

*Einführung in das Programmieren – Prolog
Sommersemester 2006*

Teil 2: Arithmetik

Version 1.0

Gliederung der LV

Teil 1: Ein motivierendes Beispiel

Teil 2: Einführung und Grundkonzepte

- Syntax, Regeln, Unifikation, Abarbeitung

Teil 3: Arithmetik

Teil 4: Rekursion und Listen

Teil 5: Programmfluß

- Negation, Cut

Teil 6: Verschiedenes

- Ein-/Ausgabe, Programmierstil

Teil 7: Wissensbasis

- Löschen und Hinzufügen von Klauseln

Teil 8: Fortgeschrittene Techniken

- Metainterpreter, iterative Deepening, PTTP, Differenzlisten, doppelt verkettete Listen

Arithmetische Ausdrücke

- Prolog besitzt eine Reihe von vordefinierten arithmetischen Ausdrücken, Funktionen und Operatoren

- Ausdrücke wie $3 + 5$ sind gültige Prologterme

?- $3 + 4 = 7$.

No

- Problem: “=” ist Operator für Unifikation.

- Zur Auswertung arithmetischer Ausdrücke gibt es den **is**- Operator:

-Variable is +Ausdruck

Setzt Variable auf das Ergebnis der Auswertung des Ausdrucks

Es dürfen keine freien Variablen in der rechten Seite vorhanden sein!

?- $X \text{ is } 3 + 5$.

$X = 8$

Yes

Funktionen, Operatoren etc.

+	Addition	-	Subtraktion
*	Multiplikation	/	Division
<	kleiner	>	größer
=<	kleiner gleich	>=	größer gleich
==:=	gleich	= / =	ungleich

- Verschiedene eingebaute Funktionen sind verfügbar **mod**, **exp**, **sqrt**, **sin**, ..
- Unterschied zwischen Unifikation und arithmetischen Berechnungen ist wichtig!
- Manche Operatoren können auch in Infix/Postfixschreibweise geschrieben werden

?- 3 + 4 == +(3, 4) .

Yes

- Man kann auch selbst festlegen, welche Operatoren Infix/Postfix sein sollen...

Zahlentypen

2 Typen von Zahlen: Integer und Floats

Built-In-Prädikate:

number

integer, float

round, truncate

float_fractional_part, float_integer_part

succ

random

Operatoren

- Operatoren können alternative Schreibweisen für Terme einführen
- Wichtiges Anwendungsgebiet sind arithmetische Operatoren
- Beispiele
 - $X * Y$ repräsentiert “Prolog”-Term $* (X, Y)$
 - $N \text{ is } M + 1$ repräsentiert $\text{is} (N, + (M, 1))$
- Beide Schreibweisen sind identisch
- Im Beispiel werden Funktoren $* / 2$ und $+ / 2$ zu Operatoren (Infix-Operatoren)

Operatoren

Bindungsstärke der Operatoren

- Einige Operatoren binden stärker als andere
- In der Mathematik z.B. bindet $*$ stärker als $+$.
- Wichtig: Präzedenz (Bindungsstärke eines Operators)
- Präzedenz wird definiert über Zahlenwerte (von 0 bis 1200)
- Je niedriger Präzedenz, desto stärker bindet der Operator
Beispiel: $*$ hat Präzedenz 400, $+$ hat Präzedenz 500
 $X * Y + Z$ repräsentiert als $+(* (X, Y) , Z)$
- Präzedenz eines Terms ist Präzedenz des Hauptoperators.
- Atome, Zahlen, Variablen, Strukturen und Ausdrücke in Klammern haben Präzedenz 0.

Typen von Operatoren

- Infix-Operatoren, z.B. $+$ in Prolog
- Präfix-Operatoren, z.B. \neg in der Logik
- Postfix-Operatoren, z.B. $!$ in der Mathematik
- Präfix-Operatoren und Postfix-Operatoren sind einstellige Operatoren

Assoziativität von Operatoren

- Assoziativität eines Operators ist zu spezifizieren
Beispiel: $-$ ist linksassoziativ, d.h. $25 - 10 - 3$ ist $(25 - 10) - 3$
- In Prolog repräsentiert mittels den Atomen xf , yf , xfx , xfy , yfx , yfy , fy und fx .
- f bezieht sich auf die Position des Operators
- x und y geben Hinweis auf Position der Argumente
- x bedeutet: Operand mit kleinerer Präzedenz als der des Operators
- y bedeutet: Operand mit Präzedenz kleiner oder gleich der des Operators
- Beispiel: $-$ hat Assoziativität yfx

Beispiele für Assoziativitätsschemata

- yfx : Infix, linksassoziativ
Beispiele: $+$, $-$, $*$
- xfx : Infix, nicht assoziativ
Beispiele: $=$, is

Definition von Operatoren

- Definition neuer Operatoren mittels `op/3` Prädikat
- Definition wird wirksam gemacht, wenn als Anfrage gestellt

```
?- op(400, xfx, ist_groesser_als).
```

```
Yes
```

- Mittels neuer Definition kann Anfrage auch so gestellt werden

```
elefant ist_groesser_als hund.
```

```
Yes
```

- Beispiel Flugdaten

```
ib4567 :: fra/10:55 - madrid/13:20
```

```
:: (ib4567, - (/ (fra, : (9, 30)), / (madrid, : (11, 15))))
```

Prolog-Prädikat zum Auslesen vorhandener Operatoren: `current_op`