

Logik für Informatiker

3. Prädikatenlogik

Teil 5

6.06.2019

Viorica Sofronie-Stokkermans

Universität Koblenz-Landau

e-mail: sofronie@uni-koblenz.de

Organisatorisches

Foodle: Terminfindung Nachklausur

Intervall Di. 1.10 - Fr. 11.10.2019

Bis jetzt

Prädikatenlogik

- Syntax

- Signatur
- Terme, Formeln
- Substitutionen

- Semantik

- Σ -Struktur
- Valuation (über einer Σ -Struktur \mathcal{A})
- Wert eines Terms in \mathcal{A} bzgl. β
- Wahrheitswert einer Formel in \mathcal{A} bzgl. β
- Modelle, Gültigkeit, Erfüllbarkeit

Wichtige Äquivalenzen (Zusammenfassung)

- $\forall x \forall y F \equiv \forall y \forall x F$
- $\exists x \exists y F \equiv \exists y \exists x F$
- $\forall x (F \wedge G) \equiv (\forall x F) \wedge (\forall x G)$
- $\exists x (F \vee G) \equiv (\exists x F) \vee (\exists x G)$
- $\forall x F \equiv \neg \exists x \neg F$ $\neg \forall x F \equiv \exists x \neg F$
- $\exists x F \equiv \neg \forall x \neg F$ $\neg \exists x F \equiv \forall x \neg F$

Falls x in der Formel H nicht vorkommt, dann:

- $(\forall x F(x)) \vee H \equiv \forall x (F(x) \vee H)$
- $(\forall x F(x)) \wedge H \equiv \forall x (F(x) \wedge H)$
- $(\exists x F(x)) \vee H \equiv \exists x (F(x) \vee H)$
- $(\exists x F(x)) \wedge H \equiv \exists x (F(x) \wedge H)$

Zusammenfassung

Aber Vorsicht

$$\forall x \exists y F \not\equiv \exists y \forall x F$$

- $\exists y \forall x F \models \forall x \exists y F$
- $\forall x \exists y F \not\models \exists y \forall x F$

$$\forall x (F \vee G) \not\equiv (\forall x F) \vee (\forall x G)$$

- $(\forall x F) \vee (\forall x G) \models \forall x (F \vee G)$
- $\forall x (F \vee G) \not\models \forall x F \vee \forall x G$

$$\exists x (F \wedge G) \not\equiv (\exists x F) \wedge (\exists x G)$$

- $\exists x (F \wedge G) \models (\exists x F) \wedge (\exists x G)$
- $(\exists x F) \wedge (\exists x G) \not\models \exists x (F \wedge G)$

Andere wichtige Äquivalenzen

$$(F \wedge F) \equiv F \quad (F \vee F) \equiv F \quad (\text{Idempotenz})$$

$$(F \wedge G) \equiv (G \wedge F) \quad (F \vee G) \equiv (G \vee F) \quad (\text{Kommutativität})$$

$$(F \wedge (G \wedge H)) \equiv ((F \wedge G) \wedge H)$$

$$(F \vee (G \vee H)) \equiv ((F \vee G) \vee H) \quad (\text{Assoziativität})$$

$$(F \wedge (F \vee G)) \equiv F$$

$$(F \vee (F \wedge G)) \equiv F \quad (\text{Absorption})$$

$$(F \wedge (G \vee H)) \equiv ((F \wedge G) \vee (F \wedge H))$$

$$(F \vee (G \wedge H)) \equiv ((F \vee G) \wedge (F \vee H)) \quad (\text{Distributivität})$$

$$(\neg\neg F) \equiv F \quad (\text{Doppelte Negation})$$

$$\neg(F \wedge G) \equiv (\neg F \vee \neg G)$$

$$\neg(F \vee G) \equiv (\neg F \wedge \neg G) \quad (\text{De Morgan's Regeln})$$

$$(F \rightarrow G) \equiv (\neg G \rightarrow \neg F) \quad (\text{Kontraposition})$$

$$(F \rightarrow G) \equiv (\neg F \vee G) \quad (\text{Elimination Implikation})$$

$$F \leftrightarrow G \equiv (F \rightarrow G) \wedge (G \rightarrow F) \quad (\text{Elimination Äquivalenz})$$

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

• $t_1, \dots, t_n \in T_\Sigma(X)$

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

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 $t = f(t_1, \dots, t_n)$.

Strukturelle Induktion: Formeln

Menge For_Σ der Formeln über Σ :

Die kleinste Menge, die

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

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 $F \in \{\top, \perp\}$ und für alle atomaren Formeln.

Sei F eine Formel (nicht atomar oder \top oder \perp).

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 $F = \forall xG$

Fall 7 $F = \exists xG$

Substitutionen und Valuationen

Theorem (Substitutionslemma)

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

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

Substitutionen und Valuationen

Theorem (Substitutionslemma)

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

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

Beweis: Strukturelle Induktion

Substitutionen und Valuationen

Theorem (Substitutionslemma)

Für alle Σ -Strukturen \mathcal{A} , Wertbelegungen 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} , Wertbelegungen 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

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

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 $t_i = f(s_1, \dots, s_n)$.

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

Substitutionen und Valuationen

Theorem (Substitutionslemma)

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

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

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

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 $F[t/x] = p(t_1[t/x], \dots, t_n[t/x])$.

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

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

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:

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

Beweis: ctd.

2. Induktionsvoraussetzung: Sei F eine Formel (nicht atomar oder \top oder \perp).

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 $F[t/x] = \neg(G[t/x])$.

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

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

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

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

$$\mathcal{A}(\beta[x \mapsto \mathcal{A}(\beta)(t)])(G_1) \text{ op}_B \mathcal{A}(\beta[x \mapsto \mathcal{A}(\beta)(t)])(G_2) = \mathcal{A}(\beta[x \mapsto \mathcal{A}(\beta)(t)])(\underbrace{G_1 \text{ op } G_2}_F).$$

Substitutionen und Valuationen

Theorem (Substitutionslemma)

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

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

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

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

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

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

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

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

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

$$\min\{\mathcal{A}(\beta[y \mapsto a, x \mapsto \mathcal{A}(\beta)(t)])(G) \mid a \in U_{\mathcal{A}}\} \quad (y \text{ kommt in } t \text{ nicht vor})$$

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

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

Substitutionen und Valuationen

Theorem (Substitutionslemma)

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

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

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

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

$$\mathcal{A}(\beta)(F[t/x]) = \mathcal{A}(\beta)(\forall y' (G[y'/y][t/x])) =$$

$$= \min\{\mathcal{A}(\beta[y' \mapsto a])(G[y'/y][t/x]) \mid a \in U_{\mathcal{A}}\} = \quad (\text{Ind.Vor.})$$

$$= \min\{\mathcal{A}(\beta[y' \mapsto a, x \mapsto \mathcal{A}(\beta[y' \mapsto a])(t)])(G[y'/y]) \mid a \in U_{\mathcal{A}}\} = \quad (\text{Ind.Vor.})$$

$$= \min\{\mathcal{A}(\beta[y' \mapsto a, x \mapsto \mathcal{A}(\beta[y' \mapsto a])(t), y \mapsto a])(G) \mid a \in U_{\mathcal{A}}\} =$$

$$\quad [\text{Rem: } \mathcal{A}(\beta[y' \mapsto a, x \mapsto \mathcal{A}(\beta[y' \mapsto a])(t)])(y') = a]$$

$$= \min\{\mathcal{A}(\beta[x \mapsto \mathcal{A}(\beta)(t), y \mapsto a])(G) \mid a \in U_{\mathcal{A}}\} \quad (y' \text{ kommt in } t \text{ u. } G \text{ nicht vor})$$

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

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

Substitutionen und Valuationen

Theorem (Substitutionslemma)

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

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

Fall 7: $F = \exists yG$.

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 $\beta \circ \sigma : X \rightarrow \mathcal{A}$ 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 $\beta \circ \sigma : X \rightarrow \mathcal{A}$ 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 $F \in \text{For}_\Sigma$ 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 $F \in \text{For}_\Sigma$ 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 $F \in \text{For}_\Sigma$ 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

$$Q_1x_1 \dots Q_nx_n F,$$

wobei F quantorenfrei, $Q_i \in \{\forall, \exists\}$. Hierbei heißt $Q_1x_1 \dots Q_nx_n$ der **Quantorenpräfix** und F die **Matrix** der Formel.

Pränexe Normalform

Definition: Pränexe Formeln sind von der Form

$$Q_1x_1 \dots Q_nx_n F,$$

wobei F quantorenfrei, $Q_i \in \{\forall, \exists\}$. Hierbei heißt $Q_1x_1 \dots Q_nx_n$ 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 $F \in \text{For}_\Sigma$ 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

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

Beispiel 1

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

(1) NNF

1.1. Elimination von \rightarrow

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

Beispiel 1

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

(1) NNF

1.1. Elimination von \rightarrow

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

1.2. "Nach innen schieben" von \neg

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

Beispiel 1

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

(1) NNF

1.1. Elimination von \rightarrow

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

1.2. "Nach innen schieben" von \neg

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

(2) Formel bereinigen

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

Beispiel 1

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

(1) NNF

1.1. Elimination von \rightarrow

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

1.2. "Nach innen schieben" von \neg

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

(2) Formel bereinigen

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

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

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

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

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

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

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

2. Bereinigung

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

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

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

2. Bereinigung

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

3. Alle Quantoren nach vorne \mapsto Pränexnormalform

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