Teil 2: Prädikatenlogik 1. Stufe

2.1 Syntax

Syntax:

- nicht-logische Symbole: Terme, Atomische Formeln

- logiche Symbole: Boole'sche Verknüpfungen, Quantoren

Signatur

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

wobei

- Ω eine Menge von Funktionssymbolen f mit Stelligkeit $n \geq 0$, geschrieben f/n, und
- Π Menge von Prädikatensymbolen p mit Stelligkeit $m \geq 0$, geschrieben p/m.

Falls $n = 0$, heißt f Konstante.

Falls $m = 0$, heißt p Aussagenvariable.

Variablen

Prädikatenlogik erlaubt die Formulierung abtrakter, schematischer Aussagen.

Schematization erfolgt mittels Variablen.

Wir nehmen an, daß

$$X$$

eine vorgegebene Menge von abzählbar unendlich vielen Symbolen ist, die wir für die Bezeichnung von Variablen verwenden.

Terme

Terme über Σ (bzw. Σ-Terme) werden nach folgenden syntaktischen Regeln gebildet:

$$
\begin{array}{llllll}
s, t, u, v & ::= & x & , x \in X & & \text{(Variable)} \\
& | & f(s_1, \ldots, s_n) & , f/n \in \Omega & & \text{(F-Terme)}
\end{array}
$$

Mit $T_\Sigma(X)$ bezeichnen wir die Menge der Σ-Terme

Terme

Terme sind also vollständig geklammerte Ausdrücke, die wir auch als markierte, geordnete Bäume auffassen können.

Die Knoten sind mit Funktionssymbolen oder Variablen markiert.

Jeder mit einem Funktionssymbol f der Stelligkeit n markierte Knoten hat genau n Unterbäume, einen für jedes Argument von f.

Atome

Atome über Σ genügen dieser Syntax:

$$
\begin{array}{rcll}
A, B, \alpha & ::= & p(s_1, \ldots, s_m) & , \; p/m \in \Pi \\
\Big[& \mid & (s \approx t) & \text{(Gleichung)} \quad \Big]
\end{array}
$$

Ist $m = 0$, so handelt es sich bei p um eine Aussagenvariable.

Wir verwenden insbesondere die Buchstaben P, Q, R, S, um Aussagenvariablen zu bezeichnen

Literale

$$
\begin{array}{lcll}
L & ::= & A & \text{(positives Literal)} \\
 & | & \neg A & \text{(negatives Literal)}
\end{array}
$$

Klauseln

$$
\begin{array}{rcll}
C, D & ::= & \bot & \text{(leere Klausel)} \\
 & \mid & L_1 \vee \ldots \vee L_k,\ k \geq 1 & \text{(nichtleere Klausel)}
\end{array}
$$

Formeln

Formeln über Σ:

$$
\begin{array}{rcll}
F, G, H & ::= & \bot & \text{(Falsum)} \\
& \mid & \top & \text{(Verum)} \\
& \mid & A & \text{(atomare Formel)} \\
& \mid & \neg F & \text{(Negation)} \\
& \mid & (F \wedge G) & \text{(Konjunktion)} \\
& \mid & (F \vee G) & \text{(Disjunktion)} \\
& \mid & (F \implies G) & \text{(Implikation)} \\
& \mid & (F \equiv G) & \text{(Aquivalenz)} \\
& \mid & \forall x F & \text{(Allquantifizierung)} \\
& \mid & \exists x F & \text{(Existenzquantifizierung)}
\end{array}
$$

Konventionen zur Notation

- Klammereinsparungen werden nach folgenden Regeln vorgenommen:
  - $\neg \quad >_p \quad \vee \quad >_p \quad \wedge \quad >_p \quad \Longrightarrow \quad >_p \quad \equiv$ (Präzedenzen),
  - $\vee$ und $\wedge$ sind assoziativ und kommutativ,
  - $\Longrightarrow$ ist rechtsassoziativ.

- $Qx_1, \ldots, x_n\ F$ für $Qx_1 \ldots Qx_n\ F$.

Konventionen zur Notation

- Terme und Atome in Infix-, Präfix-, Postfix- oder Mixfixnotation; Beispiele:

$$
\begin{array}{ll}
s + t & \text{für } +(s, t) \\
s \leq t & \text{für } \leq (s, t) \\
-s & \text{für } -(s) \\
0 & \text{für } 0()
\end{array}
$$

Beispiel: Peano-Arithmetik

$$
\begin{aligned}
\Sigma_{PA} &= (\Omega_{PA},\ \Pi_{PA}) \\
\Omega_{PA} &= \{0/0,\ +/2,\ */2,\ s/1\} \\
\Pi_{PA} &= \{\leq/2,\ </2\}
\end{aligned}
$$

$+,\ *,\ \leq,\ <$ infix; $* >_p + >_p < >_p \leq$

Formelbeispiele über dieser Signatur sind

$$\forall x, y (x \leq y \leftrightarrow \exists z (x + z \approx y))$$

$$\exists x \forall y (x + y \approx y)$$

$$\forall x, y (x * s(y) \approx x * y + x)$$

$$\forall x, y (s(x) \approx s(y) \rightarrow x \approx y)$$

$$\forall x \exists y\ x < y$$

Bemerkungen

Wie man an diesen Formeln sieht, sind die Symbole $\leq$, $<$, 0 redundant, da sie über $+$ in PL mit Gleichheit definiert werden können.

So definiert die erste Formel $\leq$, während die zweite Formel die Null definiert.

Die Eliminierung der zugehörigen Existenzquantoren durch Skolemisierung (siehe unten) führt dann aber die Funktionssymbole wieder ein.

Gebundene und freie Variablen

Sei $Q \in \{\exists, \forall\}$.

In QxF, heißt F der Bindungsbereich des Quantors Qx.

Ein Auftreten einer Variablen x heißt gebunden, wenn es zum Bindungsbereich eines Quantors Qx gehört.

Alle anderen Auftreten von Variablen heißen frei.

Formeln ohne freie Variablen heißen Satzformen.

Variablenfreie Formeln heißen Grundformeln.

Beispiel

$$\forall y \quad \overbrace{((\forall \overbrace{x \quad p(x))}^{Bind.} \quad \rightarrow \quad q(x, y))}^{Bindungsbereich}$$

Substitution allgemein

Substitutionen sind Abbildungen

$$\sigma : X \rightarrow T_{\Sigma}(X),$$

so daß der Bereich von σ, d.h. die Menge

$$dom(\sigma) = \{x \in X \mid \sigma(x) \neq x\},$$

endlich ist. Die Menge der eingeführten Variablen, d.h. der Variablen die in einem der Terme $\sigma(x)$, für $x \in dom(\sigma)$, auftreten, wird mit $codom(\sigma)$ bezeichnet.

Substitution allgemein

Notation: $[s_1/x_1, \dots, s_n/x_n]$, x_i pw. verschieden:

$$[s_1/x_1, \dots, s_n/x_n](y) = \begin{cases} s_i, & \text{falls } y = x_i \\ y, & \text{sonst} \end{cases}$$

Notation:

$$\sigma[x \to t](y) = \begin{cases} t, & \text{falls } y = x \\ \sigma(y), & \text{sonst} \end{cases}$$

Substitution eines Termes für eine Variable

Mit $F[s/x]$ bezeichnen wir das Resultat der Substitution aller freien Auftreten von x in F durch den Term s.

$F[s/x]$ sei durch strukturelle Induktion über den Aufbau von F wie folgt definiert:

$$
\begin{aligned}
x[s/x] &= s \\
x'[s/x] &= x' \;; \quad \text{falls } x' \neq x \\
f(s_1, \ldots, s_n)[s/x] &= f(s_1[s/x], \ldots, s_n[s/x]) \\
\bot[s/x] &= \bot \\
\top[s/x] &= \top \\
p(s_1, \ldots, s_n)[s/x] &= p(s_1[s/x], \ldots, s_n[s/x]) \\
(u \approx v)[s/x] &= (u[s/x] \approx v[s/x]) \\
\neg F[s/x] &= \neg(F[s/x]) \\
(F \rho G)[s/x] &= (F[s/x] \rho G[s/x]) \;; \quad \text{für alle binären Junktoren } \rho \\
(QxF)[s/x] &= QxF \\
(QyF)[s/x] &= Qz((F[z/y])[s/x]) \;; \text{falls } y \neq x \;, \; z \text{ neue Variable}
\end{aligned}
$$

Problematik

Der Grund für die Umbenennung der gebundenen Variablen y in eine neue „unbenutzte“ Variable z ist die Vermeidung des Einfangens freier Variablen in s. Sollte y in s auftreten, wären sonst diese Auftreten nach erfolgter Substitution gebunden.

Anwendung einer Substitution

$$
\begin{aligned}
x\sigma &= \sigma(x) \\
f(s_1, \ldots, s_n)\sigma &= f(s_1\sigma, \ldots, s_n\sigma) \\
\bot\sigma &= \bot \\
\top\sigma &= \top \\
p(s_1, \ldots, s_n)\sigma &= p(s_1\sigma, \ldots, s_n\sigma) \\
(u \approx v)\sigma &= (u\sigma \approx v\sigma) \\
\neg F\sigma &= \neg(F\sigma) \\
(F\rho G)\sigma &= (F\sigma\rho G\sigma)\ ; \quad \text{für alle binären Junktoren } \rho \\
(QxF)\sigma &= Qz((F[z/x])\sigma[x \mapsto z]);\ \text{wobei } z \text{ neue Variable}
\end{aligned}
$$

2.2. Semantik

Strukturen

Eine Σ-Algebra (Σ-Interpretation bzw. Σ-Struktur) ist ein Tripel

$$\mathcal{A} = (U,\ (f_A : U^n \to U)_{f/n \in \Omega},\ (p_A \subseteq U^m)_{p/m \in \Pi})$$

wobei $U \neq \emptyset$ eine Menge, genannt Universum von $\mathcal{A}$.

Mit Σ-Alg bezeichnen wir die Menge aller Σ-Algebren.

Wertbelegungen

Variablen für sich haben keine Bedeutung.

Hierfür müssen Wertebelegungen (Valuationen) explizit oder implizit aus dem Kontext zur Verfügung stehen.

Unter einer (Variablen-) Belegung oder einer Valuation (über einer Σ-Algebra $\mathcal{A}$) versteht man eine Abbildung $\beta : X \to U$

Wert eines Terms in $\mathcal{A}$ bzgl. β

$\mathcal{A}(\beta) : T_\Sigma(X) \to \mathcal{A}$ wird induktiv über Aufbau von F wie folgt definiert:

$$\mathcal{A}(\beta)(x) = \beta(x), \qquad x \in X$$

$$\mathcal{A}(\beta)(f(s_1, \ldots, s_n)) = f_A(\mathcal{A}(\beta)(s_1), \ldots, \mathcal{A}(\beta)(s_n)), \qquad f/n \in \Omega$$

Die Menge der Wahrheitswerte sei $\{0, 1\}$.

Wert eines Terms in $\mathcal{A}$ bzgl. β

Im Bingungsbereich eines Quantors, ist Notwendig, die Variablenbelegung zu ändern.

Hierbei bezeichne $\beta[x \to a] : X \to U$ die Belegung, für die

$$\beta[x \to a](y) := \begin{cases} a & \text{falls } x = y \\ \beta(y) & \text{sonst} \end{cases}$$

Wahrheitswert einer Formel in A bzgl. β

Die Menge der Wahrheitswerte sei $\{0, 1\}$.

$\mathcal{A}(\beta)$: Σ-Formeln $\longrightarrow \{0, 1\}$ wird induktiv über Aufbau von F wie folgt definiert:

Wahrheitswert einer Formel in A bzgl. β

$$\mathcal{A}(\beta)(\bot) = 0$$

$$\mathcal{A}(\beta)(\top) = 1$$

$$\mathcal{A}(\beta)(p(s_1, \ldots, s_n)) = 1 \;\Leftrightarrow\; (\mathcal{A}(\beta)(s_1), \ldots, \mathcal{A}(\beta)(s_n)) \in p_A$$

$$\mathcal{A}(\beta)(s \approx t) = 1 \;\Leftrightarrow\; \mathcal{A}(\beta)(s) = \mathcal{A}(\beta)(t)$$

$$\mathcal{A}(\beta)(\neg F) = 1 \;\Leftrightarrow\; \mathcal{A}(\beta)(F) = 0$$

$$\mathcal{A}(\beta)(F \rho G) = \mathrm{B}_\rho(\mathcal{A}(\beta)(F), \mathcal{A}(\beta)(G))$$

mit B_ρ die ρ zugeordnete Boolesche Funktion

$$\mathcal{A}(\beta)(\forall x F) = \min_{a \in U}\{\mathcal{A}(\beta[x \to a])(F)\}$$

$$\mathcal{A}(\beta)(\exists x F) = \max_{a \in U}\{\mathcal{A}(\beta[x \to a])(F)\}$$

Standardinterpretation $\mathbb{N}$ für Peano-Arithmetik

$U_{\mathbb{N}} = \{0, 1, 2, \ldots\}$

$0_{\mathbb{N}} = 0$

$s_{\mathbb{N}} : n \mapsto n + 1$

$+_{\mathbb{N}} : (n, m) \mapsto n + m$

$*_{\mathbb{N}} : (n, m) \mapsto n * m$

$\leq_{\mathbb{N}} = \{(n, m) \mid n \text{ kleiner-gleich } m\}$

$<_{\mathbb{N}} = \{(n, m) \mid n \text{ kleiner als } m\}$

Standardinterpretation $\mathbb{N}$ für Peano-Arithmetik

Mit $\beta : x \mapsto 1, y \mapsto 3$ ergibt sich beispielsweise:

$$\begin{array}{lcl}
\mathbb{N}(\beta)(s(x) + s(0)) & = & 3 \\
\mathbb{N}(\beta)(x + y \approx s(y)) & = & 1 \\
\mathbb{N}(\beta)(\forall x, y(x + y \approx y + x)) & = & 1 \\
\mathbb{N}(\beta)(\forall z\ z \leq y) & = & 0 \\
\mathbb{N}(\beta)(\forall x \exists y\ x < y) & = & 1
\end{array}$$

($\mathbb{N}$ ist nur eine von vielen Σ_{PA}-Interpretationen.)