

Programmierparadigmen: logische Programmierung

Clemens Degasper

6. Juni 2013

Zusammenfassung

Logische Programmierung ist eine der vier Hauptprogrammierparadigma. Es beruht auf mathematischer Logik. Ein logisches Programm besteht aus mehreren Fakten und Regeln. Der Benutzer muss sich nicht um den Lösungsweg kümmern. Er beschreibt Eigenschaften, die eine Lösung erfüllen muss und der Interpreter versucht die Lösungsaussage alleine aus den gegebenen Daten zu berechnen.

1 Einführungsbeispiel

Ein kurzes Beispiel um zu zeigen wie logische Programmierung funktioniert. Die hier gezeigten Beispiele sind alle in Prolog geschrieben, die erste und immernoch bekannteste logische Programmiersprache. Das typische erste Programm in Prolog ist nicht wie in prozeduralen Programmiersprachen ein Hallo-Welt-Beispiel, sondern eine Datenbasis mit Stammbauminformationen.

Regeln:

$$\begin{aligned} \textit{grandparent}(X, Z) &: \textit{-parent}(X, Y), \textit{parent}(Y, Z). \\ \textit{parent}(X, Y) &: \textit{-father}(X, Y). \\ \textit{parent}(X, Y) &: \textit{-mother}(X, Y). \end{aligned}$$

Fakten:

$$\begin{aligned} \textit{father}(\textit{bill}, \textit{jack}). & \quad \textit{mother}(\textit{mary}, \textit{jack}). \\ \textit{father}(\textit{bob}, \textit{bill}). & \quad \textit{mother}(\textit{jane}, \textit{mary}). \end{aligned}$$

Anfragen:

$$\begin{aligned} ? \textit{-grandparent}(\textit{bob}, \textit{jack}). & \quad \textit{yes} \\ ? \textit{-grandparent}(X, \textit{jack}). & \quad X = \textit{bob}; \\ & \quad X = \textit{jane}; \end{aligned}$$

Wie man hier sieht werden vom Benutzer nur einige Regeln und Fakten definiert und der Interpreter kennt welche Eigenschaften er zur Lösung der Anfragen benutzen muss. Ein positives Resultat bedeutet, dass die Anfrage logisch ableitbar ist. Ein negatives Resultat bedeutet nur, dass aufgrund der bekannten Eigenschaften keine Ableitung gefunden werden kann.

2 Regeln und Fakten

Ein Prolog Programm besteht immer aus Regeln und Fakten, sogenannten Formeln. Prolog ist eingeschränkt auf eine spezielle Art von Formeln, die Horn-Formeln.

2.1 Horn-Formeln

Horn Formeln sind prädikatenlogische Formeln. Sie werden hauptsächlich in der logischen Programmierung und der Logik verwendet. Eine Horn-Formel ist eine konjunktive Normalform (eine Konjunktion von Disjunktionen).

Beispiel:

$$(\neg p \vee r) \wedge (q \vee \neg s) \wedge (\neg r)$$

Horn-Klauseln lassen sich nach den Regeln der Aussagenlogik auch als materiale Implikationen darstellen.

$$\neg\phi \vee \psi = \phi \Rightarrow \psi$$

Regeln werden als Horn-Formeln dargestellt.

2.2 Fakten

<i>male(bill).</i>	<i>female(mary).</i>
<i>male(jack).</i>	<i>female(jane).</i>
<i>male(bob).</i>	<i>father(bill, jack).</i>
<i>mother(mary, jack).</i>	<i>father(bob, bill).</i>
<i>mother(jane, mary).</i>	

Bill ist männlich. Bill ist der Vater von Jack. Bob ist der Vater von Bill usw.

Der Interpreter kennt nun diesen kleinen Stammbaum und kann auch Anfragen bearbeiten. In Prolog gibt es zwei Anfragetypen : Ja/Nein-Anfragen und Suchanfragen.

? – male(Bob).

yes.

? – female(X).

X = mary

X = jane

2.3 Regeln

Neben den Fakten lassen sich auch Regeln festlegen. Regeln werden als Implikationen geschrieben. Das “ :- “ ist gleich bedeutend mit einer logischen Implikation nach links \Leftarrow .

grandparent(X, Z) : –parent(X, Y), parent(Y, Z).

Wenn X ein Elternteil von Y ist UND Y ein Elternteil von Z ist, dann ist X Großelternteil von Z. Der “,“ hat in dieser Regel die selbe Bedeutung wie ein Logisches und.

parent(X, Y) : –father(X, Y).

parent(X, Y) : –mother(X, Y).

Wenn X der Vater von Y ist ODER X die Mutter von Y, dann ist X ein Elternteil von Y. Da hier die parent Regel zweimal definiert wird, kann man die doppelte definiert als Logisches oder interpretieren.

Literatur

- [1] Leon Sterling and Ehud Shapiro, The Art of Prolog, The MIT Press, 1994.