rightarrow A)$

Die Regelmenge R wird beschrieben durch das Regelschema

$$\text{MP:} \quad \frac{A, (A \rightarrow B)}{B} \quad \text{(modus ponens).}$$

Bemerkung zum deduktiven System $\mathcal{F}_0$

- Ax1, Ax2 und Ax3 beschreiben disjunkte Formelmengen.
- Ax und R sind entscheidbar.
- Alle Axiome sind Tautologien. Da diese abgeschlossen gegen Modus Ponens sind, sind alle Theoreme Tautologien: $T(\mathcal{F}_0) \subseteq \mathit{Taut}(F_0)$.
- Modus-Ponens-Regel ist nicht eindeutig: $\frac{A, A \to B}{B}$ und $\frac{A', A' \to B}{B}$ haben gleiche Folgerung. Erschwert das Finden von Beweisen.
- Es genügt, nur Axiome für Formeln in $\to$ und $\neg$ zu betrachten. Andere Formeln sind zu einer solchen Formel logisch äquivalent.

  Will man in ganz F Beweise führen, braucht man weitere Axiome, zum Beispiel:

$$\text{Ax1}\wedge : (A \wedge B) \to \neg(A \to \neg B) \qquad \text{Ax2}\wedge : \neg(A \to \neg B) \to (A \wedge B)$$

Beispiel

Beispiel 2.8

Für jedes $A \in F_0$ gilt $\vdash (A \to A)$, also $(A \to A) \in T(\mathcal{F}_0)$

Beweis:

$$
\begin{array}{lll}
B_0 \equiv (A \to ((A \to A) \to A)) \to & & \\
\qquad ((A \to (A \to A)) \to (A \to A)) & & \text{Ax2} \\
B_1 \equiv A \to ((A \to A) \to A) & & \text{Ax1} \\
B_2 \equiv (A \to (A \to A)) \to (A \to A) & & \text{MP}(B_0, B_1) \\
B_3 \equiv A \to (A \to A) & & \text{Ax1} \\
B_4 \equiv A \to A & & \text{MP}(B_2, B_3)
\end{array}
$$

■

Deduktionstheorem

Wie findet man Beweise im System $\mathcal{F}_0$?

Einziger Hinweis: sofern Zielformel B kein Axiom ist, muss sie in der Form $(A_1 \to \ldots (A_n \to B) \ldots)$ vorkommen. Wähle geeignete A.

Hilfreich:

Satz 2.9 (Deduktionstheorem (syntaktische Version))

Seien $\Sigma \subseteq F_0$ *und* $A, B \in F_0$. *Dann gilt*

$$\Sigma, A \vdash B \quad \textit{gdw.} \quad \Sigma \vdash (A \to B).$$

Anwendungen des Deduktionstheorems

Beispiel 2.10

Um $\vdash \neg\neg A \to A$ zu zeigen, genügt es, $\neg\neg A \vdash A$ zu zeigen.

Beweis:

$B_1 \equiv \neg\neg A$	
$B_2 \equiv \neg\neg A \to (\neg\neg\neg\neg A \to \neg\neg A)$	Ax1
$B_3 \equiv \neg\neg\neg\neg A \to \neg\neg A$	MP
$B_4 \equiv (\neg\neg\neg\neg A \to \neg\neg A) \to (\neg A \to \neg\neg\neg A)$	Ax3 ■
$B_5 \equiv \neg A \to \neg\neg\neg A$	MP
$B_6 \equiv (\neg A \to \neg\neg\neg A) \to (\neg\neg A \to A)$	Ax3
$B_7 \equiv \neg\neg A \to A$	MP
$B_8 \equiv A$	MP

Anwendungen des Deduktionstheorems (Fort.)

Lemma 2.11

- *Die folgenden Theoreme gelten in* $\mathcal{F}_0$*:*

(Transitivität der Implikation)	$\vdash (A \to B) \to ((B \to C) \to (A \to C))$	(1)
(Folgerung aus Inkonsistenz)	$\vdash \neg B \to (B \to A)$	(2)
(Doppelnegation)	$\vdash B \to \neg\neg B$	(3)
(Kontraposition)	$\vdash (A \to B) \to (\neg B \to \neg A)$	(4)
(Implikation)	$\vdash B \to (\neg C \to \neg(B \to C))$	(5)
(Hilfslemma 1)	$\vdash (A \to B) \to ((A \to \neg B) \to (A \to \neg Ax))$	(E1)
(Hilfslemma 2)	$\vdash (A \to \neg Ax) \to \neg A$	(E2)
(Negation aus Inkonsistenz)	$\vdash (A \to B) \to ((A \to \neg B) \to \neg A)$	(6)
(Eliminierung von Annahmen)	$\vdash (B \to A) \to ((\neg B \to A) \to A)$	(7)

- *Es gilt* $\Sigma \vdash A$ *gdw.* $\Sigma \cup \{\neg A\}$ *inkonsistent ist.*

Korrektheit und Vollständigkeit von $\mathcal{F}_0$

Frage: Lassen sich alle Tautologien als Theoreme in System $\mathcal{F}_0$ herleiten?

Satz 2.12 (Korrektheit und Vollständigkeit von $\mathcal{F}_0$)

Sei $A \in F_0$ eine Formel der Aussagenlogik.

a) Korrektheit: *Aus $\vdash_{\mathcal{F}_0} A$ folgt $\models A$, es können nur Tautologien als Theoreme in $\mathcal{F}_0$ hergeleitet werden.*

b) Vollständigkeit: *Aus $\models A$ folgt $\vdash_{\mathcal{F}_0} A$, alle Tautologien lassen sich in $\mathcal{F}_0$ herleiten.*

Korrektheit und Vollständigkeit von $\mathcal{F}_0$ (Fort.)

Als Hilfsmittel dient:

Lemma 2.13

Betrachte $A(p_1, \ldots, p_n) \in F_0$ *mit* $n > 0$. *Sei* φ *eine Bewertung. Mit*

$$P_i := \begin{cases} p_i, & \text{falls } \varphi(p_i) = 1 \\ \neg p_i, & \text{falls } \varphi(p_i) = 0 \end{cases} \qquad A' := \begin{cases} A, & \text{falls } \varphi(A) = 1 \\ \neg A, & \text{falls } \varphi(A) = 0 \end{cases}$$

gilt $P_1, \ldots, P_n \vdash A'$.

Folgerung

Folgerung 2.14

Sei $\Sigma \subseteq F_0, A \in F_0$.

- $\Sigma \vdash_{\mathcal{F}_0} A$ *gilt genau dann, wenn* $\Sigma \models A$ *gilt.*
- Σ *ist genau dann konsistent, wenn* Σ *erfüllbar ist.*
- *Ist* Σ *endlich und* $A \in F_0$*, dann ist* $\Sigma \vdash_{\mathcal{F}_0} A$ *entscheidbar.*

Beweis der Folgerung

Beweis:

$$
\begin{aligned}
& \Sigma \vdash_{\mathcal{F}_0} A \\
\overset{2.6}{\Longleftrightarrow}\ & \text{Es gibt } A_1, \ldots, A_n \in \Sigma \text{ mit } A_1, \ldots, A_n \vdash_{\mathcal{F}_0} A \\
\overset{D.T.}{\Longleftrightarrow}\ & \text{Es gibt } A_1, \ldots, A_n \in \Sigma \text{ mit} \\
& \quad \vdash_{\mathcal{F}_0} (A_1 \to (A_2 \to \ldots (A_n \to A) \ldots)) \\
\overset{2.12}{\Longleftrightarrow}\ & \text{Es gibt } A_1, \ldots, A_n \in \Sigma \text{ mit} \\
& \quad \models (A_1 \to (A_2 \to \ldots (A_n \to A) \ldots)) \\
\overset{D.T.}{\Longleftrightarrow}\ & \text{Es gibt } A_1, \ldots, A_n \in \Sigma \text{ mit } A_1, \ldots, A_n \models A \\
\overset{K.S.}{\Longleftrightarrow}\ & \Sigma \models A
\end{aligned}
$$

■

Beweis der Folgerung (Fort.)

Beweis:

$$
\begin{aligned}
& \Sigma \text{ ist konsistent} \\
\iff & \text{Es gibt kein } A \text{ mit } \Sigma \vdash A \text{ und } \Sigma \vdash \neg A \\
\iff & \text{Es gibt kein } A \text{ mit } \Sigma \models A \text{ und } \Sigma \models \neg A \\
\iff & \Sigma \text{ ist erfüllbar (Lemma 1.7(c)).}
\end{aligned}
$$

■

Sequenzenkalkül

Es gibt weitere korrekte und vollständige deduktive Systeme.
Das folgende System geht auf Gerhard Gentzen (1909 — 1945) zurück.
Es ist besonders zur Automatisierung von Beweisen geeignet.

Definition 2.15 (Gentzen-Sequenzenkalkül)

Seien $\Gamma, \Delta \subseteq F$ endliche Mengen von Formeln. Eine Sequenz ist eine Zeichenreihe der Form $\Gamma \vdash_G \Delta$.

Semantische Interpretation von Sequenzen: Für jede Bewertung φ gibt es eine Formel $A \in \Gamma$ mit $\varphi(A) = 0$ oder es gibt $B \in \Delta$ mit $\varphi(B) = 1$.

Sind $\Gamma = \{A_1, \ldots, A_n\}$ und $\Delta = \{B_1, \ldots, B_m\}$, dann entspricht die Sequenz $\Gamma \vdash_G \Delta$ der Formel

$$(A_1 \wedge \cdots \wedge A_n) \to (B_1 \vee \cdots \vee B_m).$$

Sequenzenkalkül (Fort.)

Definition 2.15 (Gentzen-Sequenzenkalkül (Fort.))

Der Kalkül für Objekte der Form $\Gamma \vdash_G \Delta$ wird definiert durch die Axiome:

$$\text{(Ax1) } \Gamma, A \vdash_G A, \Delta \qquad \text{(Ax2) } \Gamma, A, \neg A \vdash_G \Delta \qquad \text{(Ax3) } \Gamma \vdash_G A, \neg A, \Delta$$

Die Regeln des Sequenzenkalküls sind wie folgt:

$$\mathbf{R}_{\wedge,\vee}: \quad \frac{\Gamma, A, B \vdash_G \Delta}{\Gamma, A \wedge B \vdash_G \Delta} \qquad \frac{\Gamma \vdash_G A, B, \Delta}{\Gamma \vdash_G A \vee B, \Delta}$$

$$\mathbf{R}_{\rightarrow}: \quad \frac{\Gamma, A \vdash_G \Delta, B}{\Gamma \vdash_G A \rightarrow B, \Delta} \qquad \frac{\Gamma \vdash_G A, \Delta \,; \quad \Gamma, B \vdash_G \Delta}{\Gamma, A \rightarrow B \vdash_G \Delta}$$

$$\mathbf{R}_{\neg}: \quad \frac{\Gamma, A \vdash_G \Delta}{\Gamma \vdash_G \neg A, \Delta} \qquad \frac{\Gamma \vdash_G A, \Delta}{\Gamma, \neg A \vdash_G \Delta}$$

$$\mathbf{R}_{\wedge'}: \quad \frac{\Gamma \vdash_G A, \Delta \,; \quad \Gamma \vdash_G B, \Delta}{\Gamma \vdash_G A \wedge B, \Delta}$$

$$\mathbf{R}_{\vee'}: \quad \frac{\Gamma, A \vdash_G \Delta \,; \quad \Gamma, B \vdash_G \Delta}{\Gamma, A \vee B \vdash_G \Delta}$$

Sequenzenkalkül (Fort.)

Eine Sequenz $\Gamma \vdash_G \Delta$ heißt ableitbar, falls es eine endliche Folge von Sequenzen $\Gamma_1 \vdash_G \Delta_1, \ldots, \Gamma_r \vdash_G \Delta_r$ gibt mit $\Gamma_r \equiv \Gamma$, $\Delta_r \equiv \Delta$ und

> Jedes $\Gamma_j \vdash_G \Delta_j$ mit $1 \leq j \leq r$ ist ein Axiom oder geht aus vorangehenden Folgegliedern aufgrund einer Regel hervor.

Satz 2.16

Der Sequenzenkalkül ist

korrekt: $\Gamma \vdash_G \Delta$ *impliziert* $\Gamma \models \Delta$

vollständig: $\Gamma \models \Delta$ *impliziert* $\Gamma \vdash_G \Delta$.

Dabei ist $\Gamma \models \Delta$ *mit* $\Delta \subseteq F$ *endlich definiert als* $\Gamma \models \bigvee_{B \in \Delta} B$.

Beispiel

Beweise im Sequenzenkalkül werden bottom-up konstruiert und baumartig notiert:

Beispiel 2.17

Es gilt $p \vee q, \neg p \vee r \vdash_G q \vee r$.

Beweis:

$q, \neg p \vee r \vdash q, r$	Ax1
$p, r \vdash q, r$	Ax1
$p, \neg p \vdash q, r$	Ax2
$p, \neg p \vee r \vdash q, r$	$R_{\vee'}$
$p \vee q, \neg p \vee r \vdash q, r$	$R_{\vee'}$
$p \vee q, \neg p \vee r \vdash q \vee r$	$R_{\vee}$

■

Algorithmischer Aufbau der Aussagenlogik

Wir betrachten Verfahren, die bei gegebener endlicher Menge $\Sigma \subseteq F$ und $A \in F$ entscheiden, ob $\Sigma \models A$ gilt.

Die bisher betrachteten Verfahren prüfen alle Belegungen der in den Formeln vorkommenden Variablen oder zählen die Theoreme eines geeigneten deduktiven Systems auf. Dies ist sehr aufwendig.

Nutze Erfüllbarkeitschecker:

$$\Sigma \models A \quad \text{gdw.} \quad \Sigma \cup \{\neg A\} \text{ unerfüllbar.}$$

Die Komplexität von Erfüllbarkeit bleibt weiterhin groß: SAT ist NP-vollständig.

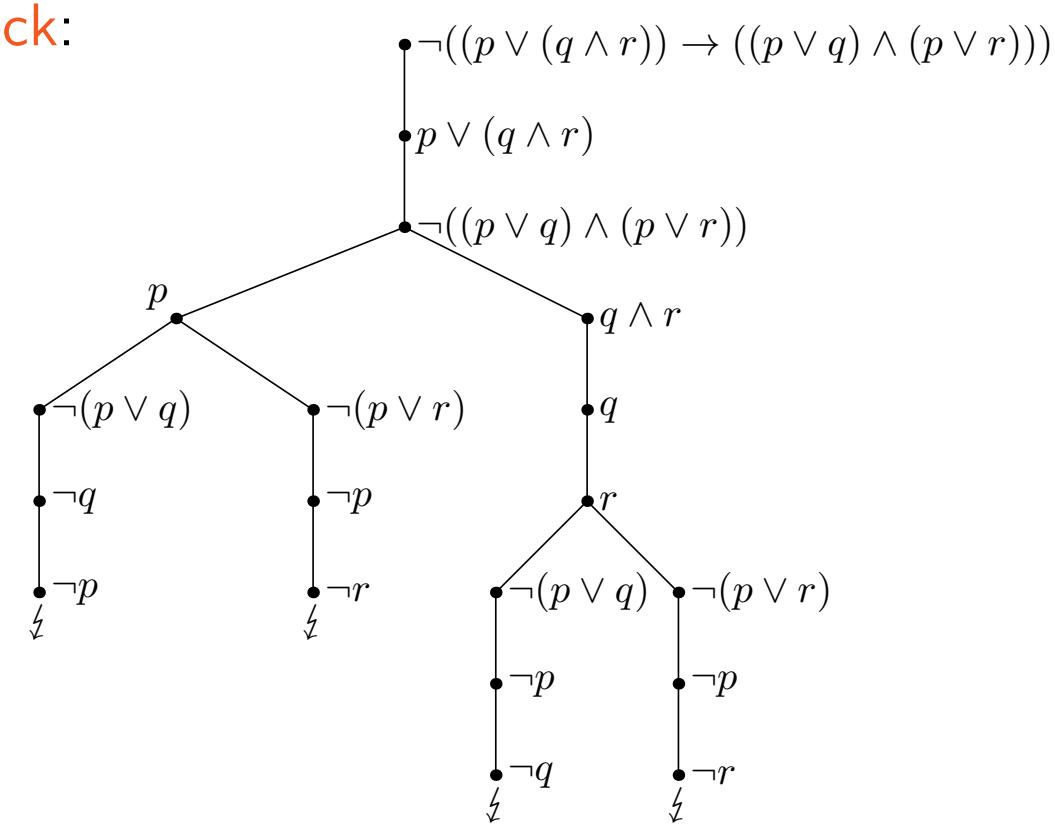
Suche nach Verfahren, die bei üblichen Eingaben schneller sind als die Brute-Force-Methode:

Semantische Tableaus Davis-Putnam Resolution.

Semantische Tableaus: Beispiel

Zeige, dass $\neg((p \vee (q \wedge r)) \rightarrow ((p \vee q) \wedge (p \vee r)))$ unerfüllbar ist.

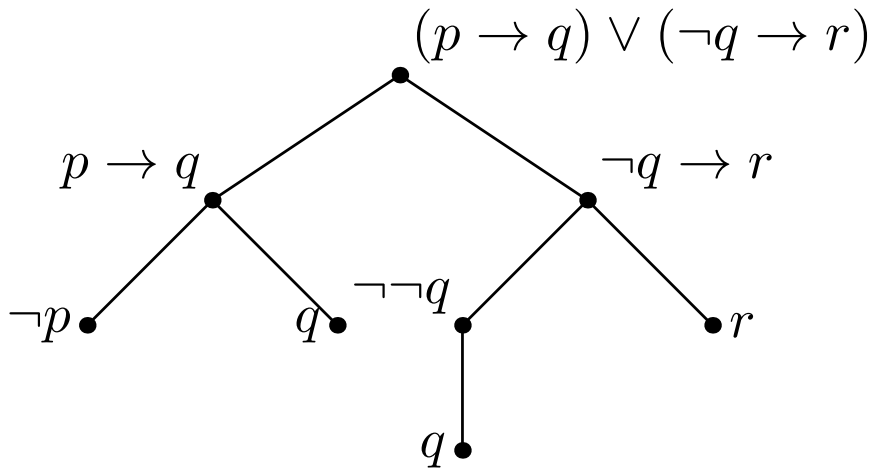
Erfüllbarkeitscheck:



Da alle Äste zu Widersprüchen führen, ist die Formel nicht erfüllbar.

Semantische Tableaus: Beispiel (Fort.)

Bestimme alle Bewertungen, die $A \equiv (p \to q) \vee (\neg q \to r)$ erfüllen:



Demnach ist $\{\varphi : F \to \mathbb{B} \mid \varphi(p) = 0 \text{ oder } \varphi(q) = 1 \text{ oder } \varphi(r) = 1\}$ die Menge aller Bewertungen, die A erfüllen.

An den Blättern lässt sich auch eine logisch äquivalente DNF ablesen, nämlich $\neg p \vee q \vee r$.

Intuition zu Tableaus

Die erfüllenden Bewertungen der Wurzelformel sind die Vereinigung der erfüllenden Bewertungen aller Äste.

- Für jede erfüllende Bewertung der Wurzel gibt es einen Ast in dem Tableau, so dass die Bewertung alle Formeln auf dem Ast erfüllt.
- Umgekehrt bestimmt jeder erfüllbare Ast erfüllende Bewertungen der Wurzelformel.

Trick: Sind Formeln maximal entfaltet (Tableau ist vollständig), sind erfüllende Bewertungen bzw. Widersprüche unmittelbar ersichtlich.

Definition von Tableaus

Zwei Arten von Formeln: β-Formeln führen zu Verzweigungen, α-Formeln führen nicht zu Verzweigungen:

- α-Formeln mit Komponenten α_1 und α_2 führen zu Folgeknoten mit Markierungen α_1 und α_2:

α	$\neg\neg A$	$A_1 \wedge A_2$	$\neg(A_1 \vee A_2)$	$\neg(A_1 \to A_2)$
α_1	A	A_1	$\neg A_1$	A_1
α_2	(A)	A_2	$\neg A_2$	$\neg A_2$

- β-Formeln mit Komponenten β_1 und β_2 führen zu Verzweigungen mit Knotenmarkierungen β_1 und β_2:

β	$\neg(A_1 \wedge A_2)$	$A_1 \vee A_2$	$A_1 \to A_2$
$\beta_1 \mid \beta_2$	$\neg A_1 \mid \neg A_2$	$A_1 \mid A_2$	$\neg A_1 \mid A_2$

Beachte: Jede Formel ist ein Literal (p oder $\neg p$ mit $p \in V$), eine α- oder eine β-Formel, und genau von einem dieser drei Typen.

Definition von Tableaus (Fort.)

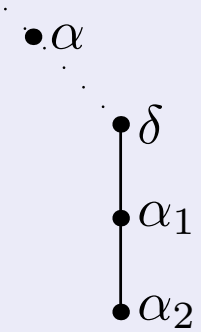
Definition 3.1 (Tableau)

Tableaus sind binäre Bäume, deren Knoten mit Formeln aus F markiert sind. Die Menge der Tableaus T_A für $A \in F$ ist induktiv definiert durch:

(a) Es gilt $\tau_A \in T_A$, wobei τ_A einen mit A beschrifteten Knoten hat:

(b) Ist $\tau \in T_A$ und δ Marke eines Blattes von τ, so lässt sich τ wie folgt zu einem Tableau $\tau' \in T_A$ fortsetzen:

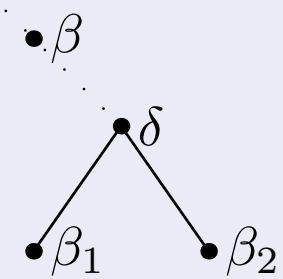
(α) Füge δ zwei aufeinander folgende Knoten hinzu, die mit α_1 und α_2 markiert sind, falls die α-Formel α auf dem Ast zu δ vorkommt:



Definition von Tableaus (Fort.)

Definition 3.1 (Tableau (Fort.))

(β) Füge in τ' als Nachfolger von δ zwei Knoten hinzu, die mit den Komponenten β_1 bzw. β_2 einer β-Formel β markiert sind, falls β auf dem Ast zu δ vorkommt:



Im Folgenden werden Äste in $\tau \in T_A$ mit ihrer Formelmenge $\Theta \subseteq F$ identifiziert.

Eigenschaften von Tableaus I: Semantik

Lemma 3.2

Sei $A \in F$ eine Formel und $\tau \in T_A$ ein Tableau für A. Dann gilt

$$A \text{ ist erfüllbar} \quad \text{gdw.} \quad \exists \text{ Ast } \Theta \in \tau : \Theta \text{ ist erfüllbar.}$$

Das Lemma folgt aus einer stärkeren Aussage. Für jede Bewertung φ gilt:

$$\varphi \text{ erfüllt } A \quad \text{gdw.} \quad \exists \text{ Ast } \Theta \in \tau : \varphi \text{ erfüllt } \Theta.$$

Die erfüllenden Bewertungen der Äste sind also genau die erfüllenden Bewertungen der Wurzelformel.

Tableaus sind nicht eindeutig, aber Lemma 3.2 hat folgende Konsequenz: Entweder hat jedes Tableau $\tau \in T_A$ einen erfüllbaren Ast oder keines.

Vollständige und abgeschlossene Mengen und Tableaus

Der Begriff der Erfüllbarkeit von Ästen ist semantischer Natur.
Das Ziel von Tableaus ist, Erfüllbarkeit von Formeln automatisch zu prüfen.
Dazu muss Erfüllbarkeit syntaktisch charakterisiert werden.

Definition 3.3

- Eine Formelmenge $\Theta \subseteq F$ heißt **vollständig**, falls mit $\alpha \in \Theta$ auch $\{\alpha_1, \alpha_2\} \subseteq \Theta$ und mit $\beta \in \Theta$ auch $\beta_1 \in \Theta$ oder $\beta_2 \in \Theta$.

  Ein Tableau τ heißt **vollständig**, falls jeder Ast $\Theta \in \tau$ vollständig ist.
- Eine Formelmenge Θ heißt **abgeschlossen**, falls es eine Formel $B \in F$ gibt mit $\{B, \neg B\} \subseteq \Theta$. Sonst heißt die Menge **offen**.

  Ein Tableau τ heißt **abgeschlossen**, wenn jeder Ast $\Theta \in \tau$ abgeschlossen ist.

Jedes Tableau kann zu einem vollständigen Tableau fortgesetzt werden.

Eigenschaften von Tableaus II: Syntax

Lemma 3.4 (Hintikka)

Sei $\Theta \subseteq F$ vollständig. Dann gilt: Θ ist erfüllbar gdw. Θ ist offen.

Abgeschlossene Mengen sind per Definition unerfüllbar.

Für die Rückrichtung sei Θ eine vollständige und offene Menge. Definiere

$$\varphi(p) := \begin{cases} 0 & \neg p \in \Theta \\ 1 & \text{sonst.} \end{cases}$$

Bewertung φ ist wohldefiniert.

Zeige mit Noetherscher Induktion nach der Länge der Formeln, dass $\varphi(A) = 1$ für alle $A \in \Theta$.

Korrektheit und Vollständigkeit von Tableaus

Satz 3.5

Eine Formel $A \in F$ ist unerfüllbar gdw. es ein abgeschlossenes Tableau $\tau \in T_A$ gibt.

Auch hier gilt: es gibt ein abgeschlossenes Tableau für A gdw. alle vollständigen Tableaus für A abgeschlossen sind.

Die Tableaumethode geht auf Evert Willem Beth (1908 — 1964) zurück.

Vollständige und offene Formelmengen sind Hintikka-Mengen, nach Jaakko Hintikka (*1929). Das Lemma von Hintikka zeigt, dass sie erfüllbar sind.

Korrektheit und Vollständigkeit von Tableaus (Fort.)

Beweis (von Satz 3.5)

Sei A nicht erfüllbar.
Jedes Tableau kann zu einem vollständigen Tableau fortgesetzt werden.
Also gibt es zu A ein vollständiges Tableau $\tau \in T_A$.
Mit Lemma 3.2 sind alle Äste $\Theta \in \tau$ nicht erfüllbar.
Mit Lemma 3.4 sind alle Äste $\Theta \in \tau$ abgeschlossen.
Also gibt es ein abgeschlossenes Tableau $\tau \in T_A$.

Für die Rückrichtung sei $\tau \in T_A$ abgeschlossen.
Abgeschlossene Äste sind unerfüllbar.
Mit Lemma 3.2 ist Formel A unerfüllbar. ■

Tableaus für Formelmengen

Sei $\Sigma \subseteq F$ eine ggf. unendliche Formelmenge.
Die Menge T_Σ der **Tableaus für Σ** ist definiert wie zuvor, nur dass

- die Konstruktion mit einer Formel $A \in \Sigma$ beginnt und
- in jedem Schritt $\sigma \in \Sigma$ an ein Blatt δ angehängt werden darf.

Tableau $\tau \in T_\Sigma$ heißt vollständig, wenn zusätzlich zu den vorherigen Bedingungen jeder Ast $\Theta \in \tau$ die Menge Σ enthält, also $\Sigma \subseteq \Theta$.

Tableaus für Formelmengen (Fort.)

Lemma 3.6

Seien $\Sigma \subseteq F$ und $\tau \in T_\Sigma$ mit $\Sigma \subseteq \Theta$ für jeden Ast $\Theta \in \tau$. Dann gilt:

$$\Sigma \text{ ist erfüllbar} \quad \text{gdw.} \quad \exists \text{ Ast } \Theta \in \tau : \Theta \text{ ist erfüllbar.}$$

Satz 3.7

Eine Formelmenge $\Sigma \subseteq F$ ist unerfüllbar gdw. T_Σ ein abgeschlossenes Tableau enthält.

Der alte Beweis funktioniert noch immer, modulo folgender Änderungen:

Lemma 3.2 ist durch Lemma 3.6 zu ersetzen.

Für die Vollständigkeit ist folgendes Lemma notwendig.

Lemma 3.8

Für jede Formelmenge $\Sigma \subseteq F$ existiert ein vollständiges Tableau $\tau \in T_\Sigma$.

Systematische Tableaukonstruktion

Beweis von Lemma 3.8 Sei Σ unendlich. Es wird eine nicht-terminierende Methode angegeben, die eine Folge von Tableaus

$$\tau_0 \subseteq \tau_1 \subseteq \ldots \quad \text{konstruiert mit} \quad \tau := \bigcup_{i \in \mathbb{N}} \tau_i \ \textbf{vollständig}.$$

Da $\Sigma \subseteq F$, ist Σ abzählbar, also $\Sigma = \{A_0, A_1, \ldots\}$.

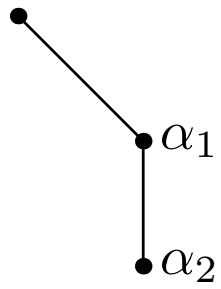
Nutze eine FIFO-Worklist $WL := \emptyset$ zur Speicherung von Knoten.
Nutze ferner einen Zähler $j := 0$, um Σ zu durchlaufen.

- $\tau_0 := \tau_{A_0}$. Ist A_0 kein Literal, push den Knoten von A_0 auf WL.
- τ_{n+1} entsteht aus τ_n wie folgt.

  Falls $WL \neq \emptyset$, pop WL. Sei der Knoten mit $Y \in F$ beschriftet.

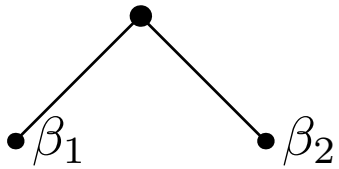
Systematische Tableaukonstruktion (Fort.)

- Ist Y eine α-Formel, erweitere jeden Ast, der durch den Knoten von Y geht, um die Teilformeln α_1 und α_2:



  Falls α_1 bzw. α_2 keine Literale sind, füge alle neuen mit α_1 bzw. α_2 beschrifteten Knoten der Worklist hinzu.

- Ist Y eine β-Formel, erweitere jeden Ast, der durch Y geht, um



  Falls die Teilformeln β_1, β_2 keine Literale sind, füge die entsprechenden Knoten der Worklist hinzu.

Systematische Tableaukonstruktion (Fort.)

Falls $WL = \emptyset$, inkrementiere j und wähle $Y := A_j$.

- Hänge mit Y beschriftete Knoten an alle Äste an.
  Sofern Y kein Literal ist, füge die Knoten der Worklist hinzu.

Systematische Tableaukonstruktion (Fort.)

Behauptung: τ ist vollständig.
Genauer: Jeder Ast $\Theta \in \tau$ ist vollständig und enthält Σ.

Beweis (Skizze):
Sei $\alpha \in \Theta$ eine α-Formel.
Dann ist sie bei der Erstellung in die Worklist aufgenommen worden.
Wegen der FIFO-Reihenfolge, wurde sie irgendwann bearbeitet.
Also sind $\{\alpha_1, \alpha_2\} \subseteq \Theta$.

Betrachte $A_i \in \Sigma$.
Irgendwann ist $j = i$ geworden.
Angenommen, das wäre nicht der Fall.
Dann gab es einen Index, bei dem die Worklist nie geleert wurde.
Das muss falsch sein.
Mit der Entnahme einer Formel $A \in F$ sind zwar endlich viele Formeln der Worklist hinzugefügt worden, die waren aber alle kleiner.

Übung: Warum folgt Terminierung?

Entscheidbarkeit und Semi-Entscheidbarkeit

Um aus der systematischen Tableaukonstruktion Semi-Entscheidbarkeit für Unerfüllbarkeit abzuleiten, passe das Verfahren wie folgt an:

Füge keine Knoten an abgeschlossene Äste an.

Lemma 3.9

(1) Die systematische Tableaukonstruktion terminiert für $\Sigma \subseteq F$ endlich.

(2) Sei $\Sigma \subseteq F$ unendlich und nicht erfüllbar. Dann terminiert die modifizierte Tableaukonstruktion mit einem abgeschlossenen Tableau.

Beachte: Der Kompaktheitssatz folgt aus der zweiten Aussage.
Ist Σ nicht erfüllbar, enthält T_Σ ein endliches, abgeschlossenes Tableau.
Also ist eine endliche Teilmenge von Σ nicht erfüllbar.

Entscheidbarkeit und Semi-Entscheidbarkeit (Fort.)

Lemma 3.10 (König)

Sei T ein unendlicher Baum mit endlichem Ausgangsgrad.
Dann gibt es einen unendlichen Pfad in T.

Zeige Lemma 3.9(2):

Im Fall der Terminierung ist das resultierende Tableau abgeschlossen.
Abgeschlossenheit ist nämlich die einzige Bedingung zur Terminierung.

Es bleibt Terminierung zu zeigen.
Angenommen das modifizierte Verfahren terminiert nicht.
Dann wird ein unendliches Tableau τ konstruiert.
Da das Tableau endlichen Ausgangsgrad hat, gibt es mit Königs Lemma einen unendlichen Pfad $\Theta \in \tau$.
Wie in Lemma 3.8 enthält der Pfad Σ, ist vollständig und offen.
Mit Hintikkas Lemma ist Θ erfüllbar.
Damit ist auch Σ erfüllbar. Widerspruch.

Entscheidbarkeit und Semi-Entscheidbarkeit (Fort.)

Bemerkung 3.11

- *Das Tableauverfahren ist ein Semi-Entscheidungsverfahren für Unerfüllbarkeit aufzählbarer Formelmengen $\Sigma \subseteq F$.*
- *Das Tableauverfahren ist eine Entscheidungsprozedur für Erfüllbarkeit endlicher Formelmengen $\Sigma \subseteq F$.*

Für die Entscheidbarkeit ist zu beachten, dass bei einer aufzählbaren Menge das Hinzufügen einer Formel $A_i \in \Sigma$ zu einem Tableau effektiv ist. Ebenso ist der Test auf Abgeschlossenheit entscheidbar.

Normalformen

Vorteile:
Die einfachere Gestalt der Normalform lässt spezielle Algorithmen zur Lösung bestimmter Fragestellungen zu.

Die Transformation sollte nicht zu teuer sein, sonst würde sich der Aufwand nicht lohnen.

Beispiele:

- Aus einer DNF lassen sich alle erfüllenden Belegungen direkt ablesen.
- Aus einer minimalen DNF lassen sich leicht Schaltnetze (mit UND-, ODER-, NEG-Gattern) herleiten.
- Die systematische Tableau-Konstruktion erlaubt es, diese Normalformen aus einem vollständigen Tableau abzulesen.

Normalformen (Fort.)

Transformiert werden kann in eine

- logisch äquivalente Formel: $A \models\!\!\models T(A)$
- erfüllbarkeitsäquivalente Formel: A erfüllbar gdw. $T(A)$ erfüllbar

Wir behandeln drei dieser Normalformen:

- Negationsnormalform (**NNF**) Form in $\neg, \vee, \wedge$
- Konjunktive Normalform (**KNF**) Form in $\neg, \vee, \wedge$
- Disjunktive Normalform (**DNF**) Form in $\neg, \vee, \wedge$

Negationsnormalform

Formel $A \in F$ ist in **Negationsnormalform (NNF)**, falls jede Negation direkt vor einer Variablen steht und keine zwei Negation einander folgen.

Definition 3.12 (NNF)

Die Menge der Formeln in **NNF** ist induktiv definiert durch

- Für $p \in V$ sind p und $\neg p$ in NNF.
- Sind A, B in NNF, dann sind auch $(A \vee B)$ und $(A \wedge B)$ in NNF.

Lemma 3.13

Zu jeder Formel $A \in F(\{\neg, \wedge, \vee, \rightarrow, \leftrightarrow\})$ gibt es $B \in F(\neg, \vee, \wedge)$ in NNF mit $A \models\!\!\models B$ und $|B| \in O(|A|)$.

Konjunktive Normalform

Definition 3.14 (Klausel)

Eine Formel $A \equiv (L_1 \vee ... \vee L_n)$ mit Literalen $L_1, \ldots, L_n$ heißt **Klausel**.

- Sind alle Literale negativ, so ist es eine **negative** Klausel.

  Sind alle Literale positiv, so ist es eine **positive** Klausel.

  Klauseln, die maximal ein positives Literal enthalten, heißen **Horn-Klauseln**.

- A wird k-Klausel genannt, falls A maximal $k \in \mathbb{N}$ Literale enthält.

  1-Klauseln werden auch Unit-Klauseln genannt.

Eine Formel $A \equiv (A_1 \wedge ... \wedge A_m)$ ist in **KNF**, falls A eine Konjunktion von Klauseln $A_1, \ldots, A_m$ ist.

- Handelt es sich um k-Klauseln, so ist A in k-**KNF**.

Konjunktive Normalform (Fort.)

Beispiel 3.15

$A \equiv (p \vee q) \wedge (p \vee \neg q) \wedge (\neg p \vee q) \wedge (\neg p \vee \neg q)$ ist in 2-KNF.

Betrachtet man Klauseln als Mengen von Literalen, so lassen sich Formeln in KNF als Mengen von Literalmengen darstellen.
Am Beispiel A:

$$\{\{p, q\}, \{p, \neg q\}, \{\neg p, q\}, \{\neg p, \neg q\}\}.$$

Lemma 3.16

Zu jeder Formel $A \in F$ gibt es eine Formel B in KNF mit $A \models\!\!\!\models B$ und $|B| \in O(2^{|A|})$.

Die Schranke ist **strikt**:
Es gibt eine Folge von Formeln $(A_n)_{n\in\mathbb{N}}$ mit $|A_n| = 2n$, für die jede logisch äquivalente Formel B_n in KNF mindestens die Länge 2^n besitzt.

Disjunktive Normalform (Fort.)

Definition 3.17 (DNF)

Eine Formel $A \in F$ ist in **DNF**, falls A eine Disjunktion von Konjunktionen von Literalen ist:

$$A \equiv (A_1 \vee \ldots \vee A_m) \quad \text{mit} \quad A_i \equiv (L_1^i \wedge \ldots \wedge L_{n_i}^i).$$

Definition 3.18 (Duale Formel)

Die **duale Formel** $d(A)$ einer Formel $A \in F$ ist definiert durch:

$$\begin{aligned} d(p) &\equiv p \quad \text{für } p \in V \\ d(\neg A) &\equiv \neg d(A) \\ d(B \vee C) &\equiv d(B) \wedge d(C) \\ d(B \wedge C) &\equiv d(B) \vee d(C). \end{aligned}$$

Zusammenhänge zwischen den Normalformen

Lemma 3.19

Für jede Formel $A \in F$ gilt:

(1) *Ist A in KNF, dann ist NNF($\neg A$) in DNF.*

(2) *Ist A in KNF, so ist $d(A)$ in DNF und umgekehrt.*

Lemma 3.20

Für jede Formel $A \in F$ gilt:

(1) *Setzt man $\psi(p) := 1 - \varphi(p)$, so gilt $\psi(d(A)) = 1 - \varphi(A)$.*

(2) *A ist Tautologie gdw. $d(A)$ ein Widerspruch ist.*

(3) *A ist erfüllbar gdw. $d(A)$ keine Tautologie ist.*

Davis-Putnam-Algorithmen

Idee: Reduziere Erfüllbarkeit für eine Formel mit $n \in \mathbb{N}$ Variablen auf das Erfüllbarkeitsproblem für Formeln mit **maximal** $n - 1$ Variablen.

Ansatz: Suche einer erfüllenden Bewertung durch iterative Auswahl der Werte einzelner Variablen — **Bottom-Up-Verfahren**.

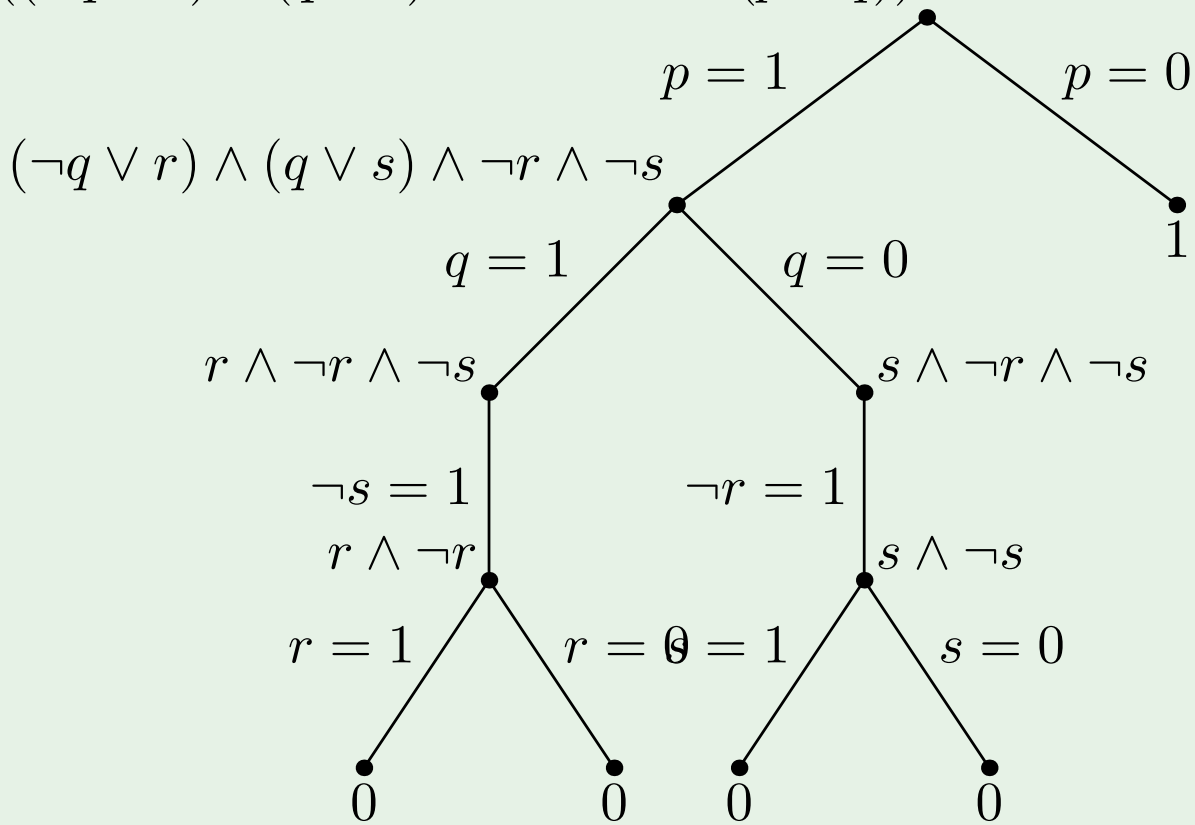
Algorithmen, die mit dieser Idee, Heuristiken und weiteren Verfeinerungen arbeiten, werden als Davis-Putnam-Algorithmen bezeichnet, nach Martin Davis (*1928) und Hilary Putnam (*1926).

Voraussetzung: Formel in **NNF** über $\neg, \wedge, \vee$.

Davis-Putnam-Algorithmen (Fort.)

Beispiel 3.21 (Darstellung der Abarbeitung als Baum)

$A \equiv \neg p \vee ((\neg q \vee r) \wedge (q \vee s) \wedge \neg r \wedge \neg s \wedge (p \vee q))$



Substitution

Definition 3.22 (Substitution)

Sei Formel $A \in F$ in NNF und $p \in V$.
Definiere $A[p/1]$ als Ergebnis des folgenden Ersetzungsprozesses:

(1) Ersetze in A jedes Vorkommen von p durch 1.

(2) Führe folgende Regeln so lange wie möglich durch:

- Tritt eine Teilformel $\neg 1$ auf, ersetze sie durch 0.
- Ersetze $\neg 0$ durch 1.
- Teilformeln $B \wedge 1$ sowie $B \vee 0$ werden durch B ersetzt.
- Teilformeln $B \vee 1$ werden durch 1 ersetzt.
- Teilformeln $B \wedge 0$ werden durch 0 ersetzt.

Analog ist $A[p/0]$ definiert, wobei p durch 0 ersetzt wird.

Allgemeiner verwende $A[l/1]$ bzw. $A[l/0]$ für Literale l.

Substitution (Fort.)

Lemma 3.23

$A[p/1]$ und $A[p/0]$ sind wohldefiniert.

Formel $A[p/i]$ mit $i \in \mathbb{B}$ ist:

- *eine Formel in NNF bzw. KNF, wenn A diese Form hatte,*
- *die leere Formel, die als wahr interpretiert wird, notiert $A[p/i] = 1$,*
- *die leere Klausel, die als falsch interpretiert wird, $A[p/i] = 0$.*

Variable $p \in V$ kommt nicht mehr in $A[p/i]$ vor.

Beispiel 3.24

Für A in KNF und Literal l gilt:
$A[l/1]$ entsteht durch Streichen aller Klauseln in A, die Literal l enthalten, und durch Streichen aller Vorkommen von $\neg l$ in den anderen Klauseln.

Korrektheit von Davis-Putnam-Algorithmen

Lemma 3.25

Eine Formel A in NNF ist erfüllbar gdw. $A[p/1] = 1$ oder $A[p/0] = 1$ oder eine der Formeln $A[p/1], A[p/0]$ erfüllbar ist.

Das Lemma folgt aus der Tatsache, dass für jede Bewertung φ gilt

$$\varphi(A) = \varphi(A[p/i]), \quad \text{wobei } i = \varphi(p).$$

Durch Testen der Formeln $A[p/1]$ und $A[p/0]$, die nun $p \in V$ nicht mehr enthalten, kann rekursiv die Erfüllbarkeit von A entschieden werden.

Regelbasierte Definition von Davis-Putnam

Definition 3.26 (Regeln für Formeln in NNF)

Pure-Literal Regel Kommt eine Variable $p \in V$ in Formel A nur positiv oder nur negativ vor, belege p mit 1 bzw. p mit 0 und kürze die Formel.

A ist erfüllbarkeitsäquivalent mit $A[p/1]$ bzw. $A[p/0]$.

Splitting-Regel Kommt eine Variable $p \in V$ sowohl positiv als auch negativ in A vor, bilde die zwei Kürzungen $A[p/1]$ und $A[p/0]$.

A ist erfüllbar gdw. bei einer der Kürzungen der Wert 1 oder eine erfüllbare Formel auftritt.

Regelbasierte Definition von Davis-Putnam (Fort.)

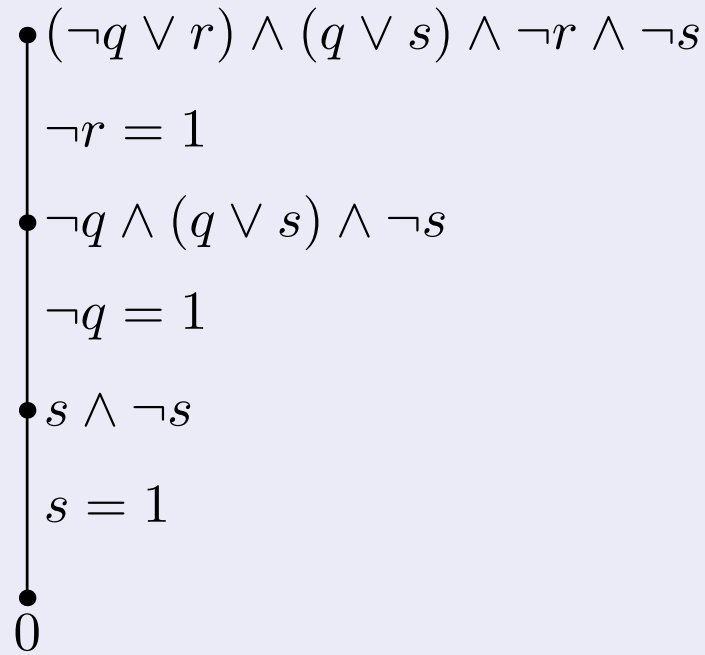
Definition 3.27 (Regeln für Formeln in **KNF**)

Unit-Regel

Sei A in KNF und enthalte eine Unit- Klausel $A_i \equiv l$.

Bilde $A[l/1]$:

A erfüllbar gdw. $A[l/1]$ erfüllbar.



Regelbasierte Definition von Davis-Putnam (Fort.)

Klausel A_1 **subsumiert** Klausel A_2, in Zeichen $A_1 \subseteq A_2$, falls jedes Literal aus A_1 auch in A_2 auftritt.

Aus der Erfüllbarkeit einer Klausel A_1 folgt sofort die Erfüllbarkeit aller Klauseln A_2, die sie subsummiert: $A_1 \subseteq A_2$.

Definition 3.27 (Regeln für Formeln in **KNF** (Fort.))

Subsumption-Rule Sei A in KNF. Streiche aus A alle Klauseln, die von anderen subsumiert werden: Funktion Subsumption_Reduce(A).

Streiche dabei auch tautologische Klauseln, die p und $\neg p$ für ein $p \in V$ enthalten.

> *Da Klauseln konjunktiv verknüpft sind, sind nur die zu berücksichtigen, die von keiner anderen subsumiert werden.*

procedure DPA — Davis-Putnam-Algorithmus

Eingabe: A in KNF
Ausgabe: Boolscher Wert für Erfüllbarkeit $\{0,1\}$

begin
if $A \in \{0,1\}$ **then** **return**(A);
p:=**pure**(A,s);
  //*liefert Variable und Belegung, falls nur positiv*
  //*oder nur negativ vorkommt, sonst* **null**
if $p \neq$ **null** **then** **return**(DPA(A[p/s]));
p:=**unit**(A,s);
  //*Unit Klausel mit Belegung, sonst* **null**
if $p \neq$ **null** **then** **return**(DPA(A[p/s]));
A:=Subsumption_Reduce(A); //*entfernt subs. Klauseln*
p:=**split**(A); //*liefert Variable in A*
if DPA($A[p/1]) = 1$ **then** **return**(1);
return(DPA(A[p/0]));
end

Auswahlkriterien für die Splitting-Regel

- Wähle die erste in der Formel vorkommende Variable.
- Wähle die Variable, die am häufigsten vorkommt.
- Wähle die Variable mit $\sum_{p \text{ in } A_i} |A_i|$ minimal.
- Wähle die Variable, die in den kürzesten Klauseln am häufigsten vorkommt.
- Berechne die Anzahl der positiven und negativen Vorkommen in den kürzesten Klauseln und wähle die Variable mit der größten Differenz.
- Weitere Heuristiken in Implementierungen vorhanden.

Resolution

Idee: Aus Klauseln $(A \vee l)$ und $(B \vee \neg l)$ wird die neue Klausel $(A \vee B)$ erzeugt, denn

$$(A \vee l) \wedge (B \vee \neg l) \models\!\models (A \vee l) \wedge (B \vee \neg l) \wedge (A \vee B).$$

Dabei sei $\neg l \equiv \neg p$, falls $l \equiv p$ mit $p \in V$. Falls $l \equiv \neg p$, dann $\neg l \equiv p$.

Ziel: Leere Klausel $\sqcup$ erzeugen, um **Unerfüllbarkeit** zu zeigen.

Resolution arbeitet auf Formeln **in KNF**. Dabei ist es günstig, Klauseln als Mengen darzustellen:

$$(p \vee \neg q \vee p) \quad \text{dargestellt als} \quad \{p, \neg q\}.$$

Resolution geht zurück auf John Alan Robinson (*1928).

Resolution (Fort.)

Definition 3.28 (Resolvente)

Seien K_1, K_2 Klauseln und l ein Literal mit $l \in K_1$ und $\neg l \in K_2$. Dann ist

$$R \equiv (K_1 \setminus \{l\}) \cup (K_2 \setminus \{\neg l\})$$

die **Resolvente von K_1 und K_2 nach l**.

Beachte: Die Resolvente kann die leere Klausel $\sqcup$ sein.

Das Hinzufügen von Resolventen führt zu äquivalenten Formeln.

Lemma 3.29

Sei A in KNF und R Resolvente zweier Klauseln aus A. Dann gilt

$$A \models\!\!\models A \cup \{R\}.$$

Resolution (Fort.)

Definition 3.30 (Herleitungen)

Seien A in KNF und K Klausel. Eine Folge $K_1, \ldots, K_n$ von Klauseln mit $K_n \equiv K$ ist eine **Herleitung von K aus A**, $A \underset{Res}{\vdash} K$, falls für $1 \leq k \leq n$:

$$K_k \in A \text{ oder } K_k \text{ ist eine Resolvente zweier } K_i,\ K_j \text{ mit } i, j < k.$$

Lemma 3.31

Als Kalkül ist Resolution korrekt aber **nicht vollständig**:

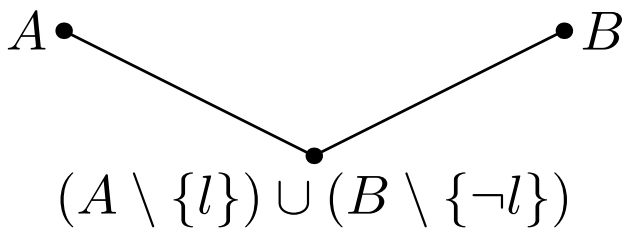
$$A \underset{Res}{\vdash} K \text{ impliziert } A \models K. \text{ Die Umkehrung gilt nicht.}$$

Satz 3.32 (Korrektheit und Widerlegungsvollständigkeit, Robinson)

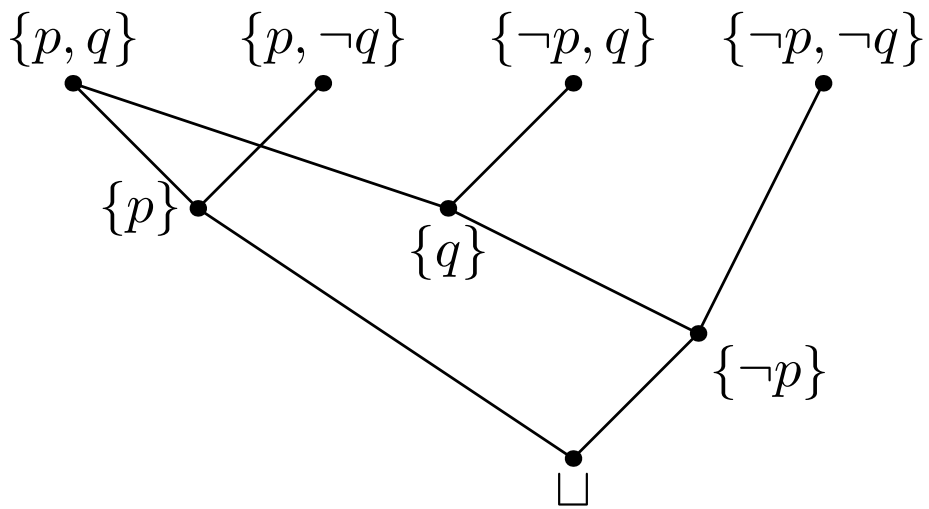
Eine Formel A in KNF ist unerfüllbar gdw. $A \underset{Res}{\vdash} \square$.

Darstellung

Darstellung einer Resolvente zweier Klauseln A, B nach l:



Darstellung von Herleitungen als gerichteter, azyklischer Graph (DAG):



Resolventenmethode: Heuristiken

Starke Herleitungen: Sei A in KNF und unerfüllbar.
Dann gibt es eine Herleitung $K_1, \ldots K_n \equiv \sqcup$, so dass

1. in der Herleitung keine Klausel mehrfach auftritt,
2. in der Herleitung keine Tautologie auftritt,
3. in der Herleitung keine schon subsumierte Klausel auftritt:
   Es gibt keine K_i, K_j mit $i < j$ und $K_i \subseteq K_j$.

Resolventenmethode: Heuristiken (Fort.)

- Stufenstrategie (Resolutionsabschluss)
  (Alle erfüllenden Bewertungen)
- Stützmengenrestriktion
  (Set-of-Support, Unit-Klauseln bevorzugen)
- P-(N-)Resolution
- Lineare Resolution (SL-Resolution, PROLOG-Inferenzmaschine).

Resolventenmethode: Heuristiken (Fort.)

Beispiel: $A \equiv \{\{\neg p, \neg q, \neg r\}, \{p, \neg s\}, \{q, \neg r\}, \{r, \neg t\}, \{t\}\}$
Stufen:

0	1	2	3
1. $\{\neg p, \neg q, \neg r\}$	6. $\{\neg q, \neg r, \neg s\}$ (1,2)	11. $\{\neg r, \neg s\}$ (6,3)	21. $\{\neg s, \neg t\}$ (11,4)
2. $\{p, \neg s\}$	7. $\{\neg p, \neg r\}$ (1,3)	12. $\{\neg q, \neg s, \neg t\}$ (6,4)	22. $\{\neg s\}$ (11,10)
3. $\{q, \neg r\}$	8. $\{\neg p, \neg q, \neg t\}$ (1,4)	13. $\{\neg p, \neg t\}$ (7,4)	⋮
4. $\{r, \neg t\}$	9. $\{q, \neg t\}$ (3,4)	14. $\{\neg p, \neg r, \neg t\}$ (8,3)	
5. $\{t\}$	10. $\{r\}$ (4,5)	15. $\{\neg p, \neg q\}$ (8,5)	
		16. $\{q\}$ (10,3)	
		17. $\{\neg r, \neg s, \neg t\}$ (6,9)	
		18. $\{\neg q, \neg s\}$ (6,10)	
		19. $\{\neg p\}$ (7,10)	
		20. $\{\neg p, \neg t\}$ (8,9)	

Erhalte die erfüllende Bewertung

$$\varphi(q) = 1, \varphi(p) = 0, \varphi(s) = 0, \varphi(r) = 1, \varphi(t) = 1.$$

Teil II

Prädikatenlogik erster Stufe (mit Gleichheit)

Grundlagen der Prädikatenlogik

Ziel: Formulierung und Folgerung von Beziehungen zwischen Elementen eines Datenbereichs.

Anwendungen:

- Lösung von Anfragen auf Datenmengen in der KI oder in Informationssystemen.
- Formulierung von Integritätsbedingungen auf Daten: Schleifeninvarianten eines Programms, Constraints auf XML-Dateien oder Datenbankeinträgen.
- Lösung von Constraint-Systemen beim Testen oder Planen.
- Logisches Programmieren.

Syntax der Prädikatenlogik 1879 im Artikel Begriffsschrift von Gottlob Frege (1848 — 1925).

Semantik erst 1934 durch Alfred Tarski (1901 — 1983).

Grundlagen der Prädikatenlogik (Fort.)

Semantisch: Elemente eines Datenbereich, Funktionen auf diesen Elementen und Beziehungen zwischen diesen Elementen.

Syntaktisch:

- Terme beschreiben Elemente des Datenbereichs.
  Zur Bezeichnung der Elemente: Konstanten und Variablen.
  Zur Berechnung weiterer Elemente: Funktionssymbole.
- Formeln treffen Aussagen über die Elemente: wahr oder falsch.
  Zum Fassen von Beziehungen zwischen Elementen: Prädikatssymbole.
  Verknüpfung der resultierenden Wahrheitswerte über logische Junktoren und Quantoren.

Funktions- und Prädikatssymbole **hängen von der Anwendung ab**, sind also Parameter in der Definition der Syntax.

Logische Symbole sind **fest**.

Grundlagen der Prädikatenlogik (Fort.)

Beispiel 4.1 (Beschreibung mathematischer Beziehungen)

Syntax: Konstanten 1, 2, 3, Funktionssymbole $+$, $/$, Variablen x, y, z, Pädikat $<$, Junktoren $\rightarrow$, $\wedge$, Quantoren $\forall$, $\exists$

$$\text{Terme:} \quad 1,\ 1+\frac{2}{3},\ x+\frac{3}{2}$$

$$\text{Formeln:} \quad x<3,\ \forall x \forall y(x<y \rightarrow \exists z(x<z \wedge z<y))$$

Semantik: Datenbereich $\mathbb{Q}$, Konstante 1 bis Prädikat $<$ mit der üblichen Bedeutung.

Grundlagen der Prädikatenlogik (Fort.)

Beispiel 4.2 (Beschreibung von Beziehungen zwischen Daten)

Syntax: Variablen x, y, Funktion $\mathit{weiteDerReise}(-)$, Pädikate $\mathit{istHund}(-)$, $\mathit{istFisch}(-)$, $<$, Quantor $\forall$.

Formel:

$$\forall x \forall y \big((\mathit{istHund}(x) \wedge \mathit{istFisch}(y)) \rightarrow \mathit{weiteDerReise}(x) < \mathit{weiteDerReise}(y) \big)$$

Semantik: Datenbereich $\{\mathit{Lassie}, \mathit{Nemo}\} \cup \mathbb{N} \cup \{\bot\}$.

Funktion $\mathit{weiteDerReise}(x)$ liefert die Weite der Reise eines Tieres des Datenbereichs und $\bot$, falls kein Tier eingegeben wird.

Prädikat $\mathit{istFisch}(x)$ liefert wahr, falls x ein Fisch ist.

Syntax der Prädikatenlogik erster Stufe

Definition 4.3 (Signatur)

Eine Signatur ist ein Paar $S = (\mathit{Funk}, \mathit{Präd})$ mit

- Funk einer Menge von Funktionssymbolen $f, g, \ldots \in \mathit{Funk}$ und
- $\mathit{Präd}$ einer Menge von Prädikatssymbolen $p, q, \ldots \in \mathit{Präd}$.

Jedes Funktions- und jedes Prädikatssymbol hat eine Stelligkeit $k \in \mathbb{N}$.
Schreibe auch $f/_k \in \mathit{Funk}$ bzw. $p/_k \in \mathit{Präd}$, falls f bzw. p Stelligkeit k hat.
0-stellige Funktionen und Prädikate heißen Konstanten.

Voraussetzungen:

- Funk und $\mathit{Präd}$ seien entscheidbar, nicht notwendigerweise endlich.
- Neben der Signatur gebe es eine abzählbare Menge V an Variablen.
  Es seien V, Funk, $\mathit{Präd}$ paarweise disjunkt und enthalten nicht

$$\neg, \wedge, \vee, \rightarrow, \leftrightarrow, \exists, \forall, \ , \ , = , (,).$$

Syntax der Prädikatenlogik erster Stufe (Fort.)

Definition 4.4 (Syntax der Prädikatenlogik erster Stufe)

Sei $S = (\mathit{Funk}, \mathit{Präd})$ eine Signatur.
Die Menge $\mathit{Term}(S)$ aller Terme über S ist induktiv definiert als

$$t ::= x \mid f(t_1, \ldots, t_k), \quad \text{wobei } x \in V \text{ und } f/_k \in \mathit{Funk}.$$

Die Menge $FO(S)$ der prädikatenlogischen Formeln erster Stufe über S ist induktiv definiert als

$$\begin{aligned} A ::= {} & t_1 = t_2 \mid p(t_1, \ldots, t_k) \mid \\ & (\neg A) \mid (A_1 \wedge A_2) \mid (A_1 \vee A_2) \mid (A_1 \to A_2) \mid (A_1 \leftrightarrow A_2) \mid \\ & (\exists x A) \mid (\forall x A) \end{aligned}$$

mit $t_1, t_2, \ldots, t_k \in \mathit{Term}(S)$, $p/_k \in \mathit{Präd}$ und $x \in V$.

Nenne $t_1 = t_2$ und $p(t_1, \ldots, t_k)$ auch atomare Formeln.

Syntax der Prädikatenlogik erster Stufe (Fort.)

Zur besseren **Lesbarkeit**:

- Äußere Klammern weglassen.
- **Prioritäten**: $\neg, \wedge, \vee, \rightarrow, \leftrightarrow$
- $\forall x_1, \ldots, x_n A$ steht für $\forall x_1 (\ldots (\forall x_n A) \ldots)$
  $\exists x_1, \ldots, x_n A$ steht für $\exists x_1 (\ldots (\exists x_n A) \ldots)$
- Für zweistellige Prädikats- und Funktionssymbole wird auch Infix-Notation genutzt. Beispiel $t_1 < t_2$ statt $< (t_1, t_2)$.

Syntax der Prädikatenlogik erster Stufe (Fort.)

Definition 4.5 (Freie und gebundene Variablen)

In einer Formel (QxA) mit $Q \in \{\exists, \forall\}$ ist A der Geltungsbereich von Qx.

Ein Vorkommen einer Variablen $x \in V$ in einer Formel heißt gebunden, falls es im Geltungsbereich eines Quantors Qx vorkommt.

Sonstige Vorkommen einer Variablen heißen frei.

Formeln ohne freie Vorkommen heißen abgeschlossen.

Die Menge $V(A)$ enthält die Variablen in $A \in FO(S)$. Ähnlich enthalten $FV(A)$ und $GV(A)$ die Variablen, die gebunden bzw. frei in A vorkommen.

Lemma 4.6

(a) *Ist S entscheidbar, dann sind $Term(S)$ und $FO(S)$ entscheidbar.*

(b) *Zusammengesetzte Terme und Formeln lassen sich eindeutig zerlegen.*

(c) *Freie und gebundene Vorkommen lassen sich effektiv bestimmen.*

Semantik der Prädikatenlogik erster Stufe (Fort.)

Terme und Formeln sind syntaktische Objekte ohne Bedeutung.

Was bedeutet ein Term? Was bedeutet eine Formel?

Definition 4.7 (Struktur)

Sei $S = (\mathit{Funk}, \mathit{Präd})$ eine Signatur. Eine Struktur der Signatur S, auch S-Struktur genannt, ist ein Paar $\mathcal{M} = (D, I)$ bestehend aus

einer nicht-leeren Menge D, dem Datenbereich,

und einer Interpretation I der Funktions- und Prädikatssymbole aus S. Dabei bildet I jedes $f/_k \in \mathit{Funk}$ auf eine k-stellige Funktion

$$I(f) : D^k \to D \qquad \text{(schreibe auch } f^{\mathcal{M}} \text{ statt } I(f))$$

und jedes $p/_k \in \mathit{Präd}$ auf ein k-stelliges Prädikat ab:

$$I(p) : D^k \to \mathbb{B} \qquad \text{(schreibe auch } p^{\mathcal{M}} \text{ statt } I(p)).$$

Semantik der Prädikatenlogik erster Stufe (Fort.)

Annahme: Strukturen sind passend zu Signaturen gewählt.

Beachte: Gleichheit ist ein logisches Symbol, nicht Teil der Signatur. Wird nicht durch die Struktur interpretiert.

Definition 4.8 (Belegung)

Eine Belegung der Variablen in $\mathcal{M} = (D, I)$ ist eine Abbildung

$$\sigma : V \to D.$$

Die Modifikation $\sigma\{x/d\}$ von σ ist die Belegung mit

$$\sigma\{x/d\}(y) := \begin{cases} d, & \text{falls } y = x \\ \sigma(y), & \text{sonst.} \end{cases}$$

Die Menge aller Belegungen wird mit D^V bezeichnet.

Semantik der Prädikatenlogik erster Stufe (Fort.)

Definition 4.9 (Semantik von Termen)

Die Semantik eines Terms $t \in \mathit{Term}(S)$ in $\mathcal{M} = (D, I)$ ist eine Funktion

$$\mathcal{M}[\![t]\!] : D^V \to D,$$

die induktiv wie folgt definiert ist:

$$\mathcal{M}[\![x]\!](\sigma) := \sigma(x)$$
$$\mathcal{M}[\![f(t_1, \ldots, t_k)]\!](\sigma) := f^{\mathcal{M}}(\mathcal{M}[\![t_1]\!](\sigma), \ldots, \mathcal{M}[\![t_k]\!](\sigma)).$$

Dabei ist $\mathcal{M}[\![t]\!](\sigma)$ der Datenwert von t in $\mathcal{M}$ unter Belegung σ.

Semantik der Prädikatenlogik erster Stufe (Fort.)

Definition 4.10 (Semantik von Formeln)

Die Semantik einer Formel $A \in FO(S)$ in $\mathcal{M} = (D, I)$ ist eine Funktion

$$\mathcal{M}[\![A]\!] : D^V \to \mathbb{B},$$

die induktiv wie folgt definiert ist:

$$\mathcal{M}[\![t_1 = t_2]\!](\sigma) := 1 \quad \text{gdw.} \quad \mathcal{M}[\![t_1]\!](\sigma) = \mathcal{M}[\![t_2]\!](\sigma)$$
$$\mathcal{M}[\![p(t_1, \ldots, t_k)]\!](\sigma) := p^{\mathcal{M}}(\mathcal{M}[\![t_1]\!](\sigma), \ldots, \mathcal{M}[\![t_k]\!](\sigma))$$

$\neg, \wedge, \vee, \to, \leftrightarrow$ wie in der Aussagenlogik: $\mathcal{M}[\![\neg A]\!](\sigma) := 1 - \mathcal{M}[\![A]\!](\sigma)$ etc.

$$\mathcal{M}[\![\exists x A]\!](\sigma) := 1 \quad \text{gdw. es gibt } d \in D \text{ mit } \mathcal{M}[\![A]\!](\sigma\{x/d\}) = 1$$
$$\mathcal{M}[\![\forall x A]\!](\sigma) := 1 \quad \text{gdw. für alle } d \in D \text{ gilt } \mathcal{M}[\![A]\!](\sigma\{x/d\}) = 1.$$

Nenne $\mathcal{M}[\![A]\!](\sigma)$ den Wahrheitswert von A in $\mathcal{M}$ unter Belegung σ.

Semantik der Prädikatenlogik erster Stufe (Fort.)

Lemma 4.11 (Koinzidenzlemma)

Betrachte $A \in FO(S)$, $\mathcal{M} = (D, I)$ und $\sigma_1, \sigma_2 \in D^V$. Falls $\sigma_1(x) = \sigma_2(x)$ für alle $x \in FV(A)$, dann gilt

$$\mathcal{M}[\![A]\!](\sigma_1) = \mathcal{M}[\![A]\!](\sigma_2).$$

Insbesondere ist die Semantik $\mathcal{M}[\![A]\!](\sigma)$ geschlossener Formeln $A \in FO(S)$ unabhängig von der Belegung $\sigma \in D^V$:

entweder gilt A unter allen Belegungen oder unter keiner.

Semantik der Prädikatenlogik erster Stufe (Fort.)

Definition 4.12 (Erfüllbarkeit, Tautologie)

Seien $A \in FO(S)$, $\mathcal{M} = (D, I)$ und $\sigma \in D^V$.

- A gilt in $\mathcal{M}$ unter σ, in Zeichen $\mathcal{M}, \sigma \models A$, falls $\mathcal{M}[\![A]\!](\sigma) = 1$.
- Falls A geschlossen, ist die Belegung nach Lemma 4.11 unerheblich. Schreibe $\mathcal{M} \models A$ und sage $\mathcal{M}$ ist ein Modell für A.
- A ist eine Tautologie oder allgemeingültig, in Zeichen $\models A$, falls für alle S-Strukturen $\mathcal{M}$ und alle Belegungen $\sigma \in D^V$ gilt: $\mathcal{M}, \sigma \models A$.
- A ist erfüllbar, falls es eine S-Struktur $\mathcal{M}$ und eine Belegung $\sigma \in D^V$ gibt mit $\mathcal{M}, \sigma \models A$.

Lemma 4.13

Formel $A \in FO(S)$ ist allgemeingültig gdw. $\neg A$ unerfüllbar ist.

Semantik der Prädikatenlogik erster Stufe (Fort.)

Definition 4.14 (Logische Äquivalenz)

Formeln $A, B \in FO(S)$ heißen logisch äquivalent, in Zeichen $A \models\!\models B$, falls für alle Strukturen $\mathcal{M}$ und alle Belegungen σ gilt:

$$\mathcal{M}[\![A]\!](\sigma) = \mathcal{M}[\![B]\!](\sigma).$$

Lemma 4.15

Logische Äquivalenz ist eine Kongruenz: ersetzt man in einer Formel $A \in FO(S)$ eine Teilformel B durch $C \models\!\models B$, dann erhält man $A' \models\!\models A$.

Semantik der Prädikatenlogik erster Stufe (Fort.)

Lemma 4.16 (Logische Äquivalenzen)

Seien $A, B \in FO(S)$. Dann gilt:

$$\neg\forall xA \models\!\models \exists x\neg A \qquad \neg\exists xA \models\!\models \forall x\neg A \tag{8}$$

$$\forall xA \wedge \forall xB \models\!\models \forall x(A \wedge B) \qquad \exists xA \vee \exists xB \models\!\models \exists x(A \vee B) \tag{9}$$

$$\forall x\forall yA \models\!\models \forall y\forall xA \qquad \exists x\exists yA \models\!\models \exists y\exists xA. \tag{10}$$

Falls außerdem $x \notin FV(B)$, gilt

$$QxA \text{ op } B \models\!\models Qx(A \text{ op } B) \text{ mit } Q \in \{\forall, \exists\} \text{ und op} \in \{\wedge, \vee\}. \tag{11}$$

Bemerkung: Äquivalenzen (8), (9) und (11) ziehen Quantoren nach außen.

Bei der Verwendung logischer Äquivalenzen ist Vorsicht geboten:

$$\forall xA \vee \forall xB \not\models\!\models \forall x(A \vee B) \qquad \exists xA \wedge \exists xB \not\models\!\models \exists x(A \wedge B).$$

Substitution

Substitutionen ersetzen Variablen durch Terme.

Sie sind das syntaktische Gegenstück zum semantischen Konzept der Belegungen, genauer deren Modifikation.

Definition 4.17 (Substitution)

Eine Substitution der Signatur S ist eine endliche Abbildung

$$\theta : V \to \mathit{Term}(S).$$

Substitutionen werden oft direkt angegeben als $\theta = \{x_1/t_1, \ldots, x_n/t_n\}$.

Substitution (Fort.)

Die Anwendung von Substitutionen auf Terme und Formeln vermeidet es, neue Bindungen einzuführen.

Definition 4.18 (Anwendung von Substitutionen)

Die Anwendung von θ auf $t \in \mathit{Term}(S)$ liefert einen neuen Term $t\theta \in \mathit{Term}(S)$, der induktiv wie folgt definiert ist:

$$x\theta := \theta(x) \qquad f(t_1, \ldots, t_n)\theta := f(t_1\theta, \ldots, t_n\theta).$$

Für $A \in \mathit{FO}(S)$ liefert die Anwendung von θ die Formel $A\theta \in \mathit{FO}(S)$ mit

$$(t_1 = t_2)\theta := t_1\theta = t_2\theta \qquad (\neg A)\theta := \neg(A\theta)$$
$$p(t_1, \ldots, t_n)\theta := p(t_1\theta, \ldots, t_n\theta) \qquad (A \text{ op } B)\theta := A\theta \text{ op } B\theta$$
$$(Qx.A)\theta := Qy(A\{x/y\}\theta),$$

wobei $y \notin V(A) \cup \mathit{Dom}(\theta) \cup \mathit{Ran}(\theta)$.

Substitution (Fort.)

Der Zusammenhang zwischen Substitutionen und der Modifikation von Belegungen ist wie folgt:

Lemma 4.19 (Substitutionslemma)

$\mathcal{M}[\![A\{x/t\}]\!](\sigma) = \mathcal{M}[\![A]\!](\sigma\{x/\mathcal{M}[\![t]\!](\sigma)\})$.

Der Beweis wird mittels Induktion über den Aufbau von Termen und Formeln geführt.

Korollar 4.20

i) *Ist* $A \in FO(S)$ *allgemeingültig, dann auch* $A\{x/t\}$.

ii) *Die Formel* $\forall x.A \to A\{x/t\}$ *ist allgemeingültig.*

Substitution (Fort.)

Ähnlich zum Substitutionslemma erhält man:

Lemma 4.21 (Gebundene Umbenennung erhält logische Äquivalenz)

$QxA \models\models Qy(A\{x/y\})$.

Bemerkung: Gebundene Umbenennung kann die Vorkommen von gebundenen Variablen in einer Formel eindeutig machen.

Normalformen

Erzeuge Formeln von einfacherer Gestalt, für die sich Aussagen eher zeigen und effizientere Algorithmen entwerfen lassen.

Pränexnormalform: Alle Quantoren außen (bis auf logische Äquivalenz).
Skolemform: Pränexnormalform und außerdem nur Allquantoren (bis auf Erfüllbarkeitsäquivalenz).

Lemma 4.22 (Existentieller und universeller Abschluss)

Betrachte $A \in FO(S)$ mit $FV(A) = \{x_1, \ldots, x_n\}$. Dann gilt

$$A \text{ ist allgemeingültig} \quad \text{gdw.} \quad \forall x_1 \ldots \forall x_n . A \text{ ist allgemeingültig}$$
$$A \text{ ist erfüllbar} \quad \text{gdw.} \quad \exists x_1 \ldots \exists x_n . A \text{ ist erfüllbar.}$$

Formel $\forall x_1 \ldots \forall x_n . A$ ist der universelle Abschluss von A.
Formel $\exists x_1 \ldots \exists x_n . A$ ist der existentielle Abschluss von A.

Normalformen (Fort.)

Eine Formel $A \in FO(S)$ heißt bereinigt, falls

i) keine Variable frei und gebunden vorkommt und

ii) jede Variable höchstens einmal gebunden wird.

Durch wiederholte Anwendung gebundener Umbenennung in Lemma 4.21 kann man jede Formel bereinigen.

Lemma 4.23

Zu jeder Formel $A \in FO(S)$ gibt es eine bereinigte Formel $B \in FO(S)$ mit $A \models\!\!\models B$.

Normalformen (Fort.)

Nächstes Ziel: Ziehe Quantoren nach außen.
Trick: Nutze Äquivalenzen aus Lemma 4.16.

Definition 4.24

Eine Formel der Gestalt $A \equiv Q_1 y_1 \dots Q_n y_n . B$ ist in Pränexnormalform, wobei $Q_1, \dots, Q_n \in \{\forall, \exists\}$ und B quantorenfrei.

Sage $A \in FO(S)$ ist in BPF, falls A bereinigt und in Pränexnormalform ist.

Satz 4.25

Zu jeder Formel $A \in FO(S)$ gibt es eine Formel $B \in FO(S)$ in BPF mit $A \models\!\!\models B$.

Hinter dem Beweis (Tafel) verbirgt sich ein rekursiver Algorithmus.
Es dient dem Verständnis, dieses Verfahren selbst herauszuarbeiten.

Normalformen (Fort.)

Letzter Schritt: Eliminiere Existenzquantoren.

Trick: Stelle Schachtelung der Quantoren für alle $y_1 \ldots y_n$ gibt es ein z als Funktion $z = f(y_1, \ldots, y_n)$ dar:

$$\forall y_1 \ldots \forall y_n \exists z.A \quad \text{ergibt} \quad \forall y_1 \ldots \forall y_n.\,(A\{z/f(y_1, \ldots, y_n)\})\,.$$

Dabei ist $f/_n$ ein frisches Funktionssymbol aus der Menge *Sko* der Skolemfunktionen. Frisch heißt, *Sko* ist disjunkt von S.

Die Einführung von Skolemfunktionen für existenzquantifizierte Variablen wird Skolemisierung genannt.

Skolemisierung erhält nur Erfüllbarkeitsäquivalenz, logische Äquivalenz muss aufgegeben werden.

Skolemisierung geht zurück auf Thoralf Albert Skolem (1887 – 1963).

Normalformen (Fort.)

Definition 4.26 (Skolemformel)

Für eine Formel $A \in FO(S)$ in BPF ist die Skolemformel $B \in FO(S \uplus Sko)$ (wieder in BPF) durch folgendes Verfahren definiert:

while A hat Existenzquantoren **do**
  Sei $A \equiv \forall y_1 \ldots \forall y_n \exists z.B$ mit B in BPF
  Sei $f/n \in Sko$ ein Skolemsymbol außerhalb von B
  Setze $A \equiv \forall y_1 \ldots \forall y_n (B\{z/f(y_1, \ldots, y_n)\})$
end while

Beachte: Skolemfunktionen werden von außen nach innen eingeführt.

Satz 4.27 (Skolem)

Für jede Formel $A \in FO(S)$ in BPF und die zugehörige Skolemformel $B \in FO(S \uplus Sko)$ gilt:

$$A \text{ ist erfüllbar} \quad \text{gdw.} \quad B \text{ ist erfüllbar.}$$

Das Allgemeingültigkeitsproblem

Untersuche Berechenbarkeit des **Allgemeingültigkeitsproblems**:

Gegeben: Eine Formel $A \in FO(S)$.
Frage: Ist A allgemeingültig?

Ziel: Allgemeingültigkeit ist **vollständig in der Klasse der semi-entscheidbaren Probleme**. Genauer:

Obere Schranke:
Allgemeingültigkeit ist semi-entscheidbar.

Untere Schranke:
Das Allgemeingültigkeitsproblem ist hart in der Klasse der semi-entscheidbaren Probleme.

Insbesondere ist Allgemeingültigkeit **unentscheidbar**.

Herbrand-Theorie

Um semi-Entscheidbarkeit der Allgemeingültigkeit zu zeigen, nutze

$$A \in FO(S) \text{ ist allgemeingültig} \quad \text{gdw.} \quad \neg A \text{ ist unerfüllbar.}$$

Ziel: Unerfüllbarkeit ist semi-entscheidbar.

Problem: Bei der Wahl von $\mathcal{M} = (D, I)$ ist der Datenbereich beliebig.

- Keine Aussage über die Mächtigkeit von D.
- Keine Information über die Struktur von I.

Wie soll man Strukturen aufzählen und auf Modelleigenschaft prüfen?

Kernbeitrag: Die Suche nach Modellen kann auf **kanonische Strukturen** eingeschränkt werden.
Um ein Modell für A zu finden, **genügt es**, in folgendem Datenbereich zu suchen

$$D_{\mathcal{H}} = \text{Alle variablenfreien Terme über Signatur } S.$$

Herbrand-Theorie (Fort.)

Annahme: $FO^{\neq}(S)$ mit $S = (Funk, Präd)$, wobei $Funk$ eine Konstante enthält.

Definition 4.28 (Herbrand-Struktur)

Eine Struktur $\mathcal{H}$ von S heißt Herbrand-Struktur, falls $\mathcal{H} = (D_{\mathcal{H}}, I_{\mathcal{H}})$.
Dabei ist $D_{\mathcal{H}}$ die kleinste Menge, für die gilt:

i) Falls $a/_0 \in Funk$, dann $a \in D_{\mathcal{H}}$

ii) Falls $f/_n \in Funk$ und $t_1, \ldots, t_n \in D_{\mathcal{H}}$, dann $f(t_1, \ldots, t_n) \in D_{\mathcal{H}}$.

Die Interpretation $I_{\mathcal{H}}(f) : D_{\mathcal{H}}^n \to D_{\mathcal{H}}$ der Funktionssymbole $f/_n \in Funk$ ist festgelegt als

$$I_{\mathcal{H}}(f)(t_1, \ldots, t_n) := f(t_1, \ldots, t_n).$$

Die Interpretation der Prädikatssymbole ist noch offen, eine Herbrand-Struktur muss nur diesen beiden Einschränkungen genügen.

Herbrand-Theorie (Fort.)

Gegeben sei eine geschlossene Formel $A \in FO^{\neq}(S)$. Eine Herbrand-Struktur $\mathcal{H}$ mit $\mathcal{H} \models A$ heißt auch **Herbrand-Modell** von A.

Satz 4.29 (Herbrand)

Sei $A \in FO^{\neq}(S)$ eine geschlossene Formel in Skolemnormalform. Dann gilt:

A ist erfüllbar **gdw.** *A hat ein* **Herbrand-Modell**.

Korollar 4.30 (Satz von Löwenheim-Skolem)

Sei $A \in FO(S)$ erfüllbar. Dann besitzt A ein Modell $\mathcal{M} = (D, I)$, dessen Datenbereich D **abzählbar** *ist.*

Semi-Entscheidbarkeit der Allgemeingültigkeit

Definition 4.31 (Herbrand-Expansion)

Sei $A \equiv \forall y_1 \ldots \forall y_n . B \in FO^{\neq}(S)$ geschlossen und in Skolemnormalform. Dann ist die Herbrand-Expansion $E(A)$ von A definiert als

$$E(A) := \{B\{y_1/t_1\} \ldots \{y_n/t_n\} \quad | \quad t_1, \ldots, t_n \in D_{\mathcal{H}}\}.$$

Es werden also alle Variablen in B durch Terme in $D_{\mathcal{H}}$ ersetzt.

Beobachtung:

- Die Formeln in $E(A)$ lassen sich wie aussagenlogische Formeln behandeln, da sie keine Variablen enthalten.
- Betrachte Herbrand-Struktur zur Interpretation der Formeln in $E(A)$.
- Sie gibt Wahrheitswerte der aussagenlogischen Variablen in $E(A)$ an.

Semi-Entscheidbarkeit der Allgemeingültigkeit (Fort.)

Satz 4.32 (Gödel-Herbrand-Skolem)

Für eine geschlossene Formel $A \in FO^{\neq}(S)$ in Skolemnormalform gilt:

$$A \text{ ist erfüllbar} \quad \text{gdw.} \quad E(A) \text{ ist aussagenlogisch erfüllbar.}$$

Intuition: Die prädikatenlogische Formel A wird durch die aussagenlogischen Formeln in $E(A)$ approximiert.

Kombiniere Satz 4.32 mit dem Kompaktheitssatz der Aussagenlogik.

Korollar 4.33

Eine geschlossene Formel $A \in FO^{\neq}(S)$ in Skolemnormalform ist unerfüllbar gdw. es eine **endliche** *Teilmenge von $E(A)$ gibt, die unerfüllbar ist.*

Semi-Entscheidbarkeit der Allgemeingültigkeit (Fort.)

Damit folgt semi-Entscheidbarkeit der Allgemeingültigkeit:

$$A \in FO(S) \text{ ist allgemeingültig} \quad \text{gdw.} \quad \neg A \text{ ist unerfüllbar.}$$

Überführe $\neg A$ in eine geschlossene Formel $B \in FO^{\neq}(S \uplus Sko)$ in Skolemnormalform.

Obige Argumentation liefert den **Algorithmus von Gilmore**, der Unerfüllbarkeit von B semi-entscheidet.

Semi-Entscheidbarkeit der Allgemeingültigkeit (Fort.)

Algorithmus von Gilmore:

Input: $A \in FO^{\neq}(S \uplus Sko)$ geschlossen und in Skolemnormalform.
Sei $E(A) = \{A_1, A_2, \ldots\}$ eine Aufzählung von $E(A)$.

n:=1
while $A_1 \wedge \ldots \wedge A_n$ ist aussagenlogisch erfüllbar **do**
  n:=n+1
end while
return unerfüllbar

Mit Korollar 4.33:
Terminiert und liefert korrektes Ergebnis auf unerfüllbaren Formeln.
Terminiert nicht auf erfüllbaren Formeln.

Satz 4.34

Das Allgemeingültigkeitsproblem ist semi-entscheidbar.

Semi-Entscheidbarkeit der Allgemeingültigkeit (Fort.)

Beachten Sie, dass aus Semi-Entscheidbarkeit der Allgemeingültigkeit **nicht** mittels Negation der Formel die Entscheidbarkeit folgt:

$$\not\models A \quad \text{ist } \textbf{nicht} \text{ äquivalent zu} \quad \models \neg A.$$

Nur Letzteres lässt sich mittels Herbrand-Expansion prüfen.

Zeige nun tatsächlich **Unentscheidbarkeit** der Allgemeingültigkeit.

Untere Schranke für Allgemeingültigkeit

Ziel: Das Allgemeingültigkeitsproblem ist **hart** in der Klasse der semi-entscheidbaren Probleme.

Das heißt, **jedes** semi-entscheidbare Problem besitzt eine many-one Reduktion auf Allgemeingültigkeit.

Konsequenz: Allgemeingültigkeit ist **unentscheidbar** (Halteproblem).

Definition 4.35 (Many-one Reduktion)

Eine **many-one Reduktion** eines Problems P_1 auf ein Problem P_2 ist eine totale und berechenbare Funktion $f : P_1 \to P_2$, die Instanzen des Problems P_1 auf Instanzen des Problems P_2 abbildet, so dass

$$\text{Instanz } K \text{ von } P_1 \text{ hat Lösung} \quad \text{gdw.} \quad \text{Instanz } f(K) \text{ von } P_2 \text{ hat Lösung.}$$

Untere Schranke für Allgemeingültigkeit (Fort.)

Wie beweist man Härte von Allgemeingültigkeit? Es ist eine allquantifizierte Aussage.

Betrachte ein bereits als hart bekanntes Problem. Wähle hier das **Postsche Korrespondenzproblem (PCP)**.

Gib eine many-one Reduktion **von PCP auf Allgemeingültigkeit** an.

Warum zeigt diese Reduktion Härte von Allgemeingültigkeit?

Sei P ein semi-entscheidbares Problem und f_P dessen Reduktion auf PCP. Reduktion f_P existiert, da PCP hart ist.

Sei f die Reduktion von PCP auf Allgmeingültigkeit, die noch zu finden ist.

Dann gilt:

$$\text{P} \xrightarrow{f_P} \text{PCP} \xrightarrow{f} \text{Allgemein} \quad \text{impliziert} \quad \text{P} \xrightarrow{f \circ f_P} \text{Allgemein.}$$

Das Postsche Korrespondenzproblem

Gegeben: Eine endliche Folge von Wortpaaren

$$((x_1, y_1), \ldots, (x_n, y_n)) \quad \text{mit} \quad x_i, y_i \in \{0,1\}^+.$$

Frage: Gibt es eine nicht-leere Folge $i_1, \ldots, i_k \in \{1, \ldots, n\}$ mit

$$x_{i_1} \ldots x_{i_k} = y_{i_1} \ldots y_{i_k}.$$

Eine gegebene Folge von Wortpaaren ist eine **PCP-Instanz** K.
Eine Folge von Indizes $i_1, \ldots, i_k$, die obiger Gleichheit genügt, heißt **Lösung der Instanz** K.

Satz 4.36 (Post 1946)

PCP ist vollständig in der Klasse der semi-entscheidbaren Probleme, also

(a) *PCP ist semi-entscheidbar und*

(b) *jedes semi-entscheidbare Problem besitzt eine many-one Reduktion auf PCP. Insbesondere ist PCP also* **unentscheidbar**.

Untere Schranke für Allgemeingültigkeit (Fort.)

Satz 4.37 (Church)

Das Allgemeingültigkeitsproblem ist **hart** *— und mit Satz 4.34 also* **vollständig** *— in der Klasse der semi-entscheidbaren Probleme.*

Korollar 4.38

Das Allgemeingültigkeitsproblem ist unentscheidbar.

Kompaktheitssatz der Prädikatenlogik erster Stufe

Definition 4.39 (Semantik von Formelmengen)

Sei S eine Signatur, $\Sigma \subseteq FO(S)$, $\mathcal{M} = (D, I)$ und $\sigma \in D^V$.

(i) Σ gilt in $\mathcal{M}$ unter σ, in Zeichen $\mathcal{M}, \sigma \models \Sigma$, falls für alle $A \in \Sigma$ gilt $\mathcal{M}[\![A]\!](\sigma) = 1$.

(ii) Σ ist erfüllbar, falls es $\mathcal{M}$ und σ gibt mit $\mathcal{M}, \sigma \models \Sigma$.

Satz 4.40 (Kompaktheitssatz)

Eine Formelmenge $\Sigma \subseteq FO(S)$ ist erfüllbar gdw. jede endliche Teilmenge von Σ erfüllbar ist.

Logische Folgerung

Definition 5.1 (Logische Folgerung)

Formel $A \in FO(S)$ ist eine logische Folgerung aus $\Sigma \subseteq FO(S)$, in Zeichen $\Sigma \models A$, falls für alle $\mathcal{M}$ und σ gilt:

$$\mathcal{M}, \sigma \models \Sigma \quad \text{impliziert} \quad \mathcal{M}, \sigma \models A.$$

Die Menge der Folgerungen aus Σ ist

$$\text{Folg}(\Sigma) := \{A \in FO(S) \mid \Sigma \models A\}.$$

Formelmengen $\Sigma \subseteq FO(S)$ und $\Gamma \subseteq FO(S)$ sind äquivalent, in Zeichen $\Sigma \models\!\models \Gamma$, falls

$$\Sigma \models A \text{ für alle } A \in \Gamma \qquad \text{und} \qquad \Gamma \models B \text{ für alle } B \in \Sigma.$$

Logische Folgerung (Fort.)

Bemerkung 5.2

(a) $\Sigma \models A$ *gdw.* $\Sigma \cup \{\neg A\}$ *nicht erfüllbar.*

(b) $\emptyset \models A$ *gdw.* $\models A$, *also* A *ist allgemeingültig.*

(c) Σ *nicht erfüllbar gdw.* $\Sigma \models A$ *für alle* $A \in FO(S)$.

(d) *Falls* $\Gamma \subseteq \Sigma$ *und* $\Gamma \models A$, *dann* $\Sigma \models A$.

(e) *Falls* $\Gamma \models\!\models \Sigma$, *dann ist* Γ *erfüllbar gdw.* Σ *erfüllbar ist.*

(f) *Falls* $\Gamma \models\!\models \Sigma$, *dann* $Folg(\Gamma) = Folg(\Sigma)$.

(g) $A \models\!\models B$ *gdw.* $A \models B$ *und* $B \models A$ *gdw.* $\models A \leftrightarrow B$ *gdw.* $\mathcal{M}[\![A]\!](\sigma) = \mathcal{M}[\![B]\!](\sigma)$ *für alle* $\mathcal{M}, \sigma$.

(h) *Falls* $A \models\!\models B$, *dann* $\Sigma \models A$ *gdw.* $\Sigma \models B$.

Beispiele

Beispiel 5.3

i) $\forall x A \models A$
Spezialfall von $\forall x A \to A\{x/t\}$ allgemeingültig.

ii) Im Allgemeinen ist $A \models \forall y A$ mit $y \in FV(A)$ nicht gültig.

Sei $A \equiv p(y)$ und $\mathcal{M} = (\{0,1\}, I)$ mit $I(p)(a) = 1$ gdw. $a = 0$.
Wähle $\sigma(y) = 0$, dann $\mathcal{M}[\![A]\!](\sigma) = 1$.
Aber $\mathcal{M}[\![\forall y A]\!](\sigma) = 0$ mit $\sigma\{y/1\}$.

iii) $\models \exists x(p(x) \to \forall x p(x))$
Sei $\mathcal{M} = (D, I)$. Es gilt $\mathcal{M}[\![\exists x(p(x) \to \forall x p(x))]\!] = 1$, falls

es gibt $d \in D$ mit $I(p)(d) = 0$ oder
für alle $d \in D$ gilt $I(p)(d) = 1$.

Eines von beiden gilt.

iv) $\forall x(A \to B) \models \forall x A \to \forall x B$

Logische Folgerung (Fort.)

Satz 5.4 (Wichtige Sätze)

Seien $\Gamma \subseteq FO(S)$ *und* $A, B \in FO(S)$.

Deduktionstheorem $\Gamma, A \models B$ *gdw.* $\Gamma \models A \to B$

Modus-Ponens-Regel $\Gamma \models A$ *und* $\Gamma \models A \to B$, *dann* $\Gamma \models B$

Kontrapositionsregel $\Gamma, A \models \neg B$ *gdw.* $\Gamma, B \models \neg A$

Generalisierungstheorem

Kommt $x \in V$ *in keiner Formel von* Γ *frei vor*, *dann*

$\Gamma \models A$ *gdw.* $\Gamma \models \forall x\ A$

Insbesondere: $A \models \forall x\ A$ *bzw.* $\models A \to \forall x\ A$,

falls x *nicht frei in* A *vorkommt.*

Logische Folgerung (Fort.)

Beispiel 5.5 (Anwendung der Sätze)

a) $\models \exists x \forall y A \to \forall y \exists x A$

gdw.	$\exists x \forall y A \models \forall y \exists x A$	Deduktionstheorem
gdw.	$\exists x \forall y A \models \exists x A$	Generalisierungstheorem
gdw.	$\neg \forall x \neg \forall y A \models \neg \forall x \neg A$	Bemerkung 5.2 (logische Äquivalenz)
gdw.	$\forall x \neg A \models \forall x \neg \forall y A$	Kontrapositionsregel
gdw.	$\forall x \neg A \models \neg \forall y A$	Generalisierungstheorem
gdw.	$\{\forall x \neg A, \forall y A\}$	nicht erfüllbar

b) Variante von Kongruenz

A' entstehe aus A durch erlaubte (beachte Quantoren) Ersetzung einiger Vorkommen von x durch y. Dann gilt

$$\models \forall x \forall y (x = y \to (A \leftrightarrow A')).$$

Beispiel: $\forall x \forall y (x = y \to (f(x, y) = g(x) \leftrightarrow f(y, y) = g(x))$

Das deduktive System $\mathcal{F}$

Ziel: Konstruiere ein geeignetes deduktives System $\mathcal{F}(Ax, R)$ für die Prädikatenlogik erster Stufe.

Geeignet: Korrektheit ($\Rightarrow$) und **Vollständigkeit ($\Leftarrow$)**

$$\vdash_{\mathcal{F}} A \quad \text{gdw.} \quad \models A$$

$$\Sigma \vdash_{\mathcal{F}} A \quad \text{gdw.} \quad \Sigma \models A$$

Die Definition von System $\mathcal{F}$ zusammen mit dem Beweis der Vollständigkeit ist ein großer Beitrag von Kurt Gödel (1906 — 1978).

Das deduktive System $\mathcal{F}$ (Fort.)

Sei $FO_0(S)$ die Teilmenge der Formeln aus $FO(S)$ über $\neg, \to, \forall, =$.

Definition 5.6 (Deduktive Systeme)

Das **deduktive System** $\mathcal{F}(Ax, R)$ für $FO_0(S)$ besteht aus den Axiomen, die als Generalisierungen der durch folgende Schemata beschriebenen Formeln gewonnen werden können:

Ax1: Aussagenlogische Tautologien
Ax2: $\forall x\ A \to A\{x/t\}$
Ax3: $\forall x\ (A \to B) \to (\forall x\ A \to \forall x\ B)$
Ax4: $A \to \forall x\ A$, falls $x \notin FV(A)$
Ax5: $x = x$
Ax6: $x = y \to (A \to A')$, wobei A' aus A durch Ersetzen einiger freier Vorkommen von x durch y entsteht (sofern erlaubt).

Das einzige Regelschema ist **Modus Ponens**: $\dfrac{A, A \to B}{B}$

Das deduktive System $\mathcal{F}$ (Fort.)

Definition 5.7 (und Bemerkung)

Seien $A \in FO(S)$ und $\{x_1, \ldots, x_n\} \subseteq FV(A)$. Die Formel

$$\forall x_1 \ldots \forall x_n . A$$

ist eine Generalisierung von A.

Mit Satz 2.12 lassen sich alle aussagenlogischen Tautologien aus drei Axiomenschemata mit Modus Ponens herleiten.

Deduktionstheorem und Generalisierungstheorem

Satz 5.8

Seien $\Gamma \subseteq FO(S)$ *und* $A, B \in FO(S)$.

a) **Deduktionstheorem**

$$\Gamma \underset{\mathcal{F}}{\vdash} A \to B \qquad \textit{gdw.} \qquad \Gamma, A \underset{\mathcal{F}}{\vdash} B$$

b) **Generalisierungstheorem:**

Falls $\Gamma \underset{\mathcal{F}}{\vdash} A$ *und* x *nicht frei in* Γ *vorkommt, so* $\Gamma \underset{\mathcal{F}}{\vdash} \forall x\, A$

c) **Kontrapositionstheorem:**

$$\Gamma, A \vdash \neg B \qquad \textit{gdw.} \qquad \Gamma, B \vdash \neg A.$$

Es gelten also für System $\mathcal{F}$ die für das deduktive System $\mathcal{F}_0$ der Aussagenlogik bekannten Sätze.

Konsistenz

Definition 5.9

Eine Formelmenge $\Gamma \subseteq FO(S)$ heißt **konsistent**, falls es kein $A \in FO(S)$ gibt mit $\Gamma \underset{\mathcal{F}}{\vdash} A$ und $\Gamma \underset{\mathcal{F}}{\vdash} \neg A$.

Bemerkung 5.10

- Γ *ist konsistent gdw. jede endliche Teilmenge von* Γ *konsistent ist.*
- *Ist* Γ *inkonsistent, dann gilt* $\Gamma \underset{\mathcal{F}}{\vdash} A$ *für jede Formel* A*.*
- *Gilt* $\Gamma \vdash A$*, dann ist* $\Gamma \cup \{\neg A\}$ *inkonsistent.*
- *Ist* Γ *inkonsistent, so ist* Γ *nicht erfüllbar.*
- *Die Menge der allgemeingültigen Formeln ist konsistent.*
- *Die Menge der Theoreme von* $\mathcal{F}$ *ist konsistent.*

Das deduktive System $\mathcal{F}$ (Fort.)

Satz 5.11 (Korrektheit und Vollständigkeit von $\mathcal{F}$, Gödel)

Seien $A \in FO(S)$ und $\Sigma \subseteq FO(S)$, dann gilt:

a) $\vdash_{\mathcal{F}} A$ *gdw.* $\models A$.

b) $\Sigma \vdash_{\mathcal{F}} A$ *gdw.* $\Sigma \models A$.

c) Σ *konsistent* *gdw.* Σ *erfüllbar.*

Der Satz der Prädikatenlogik!

Beweis:

Korrektheit: Ax enthält nur allgemeingültige Formeln und (MP) führt nicht aus der Menge der allgemeingültigen Formeln hinaus.

Vollständigkeit: Siehe Enderton. ■

Theorien erster Stufe

Betrachte geschlossene Formeln aus $FO_{abg}(S)$.

Definition 5.12

Sei S eine Signatur. Eine Formelmenge $\Gamma \subseteq FO_{abg}(S)$ heißt **Theorie erster Stufe**, falls Γ abgeschlossen gegenüber logischer Folgerung ist:

$$A \in FO_{abg}(S) \quad \text{und} \quad \Gamma \models A \quad \text{impliziert} \quad A \in \Gamma.$$

Nutze T als Bezeichner für Theorien.

Alternative Definitionen in der Literatur:

- Γ Menge an Formeln aus $FO(S)$ anstatt $FO_{abg}(S)$, abgeschlossen gegen logische Folgerung.
- Γ Theorie, falls Γ abgeschlossen gegen MP und Generalisierung.

Theorien erster Stufe (Forts.)

Bemerkung 5.13

Sei S eine Signatur.

a) $T_S = \{A \in FO_{abg}(S) \mid A \text{ allgemeingültig}\}$ *ist eine Theorie.*

b) *Sei $\Sigma \subseteq FO_{abg}(S)$. Dann ist $T_\Sigma = \{A \in FO_{abg}(S) \mid \Sigma \models A\}$ die* **von Σ erzeugte Theorie** *oder durch die Axiome Σ definierte Theorie.*

c) *Sei $\mathcal{M}$ eine Struktur der Signatur S. Dann ist*

$$T_\mathcal{M} = \{A \in FO_{abg}(S) \mid \mathcal{M} \models A\}$$

die **Theorie von** $\mathcal{M}$*. Es ist auch $Th(\mathcal{M})$ als Symbol üblich.*

Theorien erster Stufe (Forts.)

Lemma 5.14 (und Definition)

(i) *Ist T eine Theorie und $A \in FO_{abg}(S)$, dann gilt*

$$T \underset{\mathcal{F}}{\vdash} A \quad \text{gdw.} \quad A \in T.$$

(ii) *Theorie T heißt* **inkonsistent**, *falls es eine Formel $A \in FO_{abg}(S)$ gibt mit $T \underset{\mathcal{F}}{\vdash} A$ und $T \underset{\mathcal{F}}{\vdash} \neg A$. In dem Fall gilt $T = FO_{abg}(S)$.*

(iii) *$T_{\mathcal{M}}$ ist* **konsistent** *für jede Struktur $\mathcal{M}$.*

(iv) *T_S ist in jeder Theorie über S enthalten.*

Theorien erster Stufe (Forts.)

Definition 5.15

Sei T eine Theorie erster Stufe über Signatur S.

a) T heißt **vollständig**, falls für jede Formel $A \in FO_{abg}(S)$ gilt: $A \in T$ oder $\neg A \in T$.

b) T heißt **(endlich, aufzählbar) axiomatisierbar**, falls es eine (endliche, aufzählbare) Teilmenge $\Sigma \subseteq FO_{abg}(S)$ gibt mit

$$T_\Sigma = T.$$

c) T heißt **entscheidbar**, falls T eine entscheidbare Teilmenge von $FO_{abg}(S)$ ist.

Theorien erster Stufe (Forts.)

Bemerkung 5.16

(a) *$T_{\mathcal{M}}$ ist* **vollständig** *für jede Struktur $\mathcal{M}$.*
Mit Lemma 5.14 ist $T_{\mathcal{M}}$ also konsistent und vollständig.

(b) *T ist erfüllbar gdw. T ist konsistent.*

(c) *Ist T aufzählbar axiomatisierbar, dann ist T aufzählbar.*

(d) *Ist T vollständig und aufzählbar axiomatisierbar, dann ist T* **entscheidbar**.

(e) *Ist T vollständig und konsistent, dann $T = T_{\mathcal{M}}$ für eine Struktur $\mathcal{M}$.*

Axiomatisierung

Ziel: Finde Axiomatisierungen wichtiger Theorien.

Insbesondere: Wann gilt $T_{\mathcal{M}} = T_\Sigma$ für Σ **aufzählbar**.

Motivation: Entscheidbarkeit!

Problem: Wann ist T_Σ vollständig für aufzählbare Σ?

Axiomatisierung (Presburger)

Betrachte die Signatur der Arithmetik **ohne Multiplikation**:

$$S_{PA} = (\{0/_0, 1/_0, +/_2\}, \{=/_2\}).$$

Die zugehörige Struktur $\mathcal{M}_{PA} = (\mathbb{N}, I_{PA})$ mit der erwarteten Interpretation wird **Presburger-Arithmetik** genannt.

Sei Σ_{PA} die Menge der folgenden Axiome, mit (Induktion) ein Schema:

$$\forall x : \neg(x + 1 = 0) \qquad \text{(Null)}$$

$$\forall x : x + 0 = x \qquad \text{(Plus Null)}$$

$$\forall x \forall y : x + 1 = y + 1 \rightarrow x = y \qquad \text{(Nachfolger)}$$

$$\forall x \forall y : x + (y + 1) = (x + y) + 1 \qquad \text{(Plus Nachfolger)}$$

$$A(0) \wedge (\forall x : \ A(x) \rightarrow A(x + 1)) \rightarrow \forall x : \ A(x), \qquad \text{(Induktion)}$$

wobei $A \in FO(S_{PA})$ eine Formel mit einer freien Variablen ist.

Axiomatisierung (Presburger)

Satz 5.17 (Vollständige Axiomatisierung der Presburger-Arithmetik)

Es gilt $T_{\mathcal{M}_{PA}} = T_{\Sigma_{PA}}$. Da Σ_{PA} aufzählbar ist, ist $T_{\mathcal{M}_{PA}}$ **entscheidbar**.

Vollständigkeit der Axiomatisierung ist anspruchsvoll.
Entscheidbarkeit folgt mit Bemerkung 5.16(d).

Es lassen sich also geschlossene Formeln aus $FO(S_{PA})$ **automatisch** auf Gültigkeit in Presburger-Arithmetik prüfen.

Zum Beispiel:

$$\forall w \forall x \exists y \exists z : x + 2y + 3w = z + 13 \quad ?$$

Man beachte die Quantoren und vergleiche mit Gauß-Elimination.

Axiomatisierung (Gödel und Peano)

Betrachte die Signatur der vollen Arithmetik:

$$S_{Arith} = (\{0/0, 1/0, +/2, \cdot/2\}, \{=/2\}).$$

Die zugehörige Struktur $\mathcal{M}_{Arith} = (\mathbb{N}, I_{Arith})$ mit der erwarteten Interpretation wird **(First-Order-)Arithmetik** genannt.

Satz 5.18 (Gödel)

$T_{\mathcal{M}_{Arith}}$ ist nicht entscheidbar.

Konsequenz 1: $T_{\mathcal{M}_{Arith}}$ ist **nicht** *aufzählbar axiomatisierbar.*

Konsequenz 2: Jedes aufzählbare Axiomensystem für $T_{\mathcal{M}_{Arith}}$ ist **unvollständig**.

Die Konsequenzen ergeben sich aus Bemerkung 5.16(a) und (d).

Axiomatisierung (Gödel und Peano)

Insbesondere sind die Peano-Axiome Σ_{Peano} **keine** vollständige Axiomatisierung von $T_{\mathcal{M}_{Arith}}$:

$$\forall x : \neg(x + 1 = 0) \qquad \text{(Null)}$$

$$\forall x : x + 0 = x \qquad \text{(Plus Null)}$$

$$\forall x \forall y : x + 1 = y + 1 \rightarrow x = y \qquad \text{(Nachfolger)}$$

$$\forall x \forall y : x + (y + 1) = (x + y) + 1 \qquad \text{(Plus Nachfolger)}$$

$$A(0) \wedge (\forall x : \ A(x) \rightarrow A(x + 1)) \rightarrow \forall x : \ A(x) \qquad \text{(Induktion)}$$

$$\forall x : x \cdot 0 = 0 \qquad \text{(Mal Null)}$$

$$\forall x \forall y : x \cdot (y + 1) = x \cdot y + x \qquad \text{(Mal Nachfolger)}$$

Es gibt also geschlossene Formeln $A \in FO(S_{Arith})$ mit $\mathcal{M}_{Arith} \models A$, für die $\Sigma_{Peano} \models A$ **nicht** gilt.

Wie kann das sein? $T_{\Sigma_{Peano}}$ besitzt **Nicht-Standard-Modelle**!

Axiomatisierung (Arrays)

Gegeben seien Funktionen für Lese- und Schreibzugriffe auf Arrays:

$$S_{McC} = (\{\text{read}/_2, \text{write}/_3\}, \{=/_2\}).$$

Betrachte **McCarthys Array-Axiome** Σ_{McC}:

$$\forall x : x = x \qquad \text{(Reflexivität)}$$

$$\forall x \forall y : x = y \rightarrow y = x \qquad \text{(Symmetrie)}$$

$$\forall x \forall y \forall z : x = y \wedge y = z \rightarrow x = z \qquad \text{(Transitivität)}$$

$$\forall a \forall i \forall j : i = j \rightarrow \text{read}(a, i) = \text{read}(a, j) \qquad \text{(Array-Kongruenz)}$$

$$\forall a \forall v \forall i \forall j : i = j \rightarrow \text{read}(\text{write}(a, i, v), j) = v \qquad \text{(Read-Write 1)}$$

$$\forall a \forall v \forall i \forall j : i \neq j \rightarrow \text{read}(\text{write}(a, i, v), j) = \text{read}(a, j). \qquad \text{(Read-Write 2)}$$

Satz 5.19

$T_{\Sigma_{McC}}$ ist **nicht entscheidbar**, *insbesondere also* **nicht vollständig**.
Entscheidbare Fragmente sind aktives Forschungsgebiet, Aaron Bradley'06.

Algorithmen der Prädikatenlogik

Ziel: **Praktische** Semi-Entscheidungsverfahren für Unerfüllbarkeit.

Anwendungen:

- **Allgemeingültigkeit**: $\models A$ gdw. $\neg A$ unerfüllbar.
- **Logische Folgerung**: $\Sigma \models A$ gdw. $\Sigma \cup \{\neg A\}$ unerfüllbar.

Idee: Systematische Version des Algorithmus von Gilmore.

Erzeuge gezielt Grundformeln, um Widersprüche herzuleiten.

Semantische Tableaus

Betrachte geschlossene Formeln in $FO^{\neq}(S)$, also Formeln ohne $=$.

Definition 6.1

Formeln aus $FO^{\neq}(S)$ lassen sich in Klassen einteilen:

- (Negierte) atomare Formeln: $p(t_1, \ldots, t_n)$, $\neg p(t_1, \ldots, t_n)$.
- α-Formeln: $A \wedge B$, $\neg(A \vee B)$, $\neg(A \to B)$, $\neg\neg A$.
- β-Formeln: $\neg(A \wedge B)$, $(A \vee B)$, $(A \to B)$.
- γ-**Formeln**: $\forall x A$, $\neg\exists x A$.
- δ-**Formeln**: $\exists x A$, $\neg\forall x A$.

Semantische Tableaus (Fort.)

Tableau-Konstruktion:

α, β-**Formeln:** Wie gehabt.

γ-**Formeln:**

γ	$\forall x\ A$	$\neg\exists x\ A$
$\gamma[t]$	$A\{x/t\}$	$\neg A\{x/t\},$

wobei t ein Grundterm ist, also keine Variablen enthält.

δ-**Formeln:**

δ	$\exists x\ A$	$\neg\forall x\ A$
$\delta[c]$	$A\{x/c\}$	$\neg A\{x/c\},$

wobei c eine Funktionskonstante und **frisch für den Ast** ist.

Semantische Tableaus (Fort.)

Bemerkung zur Konstruktion:

δ-Formeln Müssen **nur einmal** "erfüllt" werden.

Lösungen von δ-Formeln dürfen nicht eingeschränkt werden: ein x mit Eigenschaft A muss nicht als y mit Eigenschaft B funktionieren.

γ-Formeln Müssen **für alle Objekte**, die eingeführt werden, gelten. Werden also **immer weiter** betrachtet.

Intuition: Systematische Konstruktion eines **Herbrand-Modells**:

- δ-Formeln werden mit Skolemisierung behandelt.
  Führe soviele Konstanten ein, wie notwendig ist.
- Wähle als Datenbereich die Terme über den Konstanten.
- Falls die Signatur keine Funktionssymbole enthält,
  sind das gerade die Konstanten.

Semantische Tableaus (Fort.)

Die Beweise von **Korrektheit** und **Vollständigkeit** sind analog zum Fall der Ausssagenlogik.

Lemma 6.2

Sei $A \in FO^{\neq}(S)$ geschlossen und τ ein Tableau für A. Dann gilt

$$A \text{ ist erfüllbar} \quad \text{gdw.} \quad \exists \text{ Ast } \Gamma \in \tau : \Gamma \text{ ist erfüllbar.}$$

Semantische Tableaus (Fort.)

Definition 6.3

Eine Menge geschlossener Formeln $\Gamma \subseteq FO(S)$ heißt **vollständig**, falls

1. für jede α-Formel in Γ gilt $\alpha_1, \alpha_2 \in \Gamma$
2. für jede β-Formel in Γ gilt $\beta_1 \in \Gamma$ oder $\beta_2 \in \Gamma$
3. für jede γ-Formel in Γ gilt $\gamma[t] \in \Gamma$ für alle $t \in D_{\mathcal{H}(S)}$
4. für jede δ-Formel in Γ gibt es ein $t \in D_{\mathcal{H}(S)}$ mit $\delta[t] \in \Gamma$.

Die Menge heißt **abgeschlossen**, falls es ein $B \in FO(S)$ gibt mit $B, \neg B \in \Gamma$. Sonst heißt Γ **offen**.

Beachte: Eingeführte Konstanten sind in der Signatur S und somit auch in den Termen $D_{\mathcal{H}(S)}$ enthalten.

Lemma 6.4 (Hintikka)

Sei $\Gamma \subseteq FO^{\neq}(S)$ vollständig. Dann ist Γ erfüllbar gdw. Γ offen ist.

Semantische Tableaus (Fort.)

Satz 6.5

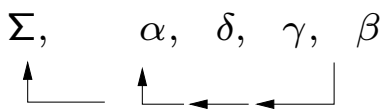
Seien $A \in FO(S)$ und $\Sigma \subseteq FO(S)$.

a) $\models A$ *gdw. es gibt ein abgeschlossenes Tableau für* $\neg A$.

b) $\Sigma \models A$ *gdw. es gibt ein abgeschlossenes Tableau für* $\Sigma \cup \{\neg A\}$.

Eine **systematische Tableau-Konstruktion** garantiert, dass alle Äste vollständig sind (ggf. unendlich).

Idee einer systematischen Tableau-Konstruktion:



Mit der Annahme einer systematischen Tableau-Konstruktion erhält man ein **Semi-Entscheidungsverfahren** für Allgemeingültigkeit.

Satz 6.6

Falls $A \in FO(S)$ allgemeingültig ist, wird von der systematischen Tableau-Konstruktion ein geschlossenes Tableau für $\neg A$ *erstellt.*

Semantische Tableaus (Fort.)

Tableaus sind **kein Entscheidungsverfahren** für Allgemeingültigkeit.

Siehe **Unentscheidbarkeit** in Satz 4.37.

Da das Verfahren korrekt und vollständig ist, **terminiert es ggf. nicht**.

$D_{\mathcal{H}(S)}$ kann unendlich sein: Funktionssymbole.

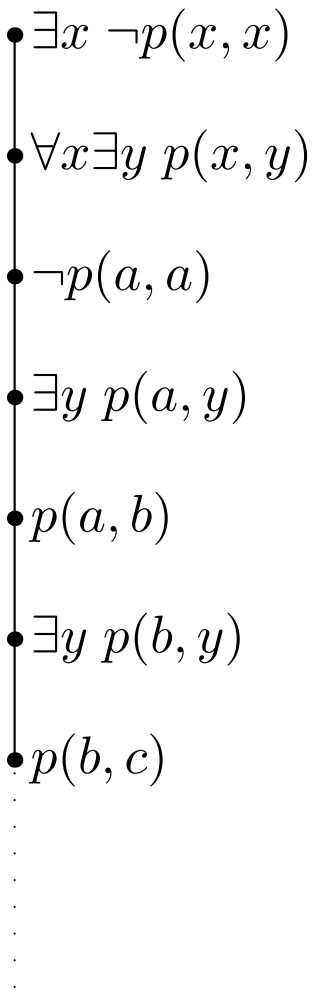
Heuristik zur Konstruktion endlicher Modelle:

Weiche Forderung **frisch** bei δ-Formeln auf.

- Erst existierende Konstanten verwenden.
- Falls diese Wahl zu Widersprüchen führt, führe neue Konstanten ein.
- Sonst Modell gefunden.

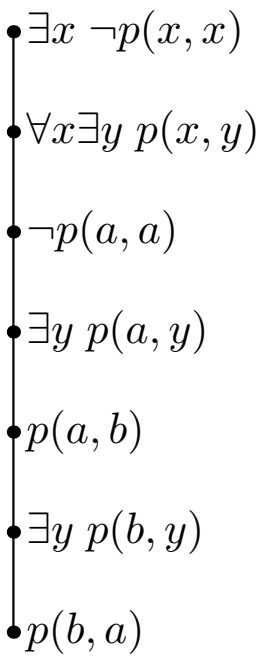
Beispiele zur Wiederverwendung von Konstanten

Gibt es Modelle für $\{\exists x\ \neg p(x,x), \forall x \exists y\ p(x,y)\}$?



Beispiele zur Wiederverwendung von Konstanten (Fort.)

Verwende Konstante a wieder:

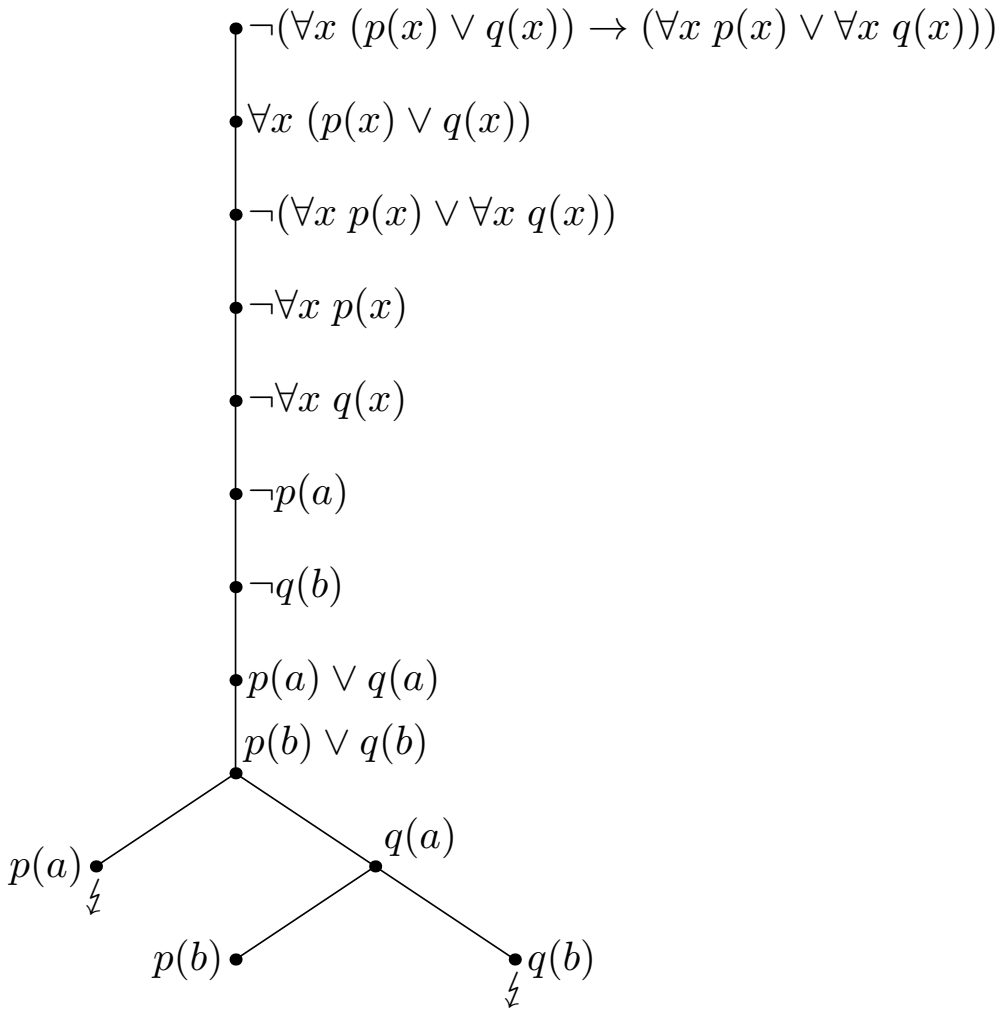


Es gibt also eine Struktur mit zwei Elementen $\{a, b\}$, die Modell ist.
Interpretation des Prädikats:

$p(x,y)$	a	b
a	0	1
b	1	$*$

Beispiele zur Wiederverwendung von Konstanten (Fort.)

Gilt $\models \forall x(p(x) \vee q(x)) \rightarrow (\forall x p(x) \vee \forall x q(x))$?



$\mathcal{M} = (\{a, b\}, \quad I(p)(a) = I(q)(b) = 0, \quad I(p)(b) = I(q)(a) = 1)$

Idee prädikatenlogischer Resolution

Ziel: **Praktisches** Semi-Entscheidungsverfahren für Unerfüllbarkeit basierend auf dem Algorithmus von Gilmore:

> Um Unerfüllbarkeit von $A \equiv \forall x_1 \ldots \forall x_n.B \in FO(S)$ in Skolemform zu zeigen, zeige Unerfüllbarkeit der Herbrand-Expansion $E(A)$.

Beispiel: Sei $A \equiv \forall x.p(x) \wedge \neg p(f(x))$ über $S = (\{a/_0, f/_1\}, \{p/_1\})$. Dann

$$E(A) = \{p(a) \wedge \neg p(f(a)), p(f(a)) \wedge \neg p(f(f(a))), \ldots\}.$$

Kernbeobachtung: Da $A \equiv \forall x_1 \ldots \forall x_n.B$ mit B in **KNF**, lässt sich Unerfüllbarkeit von $E(A)$ mittels **aussagenlogischer Resolution** prüfen:

$$\{p(a)\} \qquad \{\neg p(f(a))\} \qquad \{p(f(a))\} \qquad \{\neg p(f(f(a)))\}$$

$$\searrow \qquad \swarrow$$

$$\square$$

Idee prädikatenlogischer Resolution (Fort.)

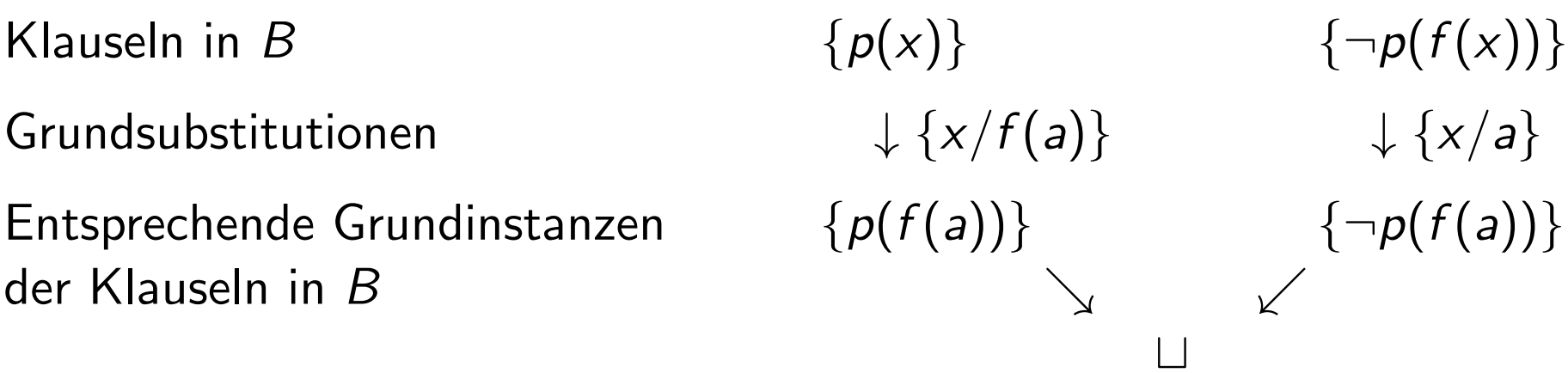
Beobachtung: Bereits die Substitutionen $\{x/a\}$ und $\{x/f(a)\}$ liefern unerfüllbare Formelmenge.

Es werden aber schon zwei Klauseln generiert, die für die Herleitung der leeren Klausel $\square$ **nicht benötigt werden**.

Idee: Generiere geeignete Substitution für jede Klausel in B — **individuell**.

Wende die Substitution nur auf diese Klausel an, nicht auf ganz B.

Am Beispiel:



Idee prädikatenlogischer Resolution (Fort.)

Problem: Algorithmische Suche nach Grundinstanzen zur Herleitung der leeren Klausel $\square$.

- Systematisches Probieren der Grundsubstitutionen — aufwendig.
- Vorausschauende Entscheidung für Grundsubstitutionen, damit später benötigte Resolutionen möglich — schwierig.

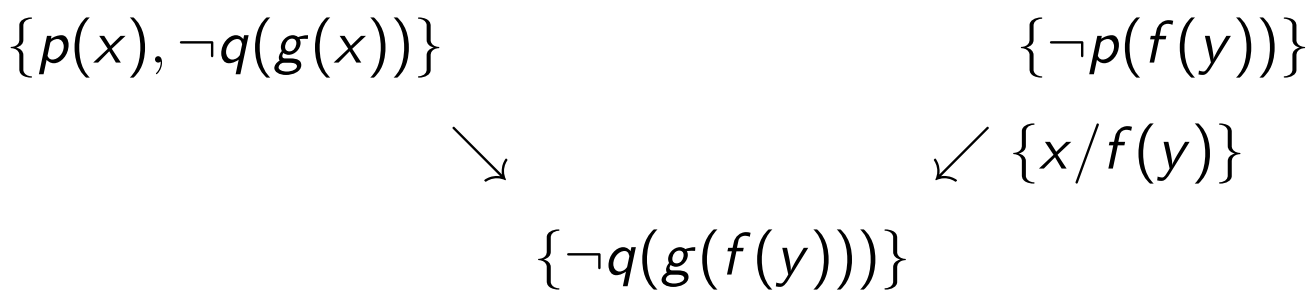
Ansatz: Führe Substitutionen zurückhaltend aus — nur sofern sie für den nächsten Resolutionsschritt notwendig sind.

Am Beispiel:

$$\{p(x), \neg q(g(x))\} \qquad \{\neg p(f(y))\}$$
$$\searrow \qquad \swarrow \{x/f(y)\}$$
$$\{\neg q(g(f(y)))\}$$

Idee prädikatenlogischer Resolution (Fort.)

Am Beispiel:



Was passiert?

Erzeuge prädikatenlogische Resolvente aus prädikatenlogischen Klauseln.

Resolutionsschritt kommt mit Substitution, die Literale in Ausgangsklauseln komplementär macht.

Führe Substitutionen zurückhaltend aus, kein Anlass für Grundsubstitutionen.

Unifikation

Ziel: Berechne Unifikator — eine Substitution, die eine Menge von Literalen identisch macht.

Am Beispiel: Für $\{p(x), p(f(y))\}$ sind

$$\Theta_1 = \{x/f(y)\} \quad \text{und} \quad \Theta_2 = \{x/f(a), y/a\}$$

Unifikatoren. Aber Θ_2 substituiert mehr als notwendig.

Definition 6.7 (Unifikator)

Eine Substitution $\Theta : \{x_1, \ldots, x_n\} \to \{t_1, \ldots, t_n\}$ ist **Unifikator einer Menge von Literalen** $\{L_1, \ldots, L_n\}$, falls

$$L_1\Theta \equiv \ldots \equiv L_n\Theta.$$

Existiert Θ, heißt die Literalmenge **unifizierbar**.

Unifikation (Fort.)

Definition 6.7 (Unifikator (Fort.))

Ein Unifikator Θ von $\{L_1, \ldots, L_n\}$ heißt **allgemeinster Unifikator**, falls für jeden Unifikator Θ' von $\{L_1, \ldots, L_n\}$ eine Substitution $\tilde{\Theta}$ existiert, so dass

$$\Theta' = \Theta\tilde{\Theta}.$$

Anschaulich gilt mit einem allgemeinsten Unifikator:

$$\begin{array}{ccc} A & \xrightarrow{\Theta} & A\Theta \\ & \Theta' \searrow & \downarrow \tilde{\Theta} \\ & & A\Theta' \equiv A\Theta\tilde{\Theta} \end{array}$$

für jede Formel $A \in FO(S)$.

Satz 6.8 (Unifikation, Robinson)

Jede unifizierbare Menge von Literalen besitzt einen allgemeinsten Unifikator.

Unifikationsalgorithmus

Input: $\{L_1, \ldots, L_n\}$.

$\Theta := \{\}$
while $\exists i, j : L_i\Theta \not\equiv L_j\Theta$ **do**
  Durchsuche Literale $L_1\Theta, \ldots, L_n\Theta$ von links nach rechts,
  bis erste Position gefunden, an der $L_k\Theta \not\equiv L_m\Theta$.
  if keines der Zeichen Variable **then**
    return nicht unifizierbar
  end if
  let $x =$ die Variable
  let $t =$ der Term in dem anderen Literal
  if $x \in V(t)$ **then** //Occur-Check
    return nicht unifizierbar
  end if
  $\Theta := \Theta\{x/t\}$
end while
return Θ

Bei positiver Terminierung ist Θ ein allgemeinster Unifikator.

Resolution

Definition 6.9 (Resolvente)

Seien K_1, K_2 prädikatenlogische Klauseln mit disjunkten Variablen. Falls es Literale $L_1, \ldots, L_m \in K_1$ und $L'_1, \ldots, L'_n \in K_2$ gibt, so dass

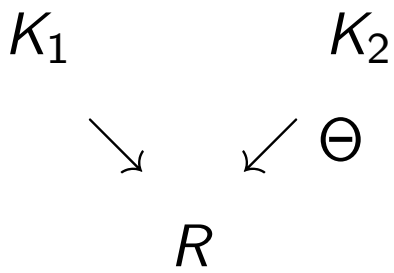
$$\{\overline{L_1}, \ldots, \overline{L_m}, L'_1, \ldots, L'_n\}$$

unifizierbar ist mit allgemeinstem Unifikator Θ, dann heißt

$$R := \big((K_1 \setminus \{L_1, \ldots, L_m\}) \cup (K_2 \setminus \{L'_1, \ldots, L'_n\})\big)\Theta$$

prädikatenlogische Resolvente von K_1 und K_2.

Notation: $K_1, K_2 \underset{Res}{\vdash} R$ oder



Bemerkung: Aussagenlogische Resolution ist ein Spezialfall mit $m = n = 1$ und $\Theta = \{\}$.

Resolution (Fort.)

Am Beispiel:

$\{p(f(x)), \neg q(z), p(z)\}$ $\qquad$ $\{\neg p(y), r(g(y), a)\}$

$\searrow$ $\qquad$ $\swarrow \Theta = \{z/f(x), y/f(x)\}$

$\{\neg q(f(x)), r(g(f(x)), a)\}$

Satz 6.10 (Korrektheit und Widerlegungsvollständigkeit, Robinson)

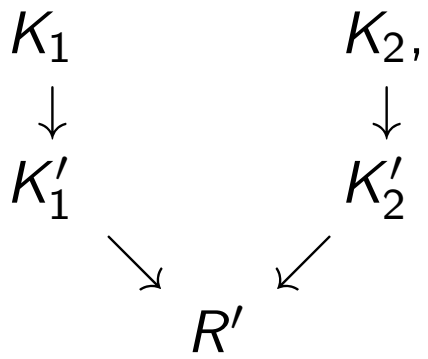
Sei $A \equiv \forall x_1 \ldots \forall x_n . B \in FO(S)$ geschlossen und in Skolemform mit B in KNF. Dann ist A unerfüllbar gdw. $B \underset{Res}{\vdash} \square$.

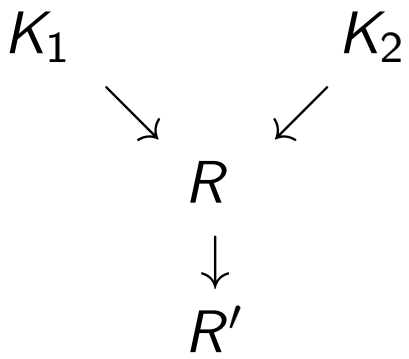
Beachte: Das Verfahren muss nicht terminieren.
Unerfüllbarkeit ist unentscheidbar.

Zum Beweis der Widerlegungsvollständigkeit

Beweisansatz: Reduziere prädikatenlogische Resolution auf (aussagenlogische) Grundresolution (oben vorgestellt).

Technik: Aussagenlogische Resolutionen von Grundinstanzen können geliftet werden in prädikatenlogische Resolutionen:

Falls 

dann 

Lemma 6.11 (Lifting-Lemma)

Seien K_1, K_2 prädikatenlogische Klauseln und K_1', K_2' Grundinstanzen mit aussagenlogischer Resolvente R'.
Dann gibt es eine prädikatenlogische Resolvente R aus K_1, K_2, so dass R' Grundinstanz von R ist.