Fixpunktsemantik logischer Programme

Pascal Hitzler
Juli 1997

Kurzüberblick im Rahmen der Vorlesung *Einführung in Prolog* von T. Cornell im Sommersemester 1997 an der Universität Tübingen. Beweise sind nur skizziert.

Inhaltsverzeichnis

0 Grundbegriffe **1**

1 Interpretationen und Modelle **2**

2 Herbrand-Interpretationen und -Modelle **3**

3 Fixpunkte **5**

4 Normale logische Programme **6**

Literatur **6**

0 Grundbegriffe

Grundbegriffe und -zusammenhänge aus [1], Kapitel *Elements of Prolog: Expressions, Unification and Resolution* werden vorausgesetzt, sowie Grundkenntnisse im Programmieren mit Prolog. Dies betrifft vor allem Begriffe wie *Sprache erster Stufe, Term, Atom, Literal, Grundinstanz, definite Programmklausel, definites Programm* etc.

Hauptreferenz für die folgenden Ausführungen ist [2].

Ist P ein definites logisches Programm, so bezeichnen wir die zugehörige Sprache erster Stufe mit $\mathcal{L}_P$.

Beispiel 0.1
Programm P:

```
p(a).
p(s(X)) :- p(X).
```

Die zugehörige Sprache erster Stufe enthält:
Konstantensymbole: a
Funtionssymbole: s/1
Prädikatensymbole: p/1
Das Programm P entspricht folgendem logischen Ausdruck erster Stufe:
$p(a) \wedge \forall x(p(x) \rightarrow p(s(x)))$ oder auch

$$p(a) \wedge \forall(p(x) \rightarrow p(s(x))).$$

1 Interpretationen und Modelle

Definition 1.1 (Präinterpretation) Eine *Präinterpretation* einer Sprache $\mathcal{L}$ erster Stufe besteht aus

1. Einer Menge $D \neq \emptyset$, genannt der *Bereich* der Präinterpretation.
2. Für jedes Konstantensymbol in $\mathcal{L}$, die Zuweisung eines Elements in $\mathcal{L}$.
3. Für jedes n-stellige Funktionssymbol in $\mathcal{L}$, die Zuweisung einer Abbildung von D^n nach D.

Definition 1.2 (Interpretation) Eine Interpretation I einer Sprache $\mathcal{L}$ erster Stufe besteht aus einer Präinterpretation J von $\mathcal{L}$ mit Bereich D und für jedes n-stellige Prädikatensymbol in $\mathcal{L}$ der Zuweisung einer Abbildung von D^n nach $\{\text{wahr}, \text{falsch}\}$ (oder, äquivalent dazu, einer Relation auf D^n).

Definition 1.3 (Wahrheitswert) Sei P ein normales logisches Programm, $\mathcal{L}_P$ die zugehörige Sprache erster Stufe und I eine Interpretation von $\mathcal{L}$ mit zugehöriger Präinterpretation J und Bereich D.
Ist A_g Grundinstanz eines Atoms A in P, so wird A_g der Wahrheitswert *wahr* (bezüglich I) zugewiesen, falls nach Zuweisung der Konstanten- und Funktionssymbole bezüglich J und Auswertung der zugewiesenen Funktionen, dem Atom A_g bezüglich der Interpretation I der Wert *wahr* zugewiesen wird. Andernfalls wird A_g der Wahrheitswert *falsch* zugeweisen.
Ist C_g Grundinstanz einer Klausel C in P, so wird ihr ein Wahrheitswert zugewiesen, indem zunächst jedem Atom ein Wahrheitswert wie oben beschrieben zugewiesen wird,

und dann der resultierende Ausdruck bezüglich der üblichen Verknüpfungstafeln für Junktoren einem Wahrheitswert zugewiesen wird. Der Klausel C wird der Wahrheitswert *wahr* zugewiesen, falls jeder Grundinstanz von C der Wert *wahr* zugewiesen wird. Dem Programm P heißt *wahr bezüglich* I, falls jeder Klausel in P der Wert *wahr* zugewiesen wird.

Definition 1.4 (Modell) Sei P ein normales logisches Programm, I eine Interpretation der zugehörigen Sprache erster Stufe und C eine Klausel in P. Dann heißt I ein *Modell* von C, wenn C wahr bezüglich I ist. I heißt ein *Modell* von P, wenn jede Klausel in P wahr bezüglich I ist.

Beispiel 1.5 Betrachte das Programm P in Beispiel 0.1 mit folgender Interpretation I:
Bereich: $\mathbb{N}$
Konstanten: $a \mapsto 0 \in \mathbb{N}$
Funktionen: $s \mapsto (x \mapsto x + 2)$
Prädikate:

$$p \mapsto \begin{cases} x \mapsto \text{wahr} & \text{falls } x \text{ gerade} \\ x \mapsto \text{falsch} & \text{falls } x \text{ ungerade} \end{cases}$$

Dann ist I ein Modell von P.

Definition 1.6 (Logische Konsequenz) Sei P ein normales logisches Programm und A ein Atom über $\mathcal{L}_P$. A heißt eine *logische Konsequenz* von P, wenn für jedes Modell M von P, A wahr bezüglich M ist. Man schreibt dann $P \models A$.

2 Herbrand-Interpretationen und -Modelle

Definition 2.1 (Herbrand-Basis) Sei P ein logisches Programm. Das *Herbrand-Universum* U_P von P besteht aus der Menge aller Grundinstanzen von Termen in $\mathcal{L}_P$. Die Herbrandbasis B_P von P besteht aus der Menge aller Grundinstanzen von Prädikatensymbolen aus $\mathcal{L}_P$ mit Termen aus U_P als Argumenten.

Definition 2.2 (Herbrand-Präinterpretation) Sei P ein logisches Programm. Die *Herbrand-Präinterpretation* für P ist gegeben durch:

1. Bereich U_P.
2. Konstantensymbole werden sich selbst zugeordnet.
3. Jedem n-stelligen Funktionssymbol f wird die Funktion $(t_1, \ldots, t_n) \mapsto f(t_1, \ldots, t_n)$ zugeordnet.

Definition 2.3 (Herbrand-Interpretation) Eine *Herbrand-Interpretation* für ein logisches Programm P ist eine Interpretation bezüglich der Herbrand-Präinterpretation für P. Jeder Herbrand-Interpretation von P kann eindeutig eine Teilmenge von B_P zugeordnet werden und umgekehrt.

Definition 2.4 (Herbrand-Modell) Ein *Herbrand-Modell* für ein logisches Programm P ist eine Herbrand-Interpretation für P, die ein Modell für P ist.

Beispiel 2.5 Ein Herbrand-Modell für P aus Beispiel 0.1 ist gegeben durch die Menge $\{p(s^n(a)) \mid n \in 2\mathbb{N}\}$.

Satz 2.6 Ist P ein logisches Programm, so hat P genau dann ein Modell, wenn P ein Herbrand-Modell hat.

Beweis: Ist I ein Modell von P, so ist

$$I' := \{p(t_1, \ldots, t_n) \in B_p \mid p(t_1, \ldots, t_n) \text{ wahr bezüglich } I\}$$

ein Herbrand-Modell von P. ■

Satz 2.7 (Model Intersection Property) Ist P ein definites Programm und $\{M_i\}_{i \in I}$ eine nichtleere Menge von Herbrand-Modellen von P, so ist $M := \bigcap_{i \in I} M_i$ ein Herbrand-Modell von P.

Beweis: M ist offensichtlich eine Herbrand-Interpretation. Es folgt leicht, daß M auch ein Modell ist. ■

Korollar 2.8 Sei P ein definites logisches Programm. Dann existiert ein kleinstes Herbrand-Modell M_P von P.

Beweis: B_P ist ein Herbrand-Modell von P. Der Schnitt über alle Herbrand-Modelle von P ist somit nach Satz 2.7 ein Herbrand-Modell von P, und nach Konstruktion das kleinste. ■

Satz 2.9 Sei P ein definites logisches Programm. Dann ist $M_P = \{A \in B_P \mid P \models A\}$.

Beweis: [2], Theorem 6.2 ■

Korollar 2.10 Sei P ein definites logisches Programm. Dann ist

$$M_P = \{A \in B_P \mid A \text{ folgt aus } P \text{ durch SLD-Resolution}\}.$$

3 Fixpunkte

Definition 3.1 Für ein normales logisches Programm bezeichne I_P die Menge aller Herbrand-Interpretationen von P. Man kann identifizieren: $I_P = \mathcal{P}(B_P)$.

Definition 3.2 (Konsequenzoperator) Sei P ein normales logisches Programm. Definiere den Konsequenzoperator von P als $T_P : I_P \to I_P$ durch $T_P(I) := \{H \in B_P \mid (H \leftarrow A_1, \ldots, A_k, B_1, \ldots, B_l)$ ist Grundinstanz einer Klausel in P und $A_i \in I, B_i \notin I\}$

Satz 3.3 Sei P ein normales logisches Programm. $I \in I_P$ ist genau dann ein Herbrand-Modell von P, wenn $T_P(I) \subseteq I$.

Beweis: I ist genau dann ein Modell von P, wenn für jede Grundinstanz $H \leftarrow A_1, \ldots, A_k$ jeder Klausel in P gilt:

$$\{A_1, \ldots, A_k\} \subseteq I \Longrightarrow H \in I.$$

Dies ist genau dann der Fall, wenn $T_P(I) \subseteq I$. ■

Proposition 3.4 Sei P ein normales logisches Programm. Dann ist I_P eine ω-cpo, das heißt $\subseteq$ ist eine partielle Ordnung auf I_P, $\emptyset \subseteq I$ für alle $I \in I_P$ ($\emptyset$ heißt *kleinstes Element von D*) und für $I_i \in I_P$ ($i \in \mathbb{N}$) mit $I_i \subseteq I_{i+1}$ ($(I_i)_{i\in\mathbb{N}}$ nennt man eine ω-Kette in I_P) gilt $\sup I_i := \bigcup I_i \in I_P$.

Beweis: Klar. ■

Proposition 3.5 Sei P ein definites logisches Programm. Dann ist T_P *ω-stetig*, das heißt:

1. $I_1 \subseteq I_2 \Longrightarrow T_P(I_1) \subseteq T_P(I_2)$ (T_P ist *monoton.*)
2. Für jede ω-Kette (I_i) in I_P gilt $T_P(\sup I_i) = \sup T_P(I_i)$.

Beweis: Monotonie ist klar. Für die ω-Stetigkeit zeigt man, daß $A \in T_P(\sup I_i) \Longleftrightarrow A \in \sup T_P(I_i)$. ■

Satz 3.6 (Fixpunktsatz von Knaster-Tarski) Ist D eine $\omega-$cpo mit kleinstem Element $\perp$ und $f : D \to D$ eine $\omega-$stetige Funktion, so hat f einen kleinesten Fixpunkt x mit $x = \sup f^n(\perp)$. Dabei ist $f^0(\perp) = \perp$ und $f^{n+1}(\perp) = f(f^n(\perp))$.

Beweis: Die Folge der $f^n(\perp)$ bildet eine ω-Kette in D aufgrund der Monotonie von f. Aufgrund der Stetigkeit von f ist x Fixpunkt. Ist y Fixpunkt von f, so ist $f^n(\perp) \leq f^n(y) = y$ für alle n aufgrund der Monotonie von f. Also ist $x \leq y$. ■

Satz 3.7 (Kowalski, van Emden) Ist P ein definites logisches Programm, so ist M_P gleich dem kleinesten Fixpunkt von T_P.

Beweis:

$$\begin{aligned} M_P &= \inf\{I \mid I \text{ ist Herbrand-Modell von } P\ \} \\ &= \inf\{I \mid T_P(I) \subseteq I\} \\ &= \sup\{I \mid T_P(I) = I\} \\ &= \text{ kleinster Fixpunkt von } T_P \end{aligned}$$

■

4 Normale logische Programme

Die oben für defnite logische Programme erhaltenen Ergebnisse sind im allgemeinen nicht auf normale logische Programme übertragbar.

Herbrand-Modelle

Die Existenz eines kleinsten Herbrand-Modells ist für normale logische Programme nicht mehr gewährleistet. Es ist also a priori nicht klar, wie Negation verstanden werden soll. Eine mögliche Antwort liefert die SLDNF-Resolution. Es gibt aber auch andere Ansätze, die je nach vorliegendem Programm sinvoll sind.

Konsequenzoperator

Der Konsequenzoperator T_P für ein gegebenes normales logisches Programm P ist im allgemeinen nicht mehr monoton. Der Fixpunktsatz von Knaster-Tarski kann dann nicht mehr angewendet werden. Es ist ein aktuelles Forschungsgebiet, nach Fixpunkten von T_P in diesem Fall zu suchen, bzw. nach Kriterien, wann solche Fixpunkte existieren.

Literatur

[1] T. Cornell, *Introduction to Prolog*, Lecture Notes, (1996), erhältlich über `http://www.sfs.nphil.uni-tuebingen.de/~cornell/teaching.html`

[2] J.W. Lloyd, *Foundations of Logic Programming*, Springer Berlin, ^{2}1988