

Prolog

Autor: Maximilian Ronniger
E-Mail: e0426625@student.tuwien.ac.at

Basierend auf dem Artikel „Describing Prolog by its interpretation and compilation“ von Jacques Cohen.

Abstrakt

Dieser Artikel bietet eine Kurze Einführung in die Programmiersprache Prolog. Dabei werden auch die Gründe für die Entwicklung der Sprache durch Alain Colmerauer und sein Team behandelt. Es wird aber auch auf Praxisnahe Themen wie die Syntax und Semantik der Sprache eingegangen. Auch wird auf Besonderheiten der Sprache eingegangen und Anwendungen aufgezeigt. Danach wird eine Auflistung der Vor- und Nachteile von Prolog durchgeführt, die je nach Anwendungsbereich zu unterschiedlichen Ergebnissen führt. Am Ende des Artikels befindet sich die Literaturliste, diese soll dem interessierten Leser zum Finden weiterführender Literatur dienen.

Inhalt:

| | | |
|-----|--|---|
| 1 | Motivation zur Entwicklung der Sprache | 2 |
| 2 | Überblick über Syntax und Semantik..... | 3 |
| 2.1 | Allgemeines..... | 3 |
| 2.2 | Syntax | 3 |
| 2.3 | Semantik..... | 4 |
| 2.4 | Beispiel | 5 |
| 3 | Besonderheiten und Anwendungsgebiete..... | 5 |
| 4 | Zusammenfassung: Vor- und Nachteile | 6 |
| 5 | Literaturliste | 6 |

1 Motivation zur Entwicklung der Sprache

Das Ziel der Entwicklung von Prolog war es eine Mensch-Maschinen Kommunikation in einer Natürlichen Sprache (in diesem Fall Französisch) zu schaffen. Diese sollte es der Maschine ermöglicht nach Eingabe von Informationen in Form von natürlichen Setzen Antworten (auch in Natürlichen Sätzen) zu geben. Die Schwierigkeit bestand darin, dass die Auswertung der Semantik mit herkömmlichen Programmiersprachen nur schwer auszuwerten war. Die ersten Lösungsansätze für dieses Problem wurden in Q-Systems durchgeführt und brachten schon recht brauchbare Ergebnisse, es traten aber Probleme auf die mit den zu Verfügung stehenden Mitteln nur unzureichend gelöst werden konnten. Also entschlossen sich Alain Colmerauer, Robert Pasero, Philippe Roussel und Jean Trudel eine neue Sprache zu Schreiben. Prolog profitiert von den Erfahrungen, die das Team mit Q-Systems und automatischer Deduktion gesammelt hatte. Die Entwicklung von Prolog war in zwei geteilt. Die eine Teil war das Natürlichsprachliche Interface (Alain Colmerauer , Robert Pasero), der andere die automatische Deduktion (Jean Trudel, Philippe Roussel). In den Jahren von 1971 bis 1973 Entstand dann die erste Version von Prolog. In dieser Zeit, genauer im Jahr 1972, entstand auch der Name PROLOG („ PROgrammation en LOGique “). In den Folgenden Jahren 1974 und 1975 fand Prolog auch den weg auf andere Universitäten.[1]

2 Überblick über Syntax und Semantik

2.1 Allgemeines

Die Syntax von Prolog ist recht einfach aufgebaut lässt sich aber trotzdem auch für Komplexe aufgaben einsetzen. Es muss angemerkt werden, das mehrere Prolog Dialekte existieren. In diesem Artikel werde ich den Edinburgh Syntax behandeln. Im folgenden sehen Sie den Unterschied zwischen dem Edinburgh Syntax (E) und dem Marseilles Syntax (M) behandelt (Übernommen aus [2]).

| | | | | | |
|-------------|-----|-----------|------------------|--------|-----------------|
| Variablen: | (M) | x | a | x' | x1 ^a |
| | (E) | X | A | Xtreme | X1 ^b |
| Konstanten: | (M) | 123 | abc ^c | | |
| | (E) | 123 | abc | | |
| Regeln: | (M) | a | -> | b c; | a ->; |
| | (E) | a | :- | b, c. | a. |
| Listen: | (M) | a.b.x.nul | x.y | | |
| | (E) | [a,b,X] | | [X Y] | |

^a (kleiner Buchstabe gefolgt von einer zahl)

^b (beginnt mit großem Anfangsbuchstabe)

^c (ganze Zahlen oder mehr als zwei Buchstaben)

2.2 Syntax

Prolog liest sich mit Daten ein, die dann in der Datenbank abgelegt werden(=Datenbasis). Diese Datenbasis umfasst Konstanten, Variablen und Regeln. Das lässt sich am Besten an Beispielen darstellen:

1. a:- b, c, d.
2. a:- e, f.
3. b:- f.
4. e.
5. f.
6. a:- f.

Zeilen eins, zwei, drei und sechs stellt eine Regel in Prolog dar. Wohingegen Zeile vier und fünf Konstanten sind. Mann Bezeichnet a in Zeile eins als <head> und b, c, b als <tail>. Der Beistrich zwischen den einzelnen Elementen des <tail> steht für ein Logisches und. Der Punkt am ende jeder Zeile schließt eine Eingabe ab. Der Strichpunkt würde für ein logisches oder Stehen, sollte aber nach Möglichkeit vermieden werden. Solche Verknüpfungen sollten wenn es der darzustellen Logik entspricht besser durch zwei zusätzliche Regeln Dargestellt werden. Das obige Beispiel würde dem einfachen Prolog Modell in Abbildung 1 entsprechen.

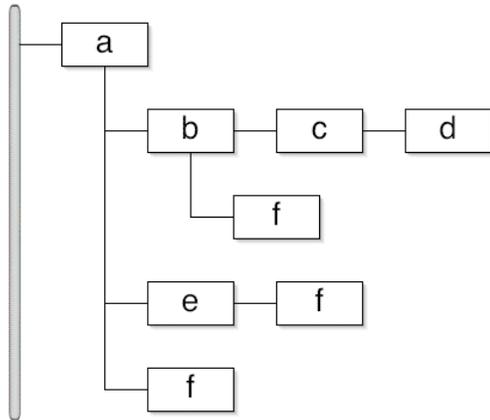


Abbildung 1

2.3 Semantik

Es gibt nach [2] drei Möglichkeiten Semantik von Prolog zu Interpretieren:

1. Logisch
2. Als Ziel das erreicht werden Muss
3. Als context-free grammar

ad 1: Bei der Betrachtung als Logische Aussage (nach der Boole'schen Algebra) wird die Regel erfüllt wenn alle Unterbringungen erfüllt werden. Also im konkreten für Zeile 1: a ist wahr wenn $b \wedge c \wedge d$ wahr sind. Wobei Konstanten immer wahr sind (sofern sie in der Datenbank vorhanden sind). Also heißt d in Zeile 4: d ist wahr.

ad 2: Bei dieser Betrachtung wird das Ziel <head> erfüllt wenn alle Ziele im <tail> erfüllt wurden. Abgebildet auf das obige Beispiel a kann erfüllt werden wenn b, c und d erfüllt werden.

ad 3: Bei der Betrachtung als context-free grammar kann das in <head> geschriebene erreicht werden wenn die in <tail> befindlichen Elemente solange ersetzt werden können, bis <tail> leer ist. Wobei Elemente in <tail> die auf Fakten verweisen herausgenommen werden können.

Konstanten können Fakten zugeordnet werden. Dies geschieht durch eine Verschachtelung. Dadurch lassen sich auch Komplexere Zusammenhänge recht einfach darstellen.

```

a (b) .
a (c) .
  
```

Zum besseren Verständnis soll das Beispiel auf der nächsten Seite dienen.

2.4 Beispiel¹

```
mann(adam).
mann(tobias).
mann(frank).
frau(eva).
frau(daniela).
frau(ulrike).
vater(adam,tobias).
vater(tobias,frank).
vater(tobias,ulrike).
mutter(eva,tobias).
mutter(daniela,frank).
mutter(daniela,ulrike).
```

Die erste Zeile liest sich dabei als „Adam ist ein Mann“. Die dritte Zeile als „Eva ist eine Frau“. Die siebende Zeile als „Adam ist Vater von Tobias“. Da die Datenbank von Prolog jetzt gefüllt ist, lassen sich nun Fragen an Prolog stellen. Fragen beginnen in Prolog immer mit einem ?-. Mann könnte also folgende Fragen stellen:

```
?- mann(tobias). /* Ist tobias ein mann? */
yes.
?- mann(heinrich). /* heinrich ist eine Konstante */
no.
?- frau(X). /* X ist eine Variable*/
X=eva ;
X=daniela ;
X=ulrike ;
no. /*keine weiteren Antworten*/
```

Alles zwischen /* und */ wird als Kommentar angesehen und nicht in die Datenbank eingelesen. Das „X“ steht hier für eine Variable. Prolog versucht jetzt alle Werte für „X“ zu finden für welche die Bedingung „X ist eine Frau“ erfüllt ist.

3 Besonderheiten und Anwendungsgebiete

Die wohl auffallendste Besonderheit ist, dass Prolog eine deklarative Programmiersprache ist. Das bedeutet, dass der Lösungsweg nicht wie bei imperativen Programmiersprachen (C++, Java) genau vorgegeben werden muss. Sondern man gibt Prolog einfach Regeln und Daten ein. Danach lässt sich eine Frage stellen, die Prolog dann mit Hilfe von automatischer Deduktion innerhalb der Regelsätze löst. Dadurch kann dem Programmierer sehr viel Arbeit abgenommen werden, wenn dieser das auch zulässt.

¹ http://de.wikipedia.org/wiki/Prolog_%28Programmiersprache%29

Die Anwendungsgebiete von Prolog sind vielseitig und haben sich im Laufe der Zeit verändert. Zu den wichtigsten Einsatzgebieten von Prolog zählen der Einsatz in Expertensystemen, in der Künstlichen Intelligenz Forschung und der Einsatz als eine der Hauptprogrammiersprachen im 5 Generationen Projekt der Japanischen Regierung. Im Laufe der Zeit hat Prolog in vielen gebieten an Bedeutung verloren. So hat Prolog nach dem Erscheinen von neuronalen Netzen in der KI Forschung an Bedeutung verloren. Im Bereich Expertensysteme gibt es noch Anwendungen wie die Liste auf http://www.ifcomputer.co.jp/IFProlog/Applications/home_de.html beweist. „Das Fifth Generation Project wird häufig als gescheitert, und Prolog fast genauso häufig als Ursache dafür genannt.“²

Heute wird Prolog noch in den Bereichen Computerlinguistik, Expertensysteme, sowie in der Lehre verwendet.

4 Zusammenfassung: Vor- und Nachteile

Cohen, J [2] führt die folgenden 5 direkten Vorteile von Prolog auf.

1. Prozeduren können Parameter enthalten, die sowohl der Ein- als auch Ausgabe dienen.
2. Prozeduren können auch ungebundene Variablen zurückliefern.
3. Prolog beinhaltet automatisches zurückverfolgen (backtracking), das bedeutet es findet alle verfügbaren Lösungen, und nicht nur die Erstbeste.
4. Generelle Mustererkennung in Zusammenarbeit mit einem Zielsuch-Mechanismus.
5. Das Programm und die Daten werden in ähnlicher Weise dargestellt.

Ein weiterer Vorteil von Prolog ist der entscheidend kürzere Quelltext der um den Faktor von 5 –10 mal kürzer ist als in herkömmlichen Programmiersprachen. Dieser tritt aber nur bei Problemen logischer Natur auf, bei klassischen Problemen verschwindet dieser Vorteil.

Zu den wenigen Nachteilen zählt der große Speicherhunger und enorme Rechenzeitverbrauch. Was allerdings in der heutigen Zeit kaum ein Problem darstellen dürfte, da sowohl Speicher als auch Rechenzeit vergleichsweise günstig zu haben sind.

5 Literaturliste

[1] Colmerauer, A. Roussel, P. *The birth of Prolog*, <http://www.lim.univ-mrs.fr/~colmer/ArchivesPublications/HistoireProlog/19novembre92.pdf>, 1992

[2] Cohen, J. *Describing Prolog by its interpretation and compilation*, Communications of the ACM 28(12), 1311 – 1324, 1985

² Schmitt, D. *Zur Entstehung der Programmiersprache Prolog.*, http://www.dietmarschmitt.de/essays/SGI/Prolog/Prolog_Kap_7.html