Arithmetik

Übersicht

- Arithmetische Operatoren
- `is/2`
- Arithmetische Vergleichsprädikate
- Beispiel: Länge einer Liste

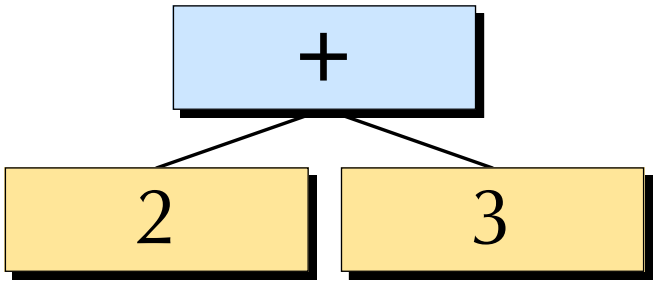
Ziel

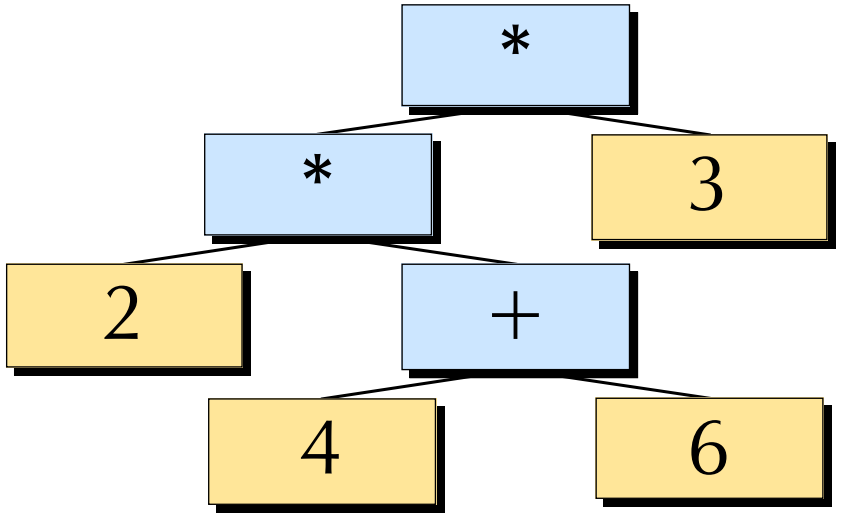
- Rechnen mit Prolog
- Die Länge einer Liste bestimmen können

Arithmetische Operatoren

Anders als bei den meisten Programmiersprachen werden Ausdrücke mit arithmetischen Operatoren nicht *sofort* evaluiert (berechnet).

- Ein Ausdruck wie 2 + 3 oder 2 * (4 + 6) * 3 wird somit nicht ausgerechnet, sondern ist ein gewöhnlicher komplexer Term.





Evaluation arithmetischer Ausdrücke

Zum Berechnen (evaluieren) eines arithmetischen Ausdrucks dient der Infix-Operator is/2, welcher:

- eine Variable und einen Term als Argumente nimmt
- den Wert des Terms berechnet (= »den Term evaluiert«)
- die Variable an den berechneten Wert bindet

Wenn das linke Argument keine Variable ist: Vergleich

```
?- X is 2 * (4 + 6) * 3.
X = 60
```

```
?- 10 is 2 + 8.
yes
```

```
?- 10 is 1 + 1.
no
```

Einige Arithmetik-Operatoren

Operator	Bedeutung	Beispiel
N + N	Addition	15 is 10 + 5
N – N	Subtraktion	10 is 15 – 5
N * N	Multiplikation	15 is 3 * 5
N / N	Fliesspunkt-Division	6.5 is 13 / 2
I // I	Ganzzahl-Division	6 is 13 // 2
abs(N)	Absolutwert	3 is abs(–3)
round(N)	Runden	4 is round(3.7)

Arithmetische Vergleichsprädikate

Die arithmetischen Vergleichsprädikate evaluieren ihre Argumente ebenfalls:

```
zwischen(Mitte, Unten, Oben) :-
  Mitte =< Oben,
  Mitte >= Unten.
```

```
?- zwischen(2, 1, 3*4).
yes
```

Keines der Argumente darf eine freie Variable sein.

```
?- X < 3.
{INSTANTIATION ERROR}
```

Arithmetische Vergleichsprädikate

Operator	Bedeutung	Beispiel
<	kleiner als	2 + 3 < 9 * 9
>	grösser als	170 > 5 * 5
=<	kleiner oder gleich	10 =< 17
>=	grösser oder gleich	8 + 10 >= 18
=:=	gleich	2 * 3 =:= 5 + 1
=\=	ungleich	17 =\= 10

Beispiel: Länge einer Liste

Schreibe ein rekursives Prädikat laenge/2, das die Länge einer Liste bestimmt.

- Erstes Argument: Liste, deren Länge zu bestimmen ist
- Zweites Argument: Die Länge (d.h. das Ergebnis)

Wie immer bei rekursiven Prädikaten unterscheiden wir:

- Abbruchbedingung
- Rekursiver Fall

Beispiel: Länge einer Liste

Abbruchbedingung

- Die Länge der leeren Liste ist 0.
- `laenge([], 0).`

Rekursiver Fall

- Die Länge einer nicht-leeren Liste ist die Länge ihres Rests plus 1.
- ```
  laenge([_|Rest], Ergebnis) :-
    laenge(Rest, RestLanege),
    Ergebnis is RestLaenge + 1.
  ```
  ```