

Unentscheidbarkeit

Satz von Church

Das Gültigkeitsproblem der Prädikatenlogik ist unentscheidbar.

Beweis s. Schönig Kap.2.3 (Rückführ. auf Postsches Korrespondenzproblem)

Satz

Das Erfüllbarkeitsproblem d. Prädikatenlogik ist unentscheidbar.

Beweis s. Schönig Kap. 2.3 (Rückführung auf Satz v. Church)

Satz

Erfüllbarkeit und Gültigkeit sind semi-entscheidbar:
Es gibt Kalküle, die genau alle gültigen Formeln herleiten.

Beweis s.u.: Wir werden solche Verfahren kennen lernen

Bemerkung: Es gibt entscheidbare Untermengen der PL (z.B. AL)

Äquivalenzen

... aus AL gelten entsprechend weiter (s. Folie 37)! Zusätzlich:

$\neg \forall x. \alpha \equiv \exists x. \neg \alpha$	deMorgansche Regel
$\neg \exists x. \alpha \equiv \forall x. \neg \alpha$	deMorgansche Regel
$\forall/\exists x. \alpha \Diamond \beta \equiv \forall/\exists x. (\alpha \Diamond \beta)$	Skopuserweiterung falls x nicht frei in β ; $\Diamond \in \{\wedge, \vee\}$
$(\forall x. \alpha \wedge \forall x. \beta) \equiv \forall x. (\alpha \wedge \beta)$ $(\exists x. \alpha \vee \exists x. \beta) \equiv \exists x. (\alpha \vee \beta)$	
$\forall x. \forall y. \alpha \equiv \forall y. \forall x. \alpha$ $\exists x. \exists y. \alpha \equiv \exists y. \exists x. \alpha$	

Beweis je „zu Fuß“ gemäß Def. Wahrheit quantifizierter Formeln, s. Folie 74

Skolemform

Wie in der AL brauchen wir in der PL Normalformen.
Neu zu behandeln sind die Quantoren!

Sei $\alpha = Q_1 x_1 \dots Q_n x_n \cdot \alpha'$ Formel in **Pränex-Normalform**, d.h. Q_1, \dots, Q_n sind Quantoren und α' ist quantorenfrei. Seien Q_1, \dots, Q_i ($i \in \{0, \dots, n\}$) Allquantoren, sei also Q_{i+1} der erste auftretende Existenzquantor. Sei f^i eine neue, in α' nicht vorkommende i -stellige Funktion, sog. **Skolem-Funktion**.

Die Ersetzung von x_{i+1} überall in α' durch $f^i(x_1, \dots, x_i)$ und Streichung des Quantors $Q_{i+1} x_{i+1}$ heißt **\exists -Elimination** oder **Skolemisierung**.

Eine Formel in Pränex-Normalform, in der alle \exists -Quantoren eliminiert sind, heißt in **Skolemform**.

Skolemisierung

Beispiel

Skolemisierung von $\forall x. \exists y. \exists t. \text{loves}(x,y,t)$
für Skolem-Funktionen $f(.)$, $g(.)$: $\forall x. \text{loves}(x,f(x),g(x))$

Intuitive Interpretation

$f(.)$ bezeichnet den/die Geliebte/n, $g(.)$ den „Lebensabschnitt“

Ist Skolemisierung eine Äquivalenzumformung?



Aber zum Glück gilt der **Satz**

Überführen einer Formel in Skolemform ist inkonsistenzerhaltend:
 α skolemisiert ist inkonsistent gdw. α inkonsistent ist.

Beweis: Betrachtung der Modelle

Prädikatenlogische Formel umformen in KNF

1. Benenne gebundene Variablen eindeutig
2. Löse \Leftrightarrow und \Rightarrow auf (s. Folie 37)
3. Bringe alle Negationszeichen direkt vor die Atomformeln, löse dabei doppelte Negation immer auf (deMorgansche Regeln, auch für Quantoren, Folie 79)
4. Schiebe alle Quantoren nach links (Pränex-NF; nutze Äquivalenzen Folie 79) und überführe die Formel in Skolemform
5. Überführe den Teil „hinter den Quantoren“ in KNF wie in AL, falls noch erforderlich

Ergebnis: Formel in Skolemform $\forall x_1 \dots \forall x_n. \alpha$, wobei α in KNF

Beispiel: Formel um: $\forall x. \forall y. [\exists z. (P(x,z) \wedge P(y,z)) \Rightarrow \exists z. Q(x,y,z)]$

an der Tafel

Tafelbeispiel

$$\forall x.\forall y.[\exists z.(P(x,z) \wedge P(y,z)) \Rightarrow \exists z.Q(x,y,z)]$$

$$\equiv \forall x.\forall y.[\exists u.(P(x,u) \wedge P(y,u)) \Rightarrow \exists v.Q(x,y,v)]$$

$$\equiv \forall x.\forall y.[\neg \exists u.(P(x,u) \wedge P(y,u)) \vee \exists v.Q(x,y,v)]$$

$$\equiv \forall x.\forall y.[\forall u.\neg(P(x,u) \wedge P(y,u)) \vee \exists v.Q(x,y,v)]$$

$$\equiv \forall x.\forall y.[\forall u.(\neg P(x,u) \vee \neg P(y,u)) \vee \exists v.Q(x,y,v)]$$

$$\equiv \forall x.\forall y.\forall u.[\neg P(x,u) \vee \neg P(y,u) \vee \exists v.Q(x,y,v)]$$

$$\equiv \forall x.\forall y.\forall u.\exists v.[\neg P(x,u) \vee \neg P(y,u) \vee Q(x,y,v)]$$

$$! \neq \forall x.\forall y.\forall u.[\neg P(x,u) \vee \neg P(y,u) \vee Q(x,y,f(x,y,u))]$$

(inkonsistenz-
erhaltend)

Klauseln in der Prädikatenlogik

Analog zur Aussagenlogik definieren wir

Ein **Literal** ist eine negierte oder nicht negierte Atomformel.

Eine **Klausel** ist eine Disjunktion von Literalen

Umformen einer Formel in Skolemform und KNF in Klauselform:

- Lass alle (All-)Quantoren weg;
- interpretiere alle Variablen als allquantifiziert;
- stelle Klausel als Menge von Literalen dar;
- stelle KNF-Formel als Menge von Klauseln dar.

Beispiel

$$\forall x. \forall y. \forall u. [\neg P(x,u) \vee \neg P(y,u) \vee Q(x,y,f(x,y,u))]$$



$$\{\{\neg P(x,u), \neg P(y,u), Q(x,y,f(x,y,u))\}\}$$

Definitionen aus der Herbrand-Theorie

Das **Herbrand-Universum** H_S einer Klauselmenge S ist die kleinste abgeschlossene Menge aller Grundterme, die folgendermaßen gebildet werden:

1. Alle in S vorkommenden Konstanten sind in H_S .
Kommt keine Konstante in S vor, setze die Konstante a in H_S .
2. Für jede Funktion f_i^k in S u. Terme t_1, \dots, t_k ist $f_i^k(t_1, \dots, t_k) \in H_S$.

Die **Herbrand-Basis** $H_S(S)$ einer Klauselmenge S ist die Menge der Grundklauseln, die aus S durch konsistentes Einsetzen aller Elemente aus H_S für alle Variablen in allen Klauseln entstehen.

Bspl.: $S = \{\{\neg P(x,u), \neg P(y,u), Q(x,y,f(x,y,u))\}\}$, dann ist $H_S(S)$ die (unendl.) Klauselmenge, in der x, y, u auf alle Arten ersetzt sind mit $a, f(a,a,a), f(f(a,a,a),a,a), f(a,f(a,a,a),a), \dots, f(f(f(a,a,a),\dots)), \dots$

Der Satz von Herbrand

Details s. Schöning Kap.2.4

Satz von Herbrand (Version für Klauselform)

Eine Menge S von Klauseln ist inkonsistent, gdw. endliche Teilmenge der Herbrand-Basis $H_S(S)$ existiert, die inkonsistent ist.

Beweis: Betrachtung der „Herbrand-Modelle“ von S (\approx Modelle über dem Herbrand-Universum H_S):

Satz von Gödel/Herbrand/Skolem in Kombination m. Endlichkeitssatz f. AL.

Vor Einführung der Resolution wurden Algorithmen untersucht, die diesen Satz „naiv“ implementierten:

Erzeuge systematisch die Herbrand-Basis;
prüfe jede endliche Teilmenge aussagenlogisch auf Inkonsistenz

2.4 Resolution in der Prädikatenlogik

Wegen impliziter Allquantifizierung der Variablen gilt:

- $P(x), P(y)$ semantisch gleich (also $P(x), \neg P(y)$ widersprüchlich);
- $P(x), \neg P(f(a))$ widersprüchlich, ebenso $P(x), \neg P(f(y))$;
aber nicht
- $P(x), \neg P(f(x))$!

Für vollständige Inferenzverfahren müssen wir semantisch gleiche Literale auch textuell gleich machen, **unifizieren**!

Eine **Substitution** ist eine Folge von Ersetzungen x/t einer Variablen x durch einen Term t , wobei x in t nicht vorkommt(!).

Bemerkung: Durch Anwendung einzelner Ersetzungen auf den ersetzenden Term früherer Ersetz. erzeuge Reihenfolgeunabhängigkeit der Ersetzungen

↪ Substitutionen werden Mengen von x/t -Paaren.

Unifikation

Eine Substitution, die eine Menge von Atomformeln textuell gleich macht, heißt **Unifikator**. (Notation: x/a „ a ersetzt x “)

Unifikatoren (wenn es sie gibt) sind nicht eindeutig! **Beispiel:**

Für $P(x,y)$ und $P(a,z)$ Unifikator $\{x/a, y/z\}$ ($\mapsto P(a,z)$)

aber auch $\{x/a, z/y\}$ ($\mapsto P(a,y)$) und $\{x/a, y/a, z/a\}$ ($\mapsto P(a,a)$)

Ein Unifikator θ einer Menge \mathcal{L} von Literalen heißt **allgemeinster Unifikator** (*most general unifier*, MGU), falls es für jeden Unifikator θ' von \mathcal{L} eine Substitution σ gibt mit $\theta' = \theta\sigma$.

Bspl: $\{x/a, z/y\}$ ist allgemeiner als $\{x/a, y/a, z/a\} = \{x/a, z/y\}\{y/a, z/a\}$

Unifikationssatz von Robinson

Jede unifizierbare Literalmenge hat einen MGU.

Bemerkung: Der MGU ist eindeutig bis auf Variablen-Umbenennungen.

Beweis: Konstruktiv durch Algorithmus UNIFY (z.B. Russell/Norvig)

UNIFY am Beispiel eingeführt

UNIFY setzt zwei Terme von links nach rechts komponentenweise gleich, bis beide fertig sind oder zwei ungleiche Fkt.-Symbole auftreten.

$$\begin{array}{l} \neg P(\underline{f(z)}, g(a, y)), h(z)) \\ \neg P(\underline{f(f(u, b))}, w), h(f(a, b))) \end{array} \rightarrow \{z / f(u, b)\}$$

Genau u, w, y, z
sind hier Variablen!

$$\begin{array}{l} \neg P(\underline{f(f(u, b))}, \underline{g(a, y)}), h(f(u, b))) \\ \neg P(\underline{f(f(u, b))}, \underline{w}), h(f(a, b))) \end{array} \rightarrow \{z / f(u, b), w / g(a, y)\}$$

$$\begin{array}{l} \neg P(\underline{f(f(u, b))}, g(a, y)), h(\underline{f(u, b)})) \\ \neg P(\underline{f(f(u, b))}, g(a, y)), h(\underline{f(a, b)})) \end{array} \rightarrow \{z / f(u, b), w / g(a, y), u / a\}$$

$$\begin{array}{l} \neg P(f(f(a, b), g(a, y)), h(f(a, b))) \\ \neg P(f(f(a, b), g(a, y)), h(f(a, b))) \end{array}$$

Komplexität der Unifizierung

- „Naive“ Unifizierung (UNIFY) quadratisch wegen OCCUR-CHECK (Test auf „wobei x in t nicht vorkommt“, s. Folie 87)
- Es gibt Unifizierungsalgorithmen mit linearer Laufzeit in der Länge der Terme (mit relativ großen Konstanten!)

Manche Implementierungen, z.B. typische PROLOG-Interpreter(!), lassen den OCCUR-CHECK weg (Effizienz)

Beispiel SWI-PROLOG:

```
p(x, x) .  
?- p(f(y), y) .  
   y = f(**)  
Yes
```

Für Programmierung ist das praktisch i.d.R. irrelevant.

Resolution, prädikatenlogische Variante

Die Klausel $R = \text{Res}(K_1, K_2)$ ist **Resolvente** zweier Klauseln K_1 und K_2 mit disjunkten Variablen (ggf. substituieren!), gdw:

1. Es gibt positive Literale $L_1, \dots, L_m \in K_1$ ($m \geq 1$) und negative $L'_1, \dots, L'_n \in K_2$ ($n \geq 1$), sodass $\mathcal{L} = \{L_1, \dots, L_m, L'_1, \dots, L'_n\}$ unifizierbar ist mit MGU θ .
(Für $m > 1$ oder $n > 1$: Literale derselben Klausel mit gleichem Vorzeichen werden **faktorisiert**.)
2. $R = [(K_1 - \{L_1, \dots, L_m\}) \cup (K_2 - \{L'_1, \dots, L'_n\})]\theta$

Beispiele

$$\{P(x, y), P(f(z), a)\} \quad \{\neg P(u, v)\}$$

$$\{P(f(z), a)\}$$

$$\theta = \{x/u, y/v\}$$

$$\{P(x, y), P(f(z), a)\} \quad \{\neg P(u, v)\}$$



$$\theta = \{x/f(z), y/a, u/f(z), v/a\}$$