

Logische Programmierung

Labor 1 : Einführung in Prolog

Inhalt

- Organisatorische Aspekte
- Einführung in Prolog
- Beispiele

Programmierung und Programmiersprachen

Imperative Programmierung

- C, Fortran (prozedural), C++, Java, Python (OO)
- Programm - Abfolge von Anweisungen
- Zuweisungsbefehl, Kontrollstrukturen - für die sequentielle Ausführungssteuerung, Verzweigungen und Zyklen.
- die Rolle des Programmierers: WAS + WIE
 1. beschreiben, WAS berechnet werden soll
 2. die Berechnung zu organisieren
 3. die Speicherverwaltung zu organisieren

Programmierung und Programmiersprachen

Deklarative Programmierung

- basiert auf Ausdrücken
- aussagekräftig, leicht zu verstehen (sie haben eine einfache Basis)
- Programm - Beschreibungen, die Informationen über Werte deklarieren
- auf Anweisungen verzichten
 1. weniger Fehler
 2. werden aus mathematischen Prinzipien generiert (...)
- die Rolle des Programmierers: WAS
- 2 Klasse: funktionale (Lisp, Haskell) und logische (**Prolog**)

Prolog

- 1970 – Marsilia Uni - Werkzeug für die Planung und Lösung von Problemen, die mit symbolischen Darstellungen von Objekten und ihren Beziehungen.
- Anwendungen: relationale Datenbanken, künstliche Intelligenz, Logik, Theorembeweis, Expert-Systeme (+ symbolische Gleichungen ...) u.s.w

Arbeiten in Prolog

Um auf einem Computer mit Prolog zu arbeiten, können wir:

- Direkt am <https://swish.swi-prolog.org/> arbeiten, oder
- Auf <https://www.swi-prolog.org/Download.html> das Programm herunterladen (nehmen Sie Stable release).

Ein Prolog Programm

- Sammlung von Definitionen, die Beziehungen oder Funktionen beschreiben zum Berechnen (symbolische Darstellungen von Objekten und Beziehungen zwischen Objekten)
- die Lösung des Problems - keine schrittweise Ausführung einer Folge von Anweisungen
- unser Programm = Sammlung von logischen Aussagen, z.B., $p, p \rightarrow q, p_1 \wedge p_2 \wedge \dots \wedge p_n \rightarrow q$
- eine nachzuweisende Ziel = $p_1 \wedge p_2 \wedge \dots \wedge p_n$

Prolog-Interpreter

Verwendete Kontrollstruktur:

1. logische Aussagen (d.h. Klauseln)

1.1. Faktes - was als wahr bekannt ist

1.2. Regel - was sich aus den gegebenen Fakten ableiten lässt

2. Ziel – GOAL (zu demonstrieren)

- es beginnt mit der Hypothese (Fakten und festgelegten Regeln)

- Wenden Sie diese Argumentation rückwärts an, um den Ziel zu

beweisen

- Top-Down, Right-Left, Depth-first, Backtracking

Prolog-Interpreter

Wichtig!

- Implikation in Prolog: $q :- p$ (“q ist wahr, wenn p wahr ist”)
- $p_1 \wedge p_2 \wedge \dots \wedge p_n \rightarrow q$ in Prolog: $q :- p_1, p_2, \dots, p_n$ (...)
- Konjunktion in Prolog: $,$
- Disjunktion in Prolog: $;$

Prolog-Interpreter (Logik -> Prolog)

Wie schreibt man diese Reihe in Prolog?

- für jedes x

$$p(x) \wedge q(x) \rightarrow r(x)$$

$$w(x) \vee s(x) \rightarrow p(x)$$

$$t(x) \rightarrow s(x) \wedge q(x)$$

- $t(a)$, $w(b)$

- Ziel: $r(a)$

- Ziel: $p(a)$

Prolog-Programmierung

Wie programmieren wir?

1. Schreiben Sie Fakten über Objekte und ihre Beziehungen.
2. Definieren Sie Regeln über Objekte und ihre Beziehungen.
3. Fragen Sie Anfrage über Objekte und ihre Beziehungen.

Prolog-Kurzgesagt

- **Beschrieben** Sie ihre Situation/ ihre Problem (Objekte und Beziehungen)
z.B.
 - Andrei hat ein Flugzeug.
 - Apfelsaft ist teuer.
- **Stelle eine Anfrage**
- Prolog kann logisch, **neue Fakte** über unsere Situation/unsere Problem **folgern**
- Im Gegenzug gibt Prolog uns **die Abzüge** zurück als Antworten

Bemerkung: **Pradikat = ein Symbol, das eine Beziehung oder ein Objekt beschreibt!**

Prolog Syntax

Wir konstruieren Fakte, Regeln und Anfrage mit der Hilfe von **Terme**.

Terme:

- einfache Terme:
 - Variablen
 - Konstanten: Atome oder Zahlen
- zusammengesetzte (komplexe oder Strukturen) Terme

Prolog Syntax – einfache Terme

- Atome: Zeichenfolgen: Kleinbuchstaben beginnen oder Anführungszeichen

z.B. mia, vincent, liebt, fuss12XTball, "VielGluck!?!", 'Und so weiter'

- Zahlen: ganze Zahlen oder Kommazahlen

z.B. 565, 255, 10, -96.

- Variabeln: Zeichenfolgen: Grossbuchstaben oder Unterstrich beginnen

z.B. X, Y, Z, M, N, P, _etwas, Var, Unbekannt_1, Unbekannt_2

Prolog Syntax – komplexe Terme

- Atome, Zahlen und Variablen sind Bausteine für komplexe Terme
- Ein komplexer Term:

Funktor(Argument1,Argument2,...,ArgumentN)

- Wichtig: der Funktor muss immer ein Atom sein.
- Stelligkeit = die Anzahl der Argumente (“arity” - englisch)

Prolog Syntax – komplexe Terme

z.B.

- playsGuitar(Alexander)
- loves(Vincent,Mia)
- jealous(marsellus,W)
- hide(X,father(father,(father(butch))))

Finden Sie die Stelligkeit für die obengennante Terme!

Bemerkung : Sie können 2 Predikate, die der gleiche Funktor, aber mit verschiedener Stelligkeit haben, definieren.

Aufgabe : Wissenbank 1

- Erstellen Sie die folgende Wissenbank in Prolog:

geist(logan).

geist(hesh).

geist(elias).

spieltFussball(elias).

party.

- Antworten Sie die folgende Fragen mit Hilfe von Prolog:

Aufgabe : Wissenbank 1

- Antworten Sie die folgende Fragen mit Hilfe von Prolog:

1. Ist logan ein geist?
2. Spielt elias Fussball?
3. Spielt hesh Fussball?
4. Ist hesh klein?
5. Schrieb "klein(hesh)" da. Was ist das Ergebnis?
6. Gibt es eine party?
7. Gibt es ein volleyspiel?

Unter Analyse : Wissensbank 2

- Lassen Sie uns die folgende Wissensbank betrachten:

happy(yolanda). %(Fakt)

listensToMusic(mia). %(Fakt)

listensToMusic(yolanda):- happy(yolanda). %(Regel)

playsGuitar(mia):- listensToMusic(mia). %(Regel)

playsGuitar(yolanda):- listensToMusic(yolanda). %(Regel)

Unter Analyse : Wissenbank 2

- Normalerweise, hat eine Regel die Struktur:

RegelKopf :- Regelkorper, und “:-” bedeutet “wenn” oder “folgt aus”

(Vorlesung : “**Regelkopf is wahr, wenn Regelkorper wahr ist**”)

- Schrieb “playsGuitar(yolanda)”. Wahr oder Falsch?

Unter Analyse : Wissenbank 2

- Wissenbank 2 hat 5 Klauseln: 2 **Fakte** und 3 **Regeln**.
- Beenden Sie eine Klausel mit Punkt “.”.
- Wissen bank 2 hat 3 **Predikate**. Welche?

Variablen in Prolog

Schrieb die folgende Wissenbank:

frau(mia).

frau(jody).

frau(yolanda).

liebt(vincent,mia).

liebt(marsellus,mia).

liebt(pumpkin,honey_bunny).

liebt(honey_bunny,pumpkin).

Variablen in Prolog

1. Was macht "frau(X)"?
2. Was macht "liebt(marsellus,X), frau(X)"?
3. Was macht "liebt(pumpkin,X), frau(X)"?

Variablen in Prolog

Schrieb die extra Regel:

```
neidisch(X,Y):- liebt(X,Z), liebt(Y,Z).
```

Schrieb auch in der Kommandozeile:

```
neidisch(marsellus,W).
```

Was bemerken Sie?

Mehr Infos

- <https://www.youtube.com/watch?v=SykxWpFwMGs>