Was bisher geschah

Wissensrepräsentation und -verarbeitung in Logiken

- klassische Aussagenlogik
- klassische Prädikatenlogik

- Syntax (Wiederholung)
- Semantik (Wiederholung)
- Substitutionen
- Prädikatenlogische Normalformen
  - bereinigte Form
  - Pränexform

Prädikatenlogische Normalformen

Eine Formel $\varphi \in \mathrm{FOL}(\Sigma, X)$ heißt in

- **bereinigter Form**, wenn $\mathrm{bvar}(\varphi) \cap \mathrm{fvar}(\varphi) = \emptyset$ und jeder Quantor eine andere Variable bindet
- **Pränexform**, wenn $\varphi = Q_1 x_1 \cdots Q_n x_n \psi$, wobei $\forall i : Q_i \in \{\forall, \exists\}$ und in ψ keine Quantoren vorkommen.
- **Skolemform**, wenn $\varphi = \forall x_1 \cdots \forall x_m \psi$, wobei $\mathrm{fvar}(\varphi) = \emptyset$ (Satz), alle x_i verschiedene Variablen sind und in ψ keine Quantoren vorkommen.
- **Klauselform**, wenn $\varphi = \forall x_1 \cdots \forall x_m \psi$, wobei alle x_i verschiedene Variablen und ψ eine CNF sind. (Darstellung als Regelmenge)

Erfüllbarkeitsäquivalenz

Definition

Zwei Formeln φ, ψ heißen erfüllbarkeitsäquivalent, wenn gilt:
φ ist genau dann erfüllbar, wenn ψ erfüllbar ist.

(Mod(φ) $\neq \emptyset$ gdw. Mod(ψ) $\neq \emptyset$)

Achtung:

- Erfüllbarkeitsäquivalenz ist schwächer als semantische Äquivalenz.

  Beispiel: $p \vee q$ und r sind erfüllbarkeitsäquivalent,
  aber nicht äquivalent.

- Erfüllbarkeitsäquivalenz bleibt beim Einsetzen in Formeln i.A. nicht erhalten.

  Beispiel: p und q sind erfüllbarkeitsäquivalent,
  aber $p \wedge \neg p$ und $p \wedge \neg q$ nicht.

Konstruktion prädikatenlogischer NF

gebundene Umbenennung $\theta : X \to X$ einer Formel $\varphi \in \mathrm{FOL}(\Sigma, X)$
ersetzt jedes gebundene Variablen-Vorkommen (auch beim bindenden Quantor) durch eine Variable, die in φ noch nicht vorkommt.

Zu jeder Formel $\varphi \in \mathrm{FOL}(\Sigma, X)$ lassen sich konstruieren:

- eine äquivalente Formel in bereinigter Form, (Konstruktion durch gebundene Umbenennungen)
- eine äquivalente Formel in Pränexform, (Konstruktion durch äquivalente Umformungen)
- eine erfüllbarkeits-äquivalente Formel (Satz) in Skolemform, (existenzieller Abschluss und schrittweise Eliminierung der $\exists$ von links nach rechts durch Einführung neuer Funktionssymbole)
- eine Erfüllbarkeits-äquivalente Formel (Satz) in Klauselform.

Beispiele:

- $(\neg \exists x (P(x,z) \vee \forall y Q(x, f(y))) \vee \forall y P(g(z,y), z))$
- $\exists x \forall y \exists z \forall u \exists v P(x,y,z,u,v)$

(Eingeschränkte) Übersetzung in Aussagenlogik

Grundinstanziierung $G : \mathrm{FOL}(\Sigma, \mathbb{X}) \to 2^{\mathrm{AL}(P)}$ in einer Σ-Struktur $\mathcal{A} = (A, [\![\cdot]\!]_{\mathcal{A}})$ (nur die endliche Trägermenge A relevant):
Grundinstanziierung der Formel $\varphi \in \mathrm{FOL}(\Sigma, \mathbb{X})$:

1. Signaturerweiterung $\Sigma' = \Sigma \cup \{(d, 0) \mid d \in A\}$:
   neues Konstantensymbol $(d, 0) \in \Sigma_F$ für jedes $d \in A$
2. Ersetzung aller Quantoren und gebundenen Variablen:
   $\forall x \varphi \mapsto \bigwedge_{d \in A} \varphi\{x \mapsto d\}$, $\exists x \varphi \mapsto \bigvee_{d \in A} \varphi\{x \mapsto d\}$,
3. Ersetzung $\beta : \mathrm{FOL}(\Sigma, \mathbb{X}) \to 2^{\mathrm{FOL}(\Sigma', \emptyset)}$ aller freien Variablen:
   Formelmenge $\beta(\varphi) =$
   $\{\varphi\{x_1 \mapsto a_1, \ldots, x_n \mapsto a_n\} \mid \mathrm{fvar}(\varphi) = \{x_1, \ldots, x_n\}, a_1, \ldots, a_n \in A\}$
4. Ersetzung (Umbenennung) jedes Grundatoms $a \in \mathrm{Atom}(\beta(\varphi))$
   durch eine Aussagenvariable $p \in P$
   (z.B. $p(a_1, \ldots, a_n) \mapsto pa_1 \cdots a_n$)

Beispiele (Tafel): Grundinstanzen von

- $p(x, x) \wedge \exists x\, p(y, x)$ in $A = \{1, 2\}$,
- $\forall x (P(x) \to \exists y N(y, x))$ in $B = \{$ Anton, Tom, Paul $\}$

Prädikatenlogik: Grundinstanziierung

Vorteile:

- Entscheidbarkeit,
- aussagenlogische Methoden anwendbar,
- Standard-Werkzeuge einsetzbar, z.B. SAT-Solver

Nachteile:

- ergibt evtl. große unübersichtliche Formelmengen,
- nur möglich, falls beide folgende Bedingungen erfüllt sind:
  1. Struktur hat endliche Trägermenge
  2. Signatur enthält keine >0-stelligen Funktionen (nur Konstanten)