Logik für Informatiker

3. Prädikatenlogik

Teil 6

23.06.2016

Viorica Sofronie-Stokkermans

Universität Koblenz-Landau

e-mail: sofronie@uni-koblenz.de

Prädikatenlogik

Syntax

1. **Logische Symbole:**
   1.1: Wie in der Aussagenlogik: $\top$, $\bot$; $\neg$; $\vee$, $\wedge$, $\rightarrow$, $\leftrightarrow$
   1.2: Quantoren: $\forall$, $\exists$.

2. **Nichtlogische Symbole:** Signatur $\Sigma = (\Omega, \Pi)$,
   2.1: Ω Menge von Funktionssymbolen. Notation: f/n: f hat Stelligkeit $n \geq 0$,
   2.2: Π Menge von Prädikatensymbolen. Notation: p/m: p hat Stelligkeit $m \geq 0$.
   (Das Gleichheitsprädikat $\approx$ kann (muss aber nicht) enthalten sein.)

   Funktionsssymbole mit Stelligkeit $n = 0$ heißen Konstante

   Prädikatensymbole mit Stelligkeit $n = 0$ heißen Aussagenvariablen

3. **Variablen:** X vorgegebene Menge von abzählbar unendlich vielen Symbolen ist, die wir für (die Bezeichnung von) Variablen verwenden.

- Terme
- Formeln
- Substitutionen

Prädikatenlogik

Semantik:

- Σ-Strukturen
- Valuationen
- Wert eines Terms in $\mathcal{A}$ bzgl. β
- Wahrheitswert einer Formel in $\mathcal{A}$ bzgl. β
- Modelle, Gültigkeit, Erfüllbarkeit
- Folgerung und Äquivalenz
- Wichtige Äquivalenzen

Strukturelle Induction

- für Terme
- für Formeln in Prädikatenlogik

Strukturelle Induktion: Terme

Menge $T_\Sigma(X)$ der Σ-Terme:

Die kleinste Menge mit:

- $X \subseteq T_\Sigma(X)$

Wenn

- $f \in \Omega$,
- n ist die Stelligkeit von f
- $t_1, \ldots, t_n \in T_\Sigma(X)$

dann $f(t_1, \ldots, t_n) \in T_\Sigma(X)$

Strukturelle Induktion: Terme

$\Sigma = (\Omega, \Pi)$

Sei $p(t)$ eine Eigenschaft der Σ-Terme in Prädikatenlogik

Behauptung: Für alle Terme t, $p(t)$ gilt

Beweis durch strukturelle Induktion:

Induktionsbasis: Zu zeigen:
$p(t)$ gilt für $t \in X$ und für alle Konstanten.

Sei t ein Term, (nicht Variable oder Konstante).
Induktionsvoraussetzung:
$p(s)$ gilt für alle Teilterme s von t mit $s \neq t$

Induktionsschritt: Zu zeigen: $p(t)$ gilt:

Fallunterschied über alle $f \in \Omega$ mit $t = f(t_1, \ldots, t_n)$.

Strukturelle Induktion: Formeln

Menge For_Σ **der Formeln über** Σ**:**

Die kleinste Menge, die

- Alle atomaren Formeln enthält,
- $\top \in \mathsf{For}_\Sigma$, $\bot \in \mathsf{For}_\Sigma$,
- Wenn $F, G \in \mathsf{For}_\Sigma$, dann auch
  $\neg F, F \wedge G, F \vee G, F \rightarrow G, F \leftrightarrow G \in \mathsf{For}_\Sigma$,
- Wenn $F \in \mathsf{For}_\Sigma$ und $x \in X$, dann
  $\forall x F \in \mathsf{For}_\Sigma, \exists x F \in \mathsf{For}_\Sigma$

Strukturelle Induktion: Formeln

Sei $p(F)$ eine Eigenschaft der Σ-Formeln in Prädikatenlogik

Behauptung: Für alle Formeln F, $p(F)$ gilt

Beweis durch strukturelle Induktion:

Induktionsbasis: Zu zeigen:
$p(F)$ gilt für $F \in \{\top, \bot\}$ und für alle atomaren Formeln.

Sei F eine Formel (nicht atomar oder $\top$ oder $\bot$).
Induktionsvoraussetzung:
$p(G)$ gilt für alle Teilformeln G von F mit $G \neq F$

Induktionsschritt: Zu zeigen: $p(F)$ gilt:

Fall 1	$F = \neg G$	Fall 2	$F = G_1 \vee G_2$
Fall 3	$F = G_1 \wedge G_2$	Fall 4	$F = G_1 \rightarrow G_2$
Fall 5	$F = G_1 \leftrightarrow G_2$	Fall 6	$F = \forall x G$
		Fall 7	$F = \exists x G$

Substitutionen und Valuationen

Theorem (Substitutionslemma)
Für alle Σ-Strukturen $\mathcal{A}$, Wertebelegungen β, Σ-Formeln F, Variablen x und Terme t gilt:

$$\mathcal{A}, \beta \models F[t/x] \text{ g.d.w. } \mathcal{A}, \beta[x \mapsto \mathcal{A}(\beta)(t)] \models F$$

Substitutionen und Valuationen

Theorem (Substitutionslemma)
Für alle Σ-Strukturen $\mathcal{A}$, Wertebelegungen β, Σ-Formeln F, Variablen x und Terme t gilt:

$$\mathcal{A}, \beta \models F[t/x] \;\text{ g.d.w. }\; \mathcal{A}, \beta[x \mapsto \mathcal{A}(\beta)(t)] \models F$$

Beweis: Strukturelle Induktion

Substitutionen und Valuationen

Theorem (Substitutionslemma)
Für alle Σ-Strukturen $\mathcal{A}$, Wertebelegungen β, Σ-Formeln F, Variablen x und Terme t gilt:

$$\mathcal{A}, \beta \models F[t/x] \text{ g.d.w. } \mathcal{A}, \beta[x \mapsto \mathcal{A}(\beta)(t)] \models F$$

Beweis: Strukturelle Induktion

Plan: Wir benutzen folgendes Lemma:

Lemma: Für alle Σ-Strukturen $\mathcal{A}$, Wertebelegungen β, Variable x und Terme t_i, t:
$\mathcal{A}(\beta)(t_i[t/x]) = \mathcal{A}(\beta[x \mapsto \mathcal{A}(\beta)(t)])(t_i)$

Substitutionen und Valuationen

Lemma: Für alle Σ-Strukturen $\mathcal{A}$, Wertebelegungen β, Variable x und Terme t_i, t:
$\mathcal{A}(\beta)(t_i[t/x]) = \mathcal{A}(\beta[x \mapsto \mathcal{A}(\beta)(t)])(t_i)$

Beweis Strukturelle Induktion

$$p(t_i) \qquad \mathcal{A}(\beta)(t_i[t/x]) = \mathcal{A}(\beta[x \mapsto \mathcal{A}(\beta)(t)])(t_i)$$

Zu zeigen: Für alle Terme t_i, $p(t_i)$ gilt.

1. **Induktionsbasis:** $p(t_i)$ gilt für $t_i \in X$ und für $t_i = c$ Konstante.

Fall 1: $t_i \in X$.

- Fall 1a: $t_i = x$. Dann: $t_i[t/x] = t$.
  $\mathcal{A}(\beta)(t_i[t/x]) = \mathcal{A}(\beta)(t) = \mathcal{A}(\beta[x \mapsto \mathcal{A}(\beta)(t)])(x) = \mathcal{A}(\beta[x \mapsto \mathcal{A}(\beta)(t)])(t_i)$.
- Fall 1a: $t_i = y$, mit $y \neq x$. Dann: $t_i[t/x] = y$
  $\mathcal{A}(\beta)(t_i[t/x]) = \mathcal{A}(\beta)(y) = \mathcal{A}(\beta[x \mapsto \mathcal{A}(\beta)(t)])(y) = \mathcal{A}(\beta[x \mapsto \mathcal{A}(\beta)(t)])(t_i)$.

Fall 2: $t_i = c$, $c/0 \in \Omega$ (Konstante). Dann: $t_i[t/x] = c$
$\mathcal{A}(\beta)(t_i[t/x]) = \mathcal{A}(\beta)(c) = c_{\mathcal{A}} = \mathcal{A}(\beta[x \mapsto \mathcal{A}(\beta)(t)])(c) = \mathcal{A}(\beta[x \mapsto \mathcal{A}(\beta)(t)])(t_i)$.

Substitutionen und Valuationen

Lemma: Für alle Σ-Strukturen $\mathcal{A}$, Wertebelegungen β, Variable x und Terme t_i, t:
$\mathcal{A}(\beta)(t_i[t/x]) = \mathcal{A}(\beta[x \mapsto \mathcal{A}(\beta)(t)])(t_i)$

Beweis (ctd.)

2. **Induktionsvoraussetzung:** Sei t_i Term (nicht Variable oder Konstante).

   Annahme: $p(s)$ gilt für alle Teilterme s von t_i (mit $s \neq t_i$)

3. **Induktionsschritt:** Zu zeigen: $p(t_i)$ gilt, wobei $t_i = f(s_1, \dots, s_n)$.

$$
\begin{array}{rcll}
\mathcal{A}(\beta)(t_i[t/x]) & = & \mathcal{A}(\beta)(f(s_1, \dots, s_n)[t/x]) = & \\
& = & \mathcal{A}(\beta)(f(s_1[t/x], \dots, s_n[t/x]) = & \text{(Anw. Subst.)} \\
& = & f_{\mathcal{A}}(\mathcal{A}(\beta)(s_1[t/x]), \dots, \mathcal{A}(\beta)(s_n[t/x])) & \\
& \overset{IV}{=} & f_{\mathcal{A}}(\mathcal{A}(\beta[x \mapsto \mathcal{A}(\beta)(t)])(s_1), \dots \mathcal{A}(\beta[x \mapsto \mathcal{A}(\beta)(t)])(s_n)) & \\
& = & \mathcal{A}(\beta[x \mapsto \mathcal{A}(\beta)(t)])(f(s_1, \dots, s_n)) & \\
& = & \mathcal{A}(\beta[x \mapsto \mathcal{A}(\beta)(t)])(t_i) &
\end{array}
$$

Substitutionen und Valuationen

Theorem (Substitutionslemma)
Für alle Σ-Strukturen $\mathcal{A}$, Wertebelegungen β, Σ-Formeln F, Variablen x und Terme t gilt:

$$\mathcal{A}, \beta \models F[t/x] \ \text{ g.d.w. } \mathcal{A}, \beta[x \to \mathcal{A}(\beta)(t)] \models F$$

Substitutionen und Valuationen

Theorem (Substitutionslemma)
Für alle Σ-Strukturen $\mathcal{A}$, Wertebelegungen β, Σ-Formeln F, Variablen x und Terme t gilt:

$$\mathcal{A}, \beta \models F[t/x] \text{ g.d.w. } \mathcal{A}, \beta[x \to \mathcal{A}(\beta)(t)] \models F$$

Beweis: Strukturelle Induktion

1. **Induktionsbasis:** Zu zeigen: Die Eigenschaft gilt für $\top$, $\perp$ und alle atomaren Formeln.

Fall 1: $F = \top$ Dann $F[t/x] = \top$;

$$\mathcal{A}(\beta)(F[t/x]) = \mathcal{A}(\beta)(\top) = 1 = \mathcal{A}(\beta[x \mapsto \mathcal{A}(\beta)(t)])(\top) = \mathcal{A}(\beta[x \mapsto \mathcal{A}(\beta)(t)])(F)$$

Fall 2: $F = \perp$ Dann $F[t/x] = \perp$;

$$\mathcal{A}(\beta)(F[t/x]) = \mathcal{A}(\beta)(\perp) = 0 = \mathcal{A}(\beta[x \mapsto \mathcal{A}(\beta)(t)])(\perp) = \mathcal{A}(\beta[x \mapsto \mathcal{A}(\beta)(t)])(F).$$

Fall 3: $F = p(t_1, \dots, t_n)$ mit $p/n \in \Pi$. Dann $F[t/x] = p(t_1[t/x], \dots, t_n[t/x])$.

$\mathcal{A}(\beta)(F[t/x]) = \mathcal{A}(\beta)(p(t_1[t/x], .., t_n[t/x])) = 1$ iff $(\mathcal{A}(\beta)(t_1[t/x]), .., \mathcal{A}(\beta)(t_n[t/x])) \in p_\mathcal{A}$

$\mathcal{A}(\beta[x \mapsto \mathcal{A}(\beta)(t)])(p(t_1, .., t_n)) = 1$ iff $(\mathcal{A}(\beta[x \mapsto \mathcal{A}(\beta)(t)])(t_1), .., \mathcal{A}(\beta[x \mapsto \mathcal{A}(\beta)(t)])(t_n)) \in p_\mathcal{A}$

Lemma: $\mathcal{A}(\beta)(t_i[t/x]) = \mathcal{A}(\beta[x \mapsto \mathcal{A}(\beta)(t)])(t_i)$. Äquivalenz folgt daraus.

Substitutionen und Valuationen

Theorem (Substitutionslemma)
Für alle Σ-Strukturen $\mathcal{A}$, Wertebelegungen β, Σ-Formeln F, Variablen x und Terme t gilt:

$$\mathcal{A}, \beta \models F[t/x] \ \text{ g.d.w. } \mathcal{A}, \beta[x \to \mathcal{A}(\beta)(t)] \models F$$

Beweis: ctd.

2. Induktionsvoraussetzung: Sei F eine Formel (nicht atomar oder $\top$ oder $\bot$).
Annahme: $p(G)$ gilt für alle Teilformeln G von F (mit $G \neq F$)

3. Induktionsschritt: Zu zeigen: $p(F)$ gilt:

Fall 1: $F = \neg G$. Dann $F[t/x] = \neg(G[t/x])$.

$\mathcal{A}(\beta)(F[t/x]) = \mathcal{A}(\beta)(\neg(G[t/x]) = 1$ iff $\mathcal{A}(\beta)(G[t/x]) = 0$ iff (Ind.Voraus.)
$\mathcal{A}(\beta[x \mapsto \mathcal{A}(\beta)(t)])(G)=0$ iff $\mathcal{A}(\beta[x \mapsto \mathcal{A}(\beta)(t)])(\neg G)=1$ iff $\mathcal{A}(\beta[x \mapsto \mathcal{A}(\beta)(t)])(F)=1$

Fall 2-5: $F = G_1 \text{op} G_2$, $\text{op} \in \{\vee, \wedge, \rightarrow, \leftrightarrow\}$. Dann $F[t/x] = G_1[t/x]\text{op}G_2[t/x]$.

$\mathcal{A}(\beta)(F[t/x]) = \mathcal{A}(\beta)(G_1[t/x]\text{op}G_2[t/x]) = \mathcal{A}(\beta)(G_1[t/x])\text{op}_B\mathcal{A}(\beta)(G_2[t/x]) =$ (I.V.)
$\mathcal{A}(\beta[x \mapsto \mathcal{A}(\beta)(t)])(G_1)\text{op}_B\mathcal{A}(\beta[x \mapsto \mathcal{A}(\beta)(t)])(G_2) = \mathcal{A}(\beta[x \mapsto \mathcal{A}(\beta)(t)])(\underbrace{G_1\text{op}G_2}_{F})$.

Substitutionen und Valuationen

Theorem (Substitutionslemma)
Für alle Σ-Strukturen $\mathcal{A}$, Wertebelegungen β, Σ-Formeln F, Variablen x und Terme t gilt:

$$\mathcal{A}, \beta \models F[t/x] \ \text{ g.d.w. } \mathcal{A}, \beta[x \to \mathcal{A}(\beta)(t)] \models F$$

Fall 6: $F = \forall y G$.

Fall 6.1: $y = x$. Dann $F[t/x] = \forall x G[t/x] = \forall x G$.

$\mathcal{A}(\beta)(F[t/x]) = \mathcal{A}(\beta)(\forall x G) = \min\{\mathcal{A}(\beta[x \mapsto a])(G) \mid a \in U_\mathcal{A}\}$

$\mathcal{A}(\beta[x \mapsto \mathcal{A}(\beta)(t)])(\forall x G) = \min\{\mathcal{A}(\beta[x \mapsto \mathcal{A}(\beta)(t)][x \mapsto a])(G) \mid a \in U_\mathcal{A}\} =$
$\min\{\mathcal{A}(\beta[x \mapsto a])(G) \mid a \in U_\mathcal{A}\}$.

Fall 6.2: $y \neq x$, t enthält nicht y. Dann $F[t/x] = \forall y G[t/x] = \forall y(G[t/x])$.

$\mathcal{A}(\beta)(F[t/x]) = \mathcal{A}(\beta)(\forall y(G[t/x]) = \min\{\mathcal{A}(\beta[y \mapsto a])(G[t/x]) | a \in U_\mathcal{A}\} =$ (Ind.Vor.)
$\min\{\mathcal{A}(\beta[y \mapsto a, x \mapsto \mathcal{A}(\beta[y \mapsto a])(t))(G) | a \in U_\mathcal{A}\} =$
$\min\{\mathcal{A}(\beta[y \mapsto a, x \mapsto \mathcal{A}(\beta)(t)])(G) | a \in U_\mathcal{A}\}$ (y kommt in t nicht vor)

$\mathcal{A}(\beta[x \mapsto \mathcal{A}(\beta)(t)])(\forall y G) = = \min\{\mathcal{A}(\beta[x \mapsto \mathcal{A}(\beta)(t)][y \mapsto a])(G) | a \in U_\mathcal{A}\} =$
$\min\{\mathcal{A}(\beta[y \mapsto a, x \mapsto \mathcal{A}(\beta)(t)])(G) = | a \in U_\mathcal{A}\}$.

Substitutionen und Valuationen

Theorem (Substitutionslemma)
Für alle Σ-Strukturen $\mathcal{A}$, Wertebelegungen β, Σ-Formeln F, Variablen x und Terme t gilt:

$$\mathcal{A}, \beta \models F[t/x] \;\text{ g.d.w. }\; \mathcal{A}, \beta[x \to \mathcal{A}(\beta)(t)] \models F$$

Fall 6: $F = \forall y G$.

Fall 6.3: $y \neq x$, t enthält y. Dann $F[t/x] = \forall y' G[y'/y][t/x]$.

$$\begin{aligned}
\mathcal{A}(\beta)(F[t/x]) &= \mathcal{A}(\beta)(\forall y'(G[y'/y][t/x]) = \\
&= \min\{\mathcal{A}(\beta[y' \mapsto a])(G[y'/y][t/x]) \mid a \in U_\mathcal{A}\} = && \text{(Ind.Vor.)} \\
&= \min\{\mathcal{A}(\beta[y' \mapsto a, x \mapsto \mathcal{A}(\beta[y' \mapsto a])(t))(G[y'/y]) \mid a \in U_\mathcal{A}\} = && \text{(Ind.Vor.)} \\
&= \min\{\mathcal{A}(\beta[y' \mapsto a, x \mapsto \mathcal{A}(\beta[y' \mapsto a])(t), y \mapsto a])(G) \mid a \in U_\mathcal{A}\} = \\
&\qquad [\text{Rem:} \mathcal{A}(\beta[y' \mapsto a, x \mapsto \mathcal{A}(\beta[y' \mapsto a])(t)])(y') = a] \\
&= \min\{\mathcal{A}(\beta[x \mapsto \mathcal{A}(\beta)(t), y \mapsto a])(G) \mid a \in U_\mathcal{A}\} && (y' \text{ kommt in } t \text{ u. } G \text{ nicht vor}
\end{aligned}$$

$$\begin{aligned}
\mathcal{A}(\beta[x \mapsto \mathcal{A}(\beta)(t)])(\forall y G) &= \min\{\mathcal{A}(\beta[x \mapsto \mathcal{A}(\beta)(t)][y \mapsto a])(G) \mid a \in U_\mathcal{A}\} = \\
&= \min\{\mathcal{A}(\beta[y \mapsto a, x \mapsto \mathcal{A}(\beta)(t)])(G) \mid a \in U_\mathcal{A}\}.
\end{aligned}$$

Substitutionen und Valuationen

Theorem (Substitutionslemma)
Für alle Σ-Strukturen $\mathcal{A}$, Wertebelegungen β, Σ-Formeln F, Variablen x und Terme t gilt:

$$\mathcal{A}, \beta \models F[t/x] \ \text{ g.d.w. } \mathcal{A}, \beta[x \to \mathcal{A}(\beta)(t)] \models F$$

Fall 7: $F = \exists y G$.

Ähnlich zu Fall 6.

Substitutionen und Valuationen

Theorem (Substitutionslemma)
Für alle Σ-Strukturen $\mathcal{A}$, Wertebelegungen β, Σ-Formeln F, Variablen x und Terme t gilt:

$$\mathcal{A}, \beta \models F[t/x] \ \text{ g.d.w. } \mathcal{A}, \beta[x \mapsto \mathcal{A}(\beta)(t)] \models F$$

Allgemeiner gilt für beliebige Substitutionen σ:

Theorem.

$$\mathcal{A}, \beta \models F\sigma \ \text{ g.d.w. } \mathcal{A}, \beta \circ \sigma \models F,$$

wobei $\beta \circ \sigma : X \to \mathcal{A}$ die Wertebelegung mit $\beta \circ \sigma(x) = \mathcal{A}(\beta)(x\sigma)$, für alle Variablen x.

Substitutionen und Valuationen

Theorem (Substitutionslemma)
Für alle Σ-Strukturen $\mathcal{A}$, Wertebelegungen β, Σ-Formeln F, Variablen x und Terme t gilt:

$$\mathcal{A}, \beta \models F[t/x] \;\text{ g.d.w. }\; \mathcal{A}, \beta[x \mapsto \mathcal{A}(\beta)(t)] \models F$$

Allgemeiner gilt für beliebige Substitutionen σ:

Theorem.

$$\mathcal{A}, \beta \models F\sigma \;\text{ g.d.w. }\; \mathcal{A}, \beta \circ \sigma \models F,$$

wobei $\beta \circ \sigma : X \to \mathcal{A}$ die Wertebelegung mit $\beta \circ \sigma(x) = \mathcal{A}(\beta)(x\sigma)$, für alle Variablen x.

Beweis: $\sigma = [t_1/x_1, \dots t_n/x_n]$
Induktion nach Anzahl n von Variablen in $dom(\sigma) = \{x_1, \dots, x_n\}$

Umbenennung von Variablen

Lemma

Für alle Σ-Formeln F, Variablen x gilt:

$$\forall x\, F \equiv \forall z\, F[z/x]$$

wobei z eine neue Variable ist.

Zusammenfassung: Syntax und Semantik

- Prädikatenlogische Signatur
- Term, Atom, Formel
- Prädikatenlogisches Modell
- Auswertung von Formeln in Modellen
- Erfüllbarkeit, Gültigkeit; Folgerung, Äquivalenz
- Eigenschaften von Quantoren (Vertauschbarkeit untereinander und mit $\wedge, \vee$)
- Substitutionslemma

Algorithmische Probleme

Gültigkeit(F):

$\models F$? (Σ fest)

Erfüllbarkeit(F):

F erfüllbar? (Σ fest)

Modelltest(F,A):

$A \models F$? (Σ fest)

Ausblick: Kalküle, Entscheidbarkeit

Kalküle

Es gibt korrekte und vollständige Kalküle für Prädikatenlogik
(z.B. Resolution, Tableaux)

Ausblick: Kalküle, Entscheidbarkeit

Kalküle

Es gibt korrekte und vollständige Kalküle für Prädikatenlogik
(z.B. Resolution, Tableaux)

Aber diese Kalküle können die Erfüllbarkeit von Formeln NICHT entscheiden

Ausblick: Kalküle, Entscheidbarkeit

Kalküle

Es gibt korrekte und vollständige Kalküle für Prädikatenlogik
(z.B. Resolution, Tableaux)

Aber diese Kalküle können die Erfüllbarkeit von Formeln NICHT entscheiden

Aussagenlogik

Allgemeingültigkeit, Erfüllbarkeit, Unerfüllbarkeit ENTSCHEIDBAR

Prädikatenlogik

- Es gibt Σ, so dass Gültigkeit(F) unentscheidbar.
- Menge der allgemeingültigen Formeln REKURSIV AUFZÄHLBAR
- Menge der unerfüllbaren Formeln REKURSIV AUFZÄHLBAR
- Menge der erfüllbaren Formeln NICHT REKURSIV AUFZÄHLBAR

Rekursiv Aufzählbar

Informelle Definition. Als rekursiv aufzählbare Menge

(auch semi-entscheidbare Menge, positiv semi-entscheidbare Menge, halb-entscheidbare Menge, berechenbar aufzählbare Menge, kurz r.e., c.e)

wird in der Berechenbarkeitstheorie eine aufzählbare Menge A (z.B. eine Teilmenge von $\mathbb{N}$) bezeichnet, wenn es einen Algorithmus gibt, der die Elemente von A aufzählt.

Äquivalent ist folgende Definition: es gibt einen Algorithmus, der 1 ausgibt wenn die Eingabe in A ist, und auf anderen Eingaben 0 ausgibt oder nicht hält.

Zusammenfassung: Syntax und Semantik

- Prädikatenlogische Signatur
- Term, Atom, Formel
- Prädikatenlogisches Modell
- Auswertung von Formeln in Modellen
- Erfüllbarkeit, Gültigkeit; Folgerung, Äquivalenz
- Eigenschaften von Quantoren (Vertauschbarkeit untereinander und mit $\wedge, \vee$)
- Unentscheidbarkeit der Erfüllbarkeit von Formeln

Unser Ziel

Kalküle zur systematischen Überprüfung von Erfüllbarkeit

(für Formeln und/oder Formelmengen)

Normalformen, Skolemisierung, Herbrandmodelle

Vorteile von Normalformen

- Reduktion der logischen Konzepte
- einfache Datenstrukturen für Beweisverfahren

Negationsnormalform

Definition. Eine Formel $F \in \mathrm{For}_\Sigma$ ist in Negationsnormalform (NNF), falls:

- $\rightarrow, \leftrightarrow$ kommen in F nicht vor
- jedes Negationszeichen in F steht direkt vor einem Atom (insbes. auch kein $\neg\neg$)

Negationsnormalform

Definition. Eine Formel $F \in For_\Sigma$ ist in Negationsnormalform (NNF), falls:

- $\rightarrow, \leftrightarrow$ kommen in F nicht vor
- jedes Negationszeichen in F steht direkt vor einem Atom (insbes. auch kein $\neg\neg$)

Beispiele:

$$P, \quad \neg P, \quad (\neg P \vee Q) \wedge (R \vee (Q \wedge \neg P))$$

$$p(x, y) \vee \neg q(y)$$

$$\neg\neg P, \quad \neg(P \vee Q)$$

$$p(x, y) \rightarrow q(x)$$

Negationsnormalform

Definition. Eine Formel $F \in For_\Sigma$ ist in Negationsnormalform (NNF), falls:

- $\rightarrow, \leftrightarrow$ kommen in F nicht vor
- jedes Negationszeichen in F steht direkt vor einem Atom (insbes. auch kein $\neg\neg$)

Beispiele:

NNF:

$$P, \quad \neg P, \quad (\neg P \vee Q) \wedge (R \vee (Q \wedge \neg P))$$

$$p(x, y) \vee \neg q(y)$$

nicht NNF:

$$\neg\neg P, \quad \neg(P \vee Q)$$

$$p(x, y) \rightarrow q(x)$$

Bereinigte Formeln

Definition. Eine Formel $F \in For_\Sigma$ ist bereinigt, falls:

- Keine Variable in F sowohl gebunden als auch frei vorkommt
- Keine Variable mehr als einmal in F quantifiziert ist

Bereinigte Formeln

Definition. Eine Formel $F \in For_\Sigma$ ist bereinigt, falls:

- Keine Variable in F sowohl gebunden als auch frei vorkommt
- Keine Variable mehr als einmal in F quantifiziert ist

Beispiele:

Bereinigt

$$P \vee Q$$

$$\forall x \exists y (p(x) \vee q(x, y) \vee \exists z \, r(x, z))$$

Bereinigte Formeln

Definition. Eine Formel $F \in For_\Sigma$ ist bereinigt, falls:

- Keine Variable in F sowohl gebunden als auch frei vorkommt
- Keine Variable mehr als einmal in F quantifiziert ist

Beispiele:

Bereinigt

$P \vee Q$

$\forall x \exists y (p(x) \vee q(x, y) \vee \exists z\, r(x, z))$

Nicht bereinigt

$p(x) \vee \forall x\, q(x)$

$(\forall x\, p(x)) \vee (\forall x\, q(x))$

Pränexe Normalform

Definition: Pränexe Formeln sind von der Form

$$Q_1 x_1 \dots Q_n x_n \; F,$$

wobei F quantorenfrei, $Q_i \in \{\forall, \exists\}$. Hierbei heißt $Q_1 x_1 \dots Q_n x_n$ der Quantorenpräfix und F die Matrix der Formel.

Pränexe Normalform

Definition: Pränexe Formeln sind von der Form

$$Q_1 x_1 \ldots Q_n x_n \; F,$$

wobei F quantorenfrei, $Q_i \in \{\forall, \exists\}$. Hierbei heißt $Q_1 x_1 \ldots Q_n x_n$ der Quantorenpräfix und F die Matrix der Formel.

Beispiele

In Pränexnormalform

$p(x) \vee q(x)$

$\forall x \exists y \, (p(x) \vee q(y))$

Nicht in Pränexnormalform

$(\forall x \, p(x)) \vee (\exists y \, q(y))$

Pränexe Normalform

Welche Formeln sind in Pränexnormalform?

$$\forall x P(x) \vee \forall x Q(x)$$

$$\forall x \forall y \neg(P(x) \rightarrow Q(y))$$

$$\forall x \exists y R(x, y)$$

$$R(x, y)$$

$$\neg \forall x R(x, y)$$

Pränexe Normalform

Theorem. Zu jeder Formel $F \in \mathrm{For}_\Sigma$ gibt es eine äquivalente Formel in Pränexnormalform

Konstruktiver Beweis

(1) Formel in NNF transformieren

1. Elimination von $\leftrightarrow$
2. Elimination von $\rightarrow$
3. "Nach innen schieben" von $\neg$
   – aussagenlogische Umformungen (de Morgans Regeln; $\neg\neg A \equiv A$)
   – $\neg\forall x F(x) \equiv \exists x \neg F(x)$
   – $\neg\exists x F(x) \equiv \forall x \neg F(x)$

(2) Formel bereinigen

(3) Alle Quantoren nach vorne (Reihenfolge unverändert lassen)

Beispiel 1

$$F = \forall x \Big(\big(\exists y R(x, y) \land \forall y \neg S(x, y)\big) \rightarrow \neg\big(\exists y R(x, y) \land P\big) \Big)$$

Beispiel 1

$$F = \forall x \Big(\big(\exists y R(x,y) \land \forall y \neg S(x,y)\big) \to \neg\big(\exists y R(x,y) \land P\big) \Big)$$

(1) NNF

1.1. Elimination von $\to$

$$\forall x \Big(\neg\big(\exists y R(x,y) \land \forall y \neg S(x,y)\big) \lor \neg\big(\exists y R(x,y) \land P\big) \Big)$$

Beispiel 1

$$F = \forall x \Big((\exists y R(x,y) \wedge \forall y \neg S(x,y)) \rightarrow \neg(\exists y R(x,y) \wedge P) \Big)$$

(1) NNF

1.1. Elimination von $\rightarrow$

$$\forall x \Big(\neg(\exists y R(x,y) \wedge \forall y \neg S(x,y)) \vee \neg(\exists y R(x,y) \wedge P) \Big)$$

1.2. "Nach innen schieben" von $\neg$

$$\forall x \Big((\forall y \neg R(x,y) \vee \exists y S(x,y)) \vee (\forall y \neg R(x,y) \vee \neg P) \Big)$$

Beispiel 1

$$F = \forall x \Big((\exists y R(x,y) \wedge \forall y \neg S(x,y)) \rightarrow \neg(\exists y R(x,y) \wedge P) \Big)$$

(1) NNF

1.1. Elimination von $\rightarrow$

$$\forall x \Big(\neg(\exists y R(x,y) \wedge \forall y \neg S(x,y)) \vee \neg(\exists y R(x,y) \wedge P) \Big)$$

1.2. "Nach innen schieben" von $\neg$

$$\forall x \Big((\forall y \neg R(x,y) \vee \exists y S(x,y)) \vee (\forall y \neg R(x,y) \vee \neg P) \Big)$$

(2) Formel bereinigen

$$\forall x \Big((\forall y_1 \neg R(x,y_1) \vee \exists y_2 S(x,y_2)) \vee (\forall y_3 \neg R(x,y_3) \vee \neg P) \Big)$$

Beispiel 1

$$F = \forall x \Big((\exists y R(x,y) \wedge \forall y \neg S(x,y)) \rightarrow \neg(\exists y R(x,y) \wedge P) \Big)$$

(1) NNF

1.1. Elimination von $\rightarrow$

$$\forall x \Big(\neg(\exists y R(x,y) \wedge \forall y \neg S(x,y)) \vee \neg(\exists y R(x,y) \wedge P) \Big)$$

1.2. “Nach innen schieben” von $\neg$

$$\forall x \Big((\forall y \neg R(x,y) \vee \exists y S(x,y)) \vee (\forall y \neg R(x,y) \vee \neg P) \Big)$$

(2) Formel bereinigen

$$\forall x \Big((\forall y_1 \neg R(x,y_1) \vee \exists y_2 S(x,y_2)) \vee (\forall y_3 \neg R(x,y_3) \vee \neg P) \Big)$$

(3) Alle Quantoren nach vorne (Reihenfolge unverändert lassen)

$$\forall x \forall y_1 \exists y_2 \forall y_3 \Big((\neg R(x,y_1) \vee S(x,y_2)) \vee (\neg R(x,y_3) \vee \neg P) \Big)$$

Beispiel 2

$$F := (\forall x((p(x) \vee q(x,y)) \wedge \exists z\, r(x,y,z))) \to ((p(z) \wedge q(x,z)) \wedge \forall z\, r(z,x,y))$$

1. NNF

$$\begin{array}{rll}
F \equiv & \neg(\forall x((p(x) \vee q(x,y)) \wedge \exists z\, r(x,y,z))) \vee ((p(z) \wedge q(x,z)) \wedge \forall z\, r(z,x,y)) & \text{Elim } \to \\
\equiv & \exists x \neg((p(x) \vee q(x,y)) \wedge \exists z\, r(x,y,z))) \vee ((p(z) \wedge q(x,z)) \wedge \forall z\, r(z,x,y)) & (\text{De Morgan, } \neg\forall \equiv \exists\neg) \\
\equiv & \exists x(\neg(p(x) \vee q(x,y)) \vee \neg\exists z\, r(x,y,z))) \vee ((p(z) \wedge q(x,z)) \wedge \forall z\, r(z,x,y)) & (\text{De Morgan}) \\
\equiv & \exists x((\neg p(x) \wedge \neg q(x,y)) \vee \forall z\, \neg r(x,y,z)) \vee ((p(z) \wedge q(x,z)) \wedge \forall z\, r(z,x,y)) & (\text{De Morgan, } \neg\exists \equiv \forall\neg) \\
=: & F_1 &
\end{array}$$

Beispiel 2

$$F := (\forall x((p(x) \vee q(x,y)) \wedge \exists z\, r(x,y,z))) \to ((p(z) \wedge q(x,z)) \wedge \forall z\, r(z,x,y))$$

1. NNF

$$\begin{array}{rll}
F \equiv & \neg(\forall x((p(x) \vee q(x,y)) \wedge \exists z\, r(x,y,z))) \vee ((p(z) \wedge q(x,z)) \wedge \forall z\, r(z,x,y)) & \text{Elim } \to \\
\equiv & \exists x \neg((p(x) \vee q(x,y)) \wedge \exists z\, r(x,y,z))) \vee ((p(z) \wedge q(x,z)) \wedge \forall z\, r(z,x,y)) & (\text{De Morgan}, \neg\forall \equiv \exists\neg) \\
\equiv & \exists x(\neg(p(x) \vee q(x,y)) \vee \neg\exists z\, r(x,y,z))) \vee ((p(z) \wedge q(x,z)) \wedge \forall z\, r(z,x,y)) & (\text{De Morgan}) \\
\equiv & \exists x((\neg p(x) \wedge \neg q(x,y)) \vee \forall z\, \neg r(x,y,z)) \vee ((p(z) \wedge q(x,z)) \wedge \forall z\, r(z,x,y)) & (\text{De Morgan}, \neg\exists \equiv \forall\neg) \\
=: & F_1 &
\end{array}$$

2. Bereinigung

$$F_1 \equiv \quad \exists x_1((\neg p(x_1) \wedge \neg q(x_1,y)) \vee \forall z_1\, \neg r(x_1,y,z_1)) \vee ((p(z) \wedge q(x,z)) \wedge \forall z_2\, r(z_2,x,y)) \quad =: \quad F_2$$

Beispiel 2

$$F := (\forall x((p(x) \vee q(x,y)) \wedge \exists z\, r(x,y,z))) \rightarrow ((p(z) \wedge q(x,z)) \wedge \forall z\, r(z,x,y))$$

1. NNF

$$\begin{array}{lll}
F \equiv & \neg(\forall x((p(x) \vee q(x,y)) \wedge \exists z\, r(x,y,z))) \vee ((p(z) \wedge q(x,z)) \wedge \forall z\, r(z,x,y)) & \text{Elim } \rightarrow \\
\equiv & \exists x \neg((p(x) \vee q(x,y)) \wedge \exists z\, r(x,y,z))) \vee ((p(z) \wedge q(x,z)) \wedge \forall z\, r(z,x,y)) & (\text{De Morgan}, \neg\forall \equiv \exists\neg) \\
\equiv & \exists x(\neg(p(x) \vee q(x,y)) \vee \neg\exists z\, r(x,y,z))) \vee ((p(z) \wedge q(x,z)) \wedge \forall z\, r(z,x,y)) & (\text{De Morgan}) \\
\equiv & \exists x((\neg p(x) \wedge \neg q(x,y)) \vee \forall z\, \neg r(x,y,z)) \vee ((p(z) \wedge q(x,z)) \wedge \forall z\, r(z,x,y)) & (\text{De Morgan}, \neg\exists \equiv \forall\neg) \\
=: & F_1 &
\end{array}$$

2. Bereinigung

$$F_1 \equiv \exists x_1((\neg p(x_1) \wedge \neg q(x_1,y)) \vee \forall z_1\, \neg r(x_1,y,z_1)) \vee ((p(z) \wedge q(x,z)) \wedge \forall z_2\, r(z_2,x,y)) \quad =: \quad F_2$$

3. Alle Quantoren nach vorne $\mapsto$ Pränexnormalform

$$F_2 \equiv \underbrace{\exists x_1 \forall z_1 \forall z_2\, ((\neg p(x_1) \wedge \neg q(x_1,y)) \vee \neg r(x_1,y,z_1)) \vee ((p(z) \wedge q(x,z)) \wedge r(z_2,x,y))}_{\text{Pränexnormalform von } F}$$

Skolemnormalform

Definition. Eine Formel $F \in \mathsf{For}_\Sigma$ ist in Skolemnormalform (SNF), falls:

- F ist in Pränexnormalform
- F enthält nur universelle Quantoren

Skolemnormalform

Definition. Eine Formel $F \in \mathrm{For}_\Sigma$ ist in Skolemnormalform (SNF), falls:

- F ist in Pränexnormalform
- F enthält nur universelle Quantoren

Beispiele:

In Skolemnormalform

$$\forall x \forall y (p(x) \vee q(y))$$

Nicht in Skolemnormalform

$$\forall x\, p(x) \vee \forall y\, q(y)$$

$$\forall x \exists y\, (p(x) \vee q(y))$$

Skolemisierung

Skolemisierung: Transformation $\Rightarrow_S$:

$$\forall x_1, \ldots, x_n \exists y F \quad \Rightarrow_S \quad \forall x_1, \ldots, x_n F[f(x_1, \ldots, x_n)/y]$$

wobei f/n ein neues Funktionssymbol (Skolemfunktion).

Skolemisierung

Beispiel

Gegeben:

$\forall w(\exists x(p(w, x) \vee \forall y(q(w, x, y) \wedge \exists z\, r(y, z))))$

Skolemisierung

Beispiel

Gegeben:

$$\forall w(\exists x(p(w,x) \vee \forall y\,(q(w,x,y) \wedge \exists z\, r(y,z))))$$

Pränexnormalform:

$$\forall w \exists x \forall y \exists z((p(w,x) \vee (q(w,x,y) \wedge r(y,z))))$$

Skolemisierung

Beispiel

Gegeben:

$\forall w(\exists x(p(w,x) \vee \forall y\,(q(w,x,y) \wedge \exists z\, r(y,z))))$

Pränexnormalform:

$\forall w \exists x \forall y \exists z((p(w,x) \vee (q(w,x,y) \wedge r(y,z))))$

Skolemisierung: $x \mapsto sk_x(w)$, $z \mapsto sk_z(w,y)$

$\forall w \forall y((p(w, sk_x(w)) \vee (q(w, sk_x(w), y) \wedge r(y, sk_z(w,y)))))$

Skolemisierung

Zusammen:

$$F \overset{*}{\Rightarrow}_P \underbrace{G}_{\text{pränexe Form}} \overset{*}{\Rightarrow}_S \underbrace{H}_{\text{pränex, kein } \exists}$$

Theorem:

Seien F, G und H wie oben angenommen. Dann:

(1) F und G sind äquivalent.

(2) G erfüllbar gdw. H erfüllbar

(bzgl. Σ-Str) (bzgl. Σ'-Str)

wobei $\Sigma' = (\Omega \cup SKF, \Pi)$, wenn $\Sigma = (\Omega, \Pi)$.

Skolemisierung

Lemma. Sei $G = \forall x_1, \ldots, x_n \exists y G_1$ und $H = \forall x_1, \ldots, x_n G_1[f(x_1, \ldots, x_n)/y]$, wobei f ein neues Funktionssymbol (Skolemfunktion) ist.

G erfüllbar (bezgl. Σ-Str.) genau dann, wenn H erfüllbar (bezgl. Σ'-Str.).

wobei: $\Sigma = (\Omega, \Pi)$ und $\Sigma' = (\Omega \cup \{f\}, \Pi)$.

Beweis: (1) G erfüllbar $\Rightarrow$ H erfüllbar

Sei $\mathcal{A}$ eine Σ-Struktur und $\beta : X \to U_{\mathcal{A}}$ mit $\mathcal{A}(\beta)(G) = 1$.

$$\begin{aligned} 1 = \mathcal{A}(\beta)(G) &= \min_{a_1,\ldots,a_n \in U_{\mathcal{A}}} \mathcal{A}(\beta[x_1 \mapsto a_1, \ldots, x_n \mapsto a_n])(\exists y G_1) \\ &= \min_{a_1,\ldots,a_n \in U_{\mathcal{A}}} \max_{b \in U_{\mathcal{A}}} \mathcal{A}(\beta[x_1 \mapsto a_1, \ldots, x_n \mapsto a_n, y \mapsto b])(G_1) \end{aligned}$$

d.h. für alle $a_1, \ldots, a_n \in U_{\mathcal{A}}$ es gibt ein $b \in U_{\mathcal{A}}$ mit:
$\mathcal{A}(\beta[x_1 \mapsto a_1, \ldots, x_n \mapsto a_n, y \mapsto b])(G_1) = 1$.

Wir definieren eine Σ'-struktur $\mathcal{A}'$, in der alle Funktionen und Prädikate in Σ wie in $\mathcal{A}$ definiert sind und in der wir für alle $a_1, \ldots, a_n \in U_{\mathcal{A}}$, $f(a_1, \ldots, a_n) := b$ definieren.

Dann $\mathcal{A}'(\beta[x_1 \mapsto a_1, \ldots, x_n \mapsto a_n])(G_1[f(x_1, \ldots, x_n)/y]) = 1$.

Skolemisierung

Lemma. Sei $G = \forall x_1, \ldots, x_n \exists y G_1$ und $H = \forall x_1, \ldots, x_n G_1[f(x_1, \ldots, x_n)/y]$, wobei f ein neues Funktionssymbol (Skolemfunktion) ist.

G erfüllbar (bezgl. Σ-Str.) genau dann, wenn H erfüllbar (bezgl. Σ'-Str.).

wobei: $\Sigma = (\Omega, \Pi)$ und $\Sigma' = (\Omega \cup \{f\}, \Pi)$.

Beweis: (2) H erfüllbar $\Rightarrow$ G erfüllbar

Sei $\mathcal{A}'$ eine Σ'-Struktur und $\beta : X \to U_{\mathcal{A}'}$ mit $\mathcal{A}'(\beta)(H) = 1$.

$$\begin{aligned} 1 = \mathcal{A}'(\beta)(H) &= \min_{a_1,\ldots,a_n \in U_{\mathcal{A}'}} \mathcal{A}'(\beta[x_1 \mapsto a_1, \ldots, x_n \mapsto a_n])(G_1[f(x_1, \ldots, x_n)/y]) \\ &= \min_{a_1,\ldots,a_n \in U_{\mathcal{A}'}} (\beta[x_1 \mapsto a_1, \ldots, x_n \mapsto a_n, y \mapsto f_{\mathcal{A}'}(a_1, \ldots, a_n)])(G_1) \end{aligned}$$

d.h. für alle $a_1, \ldots, a_n \in U_{\mathcal{A}'}$ es gibt ein $b = f_{\mathcal{A}'}(a_1, \ldots, a_n) \in U_{\mathcal{A}'}$ mit:
$\mathcal{A}(\beta[x_1 \mapsto a_1, \ldots, x_n \mapsto a_n, y \mapsto b])(G_1) = 1$.

Dann gilt: $$\begin{aligned} \mathcal{A}(\beta)(G) &= \min_{a_1,\ldots,a_n \in U_{\mathcal{A}}} \mathcal{A}(\beta[x_1 \mapsto a_1, \ldots, x_n \mapsto a_n])(\exists y G_1) \\ &= \min_{a_1,\ldots,a_n \in U_{\mathcal{A}}} \max_{b \in U_{\mathcal{A}}} \mathcal{A}(\beta[x_1 \mapsto a_1, \ldots, x_n \mapsto a_n, y \mapsto b])(G_1) = 1 \end{aligned}$$

wobei $\mathcal{A}$ ist $\mathcal{A}'$ ohne die Funktion f.

Skolemisierung

Theorem:
Seien F, G und H wie oben angenommen. Dann:

(1) F und G sind äquivalent.

(2) G erfüllbar gdw. H erfüllbar
(bzgl. Σ-Str) (bzgl. Σ'-Str)
wobei $\Sigma' = (\Omega \cup SKF, \Pi)$, wenn $\Sigma = (\Omega, \Pi)$.

Beweisidee: Die Lemma (auf Seiten 57–58) wird sukzessive für alle existentiell quantifizierten Variablen (von links nach rechts) angewandt.

Klauselnormalform (konjunktive Normalform)

Transformationsregeln $\Rightarrow_K$

(1)	$(F \leftrightarrow G)$	$\Rightarrow_K$	$(F \rightarrow G) \land (G \rightarrow F)$	
(2)	$(F \rightarrow G)$	$\Rightarrow_K$	$(\neg F \lor G)$	
(3)	$\neg(F \lor G)$	$\Rightarrow_K$	$(\neg F \land \neg G)$	
(4)	$\neg(F \land G)$	$\Rightarrow_K$	$(\neg F \lor \neg G)$	
(5)	$\neg\neg F$	$\Rightarrow_K$	F	
$(\neg\forall)$	$\neg\forall x F$	$\Rightarrow_K$	$\exists x \neg F$	
$(\neg\exists)$	$\neg\exists x F$	$\Rightarrow_K$	$\forall x \neg F$	(NNF)

(P) Pränex Normalform

(S) Skolemisierung

(6)	$(F \land G) \lor H$	$\Rightarrow_K$	$(F \lor H) \land (G \lor H)$	
(7)	$(F \land \top)$	$\Rightarrow_K$	F	
(8)	$(F \land \bot)$	$\Rightarrow_K$	$\bot$	
(9)	$(F \lor \top)$	$\Rightarrow_K$	$\top$	
(10)	$(F \lor \bot)$	$\Rightarrow_K$	F	(KNF)