symbolische Zahlendarstellung

```
5 : s(s(s(s(s(o)))))
```

Addition:

```
add(o,X,X).
add(s(X),Y,s(XRes)) :- add(X,Y,XRes).
```

Anfrage:

```
?- add(s(s(o)),s(o),Z).
     Z = s(s(s(o)))
```

kann auch für Subtraktion verwendet werden:

Anfrage:

```
?- add(s(s(o)),Z,s(s(s(o)))).
     Z = s(o)
```

Arithmetikoperationen

binäre Infixoperationen:

`+`	Addition
`-`	Subtraktion
`*`	Multiplikation
`/`	Division
`//`	ganzzahlige Division
`mod`	ganzzahliger Rest

einstellige Präfixoperationen:

`-`	negiert Zahl
`integer`	ganzzahliger Anteil
`float`	Gleitkommazahl

Präzedenzregeln wie üblich

arithmetische Vergleichsoperatoren

`=:=`	Test auf Gleichheit
`=\=`	Ungleichheit
`<`	kleiner
`>`	größer
`=<`	kleiner gleich
`>=`	größer gleich

Beachte Unterschied zu Unifikationsoperator `=`

arithmetische Addition

Programm:

```
addari(X,Y,XRes) :- XRes is X + Y .
```

Anfragen:

```
?- addari(5,3,Z).
```

```
-> Z = 8
```

```
?- addari(5,Z,8).
```

```
-> Fehlermeldung
```

keine Invertibilität, nur für Addition einsetzbar!

Standard-Listennotation

- **bisher: Listen dargestellt durch**
  - Konstante `nil` — leere Liste
  - 2-stelliger Funktor `list(X,Xs)` — `X:` Eintrag, `Xs:` Restliste
- **Prolog liefert Standardnotation**
  - `[]` — leere Liste
  - `[X|Xs]` — `X:` Eintrag, `Xs:` Restliste
- **außerdem:**
  - Aufzählung: `[1,2,3]`
  - Kombination: `[1,2 | [3,4 | []]]`
- **beide Notationen haben nichts miteinander zu tun!!!**

Klausel-/Regelreihenfolge

```
istdrin(X,[X|Xs]).
istdrin(X,[Y|Xs]) :- istdrin(X,Xs).


?- istdrin(Z,[a,b,c]).
```

Lösungen, wenn Reihenfolge der Regeln

wie oben:

```
Z=a;
Z=b;
Z=c;
No
```

vertauscht:

```
Z=c;
Z=b;
Z=a;
No
```

Termination

```
istdrin(X,[X|Xs]).
istdrin(X,[Y|Xs]) :- istdrin(X,Xs).

?- istdrin(a,Zs).
```

Lösungen, wenn Reihenfolge der Regeln

wie oben:

```
Zs=[a|Z1s];
Zs=[Z2,a|Z1s];
Zs=[Z2,Z3,a|Z1s];
```

vertauscht:

```
Stack
overflow
```

unendliche Rechnung

Prädikatenreihenfolge

```
vater(terach,abraham).
vater(terach,nachor).
vater(terach,haran).
vater(abraham,isaak).
vater(haran,lot).
vater(haran,milcah).
vater(haran,yiscah).

männlich(terach).
männlich(abraham).
männlich(nachor).
männlich(haran).
männlich(isaak).
männlich(lot).

%alternative Regeln:
sohn(X,Y) :- vater(Y,X),männlich(X).
sohn(X,Y) :- männlich(X),vater(Y,X).

Anfrage:
?- sohn(Z,abraham).
```

Beispiel zu Cut

```
v :- a,c.
v :- a.
v :- c.
%Hier Aufruf von f, der scheitert:
a :- t,f(o).
a.
c.
t.
t :- w.
%Hier Endlosschleife:
w :- w.
f(s(X)).
```

Beispiel zu Cut

```
v :- a,c.
v :- a.
v :- c.
%Hier Aufruf von f, der scheitert:
a :- t,!,f(o).
a.
c.
t.
t :- w.
%Hier Endlosschleife:
w :- w.
f(s(X)).
```