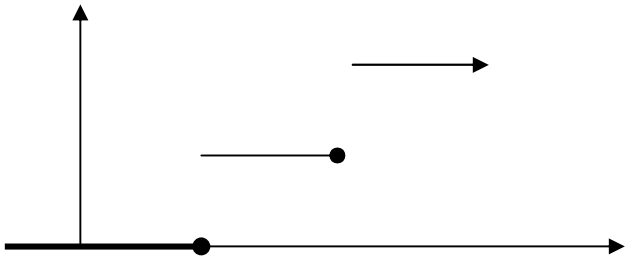
Übungsblock Prolog

Weitere vordefinierte Prädikate:

!	Cut immer beweisbar;
	verhindert Backtracking

Bsp: Treppenfunktion

```
f(X,0) :- X=<3.
f(X,2) :- 3<X,X=<6.
f(X,4) :- 6<X.
```



```
?- f(1,Y),2<Y.          %liefert fail
```

aber Ablauf ineffizient: alle 3 Alternativen werden getestet

hier besser: keine Alternativensuche, wenn eine Klausel greift

Cut einsetzen

Übungsblock Prolog

```
f(X,0) :- X=<3,!.
f(X,2) :- 3<X,X=<6,!.
f(X,4) :- 6<X.

?- f(1,Y),2<Y.              %liefert fail
```

aber nur erste Alternative wird durchlaufen
(Cut zur besseren Effizienz: „grüner“ Cut)

Übungsblock Prolog

```
f(X,0) :- X=<3,!.
f(X,2) :- 3<X,X=<6,!.
f(X,4) :- 6<X.

?- f(7,Y).              /*testet in 1. Regel 7 <=3
                          testet in 2. Regel 3 < 7; überflüssig, da
                          dies durch fail in 1. Regel ohnehin gilt */
```

Klauselvariante mit besserer Performance:

```
f(X,0) :- X=<3,!.
f(X,2) :- X=<6,!.
f(X,4) :-.
```

hier Cut sog. **„roter“ Cut**, da unbedingt nötig!

```
?- f(7,Y).              % liefert 4
?- f(1,Y).   /* würde bei fehlenden Cut Y=0;Y=2;Y=4;
                liefern, aber nur Y=0 sinnvoll */
```

Übungsblock Prolog

Wirkung vom Cut allgemein:

p(…) :- q,!,r.

soll p mittels dieser Klausel bewiesen werden und wird dabei q bewiesen, so wird kein alternativer Beweis für **q** gesucht und p ist nur beweisbar, falls r bewiesen werden kann oder falls es weitere p(..)-Regel gibt, die beweisbar ist

Wie kann mit Cut bedingte Aussage

p:- if q then r else s

formuliert werden?

p:-q,!,r.

p:-s.

Übungsblock Prolog

Arithmetik in Prolog:

?- Z is 3*4.
is vordefiniertes Prädikat für arithm. Berechnungen

rechter Term: beliebiger arithmetischer Ausdruck mit vordefinierten Operatoren, Zahlen oder Variablen, die mit anderen arithmetischen Ausdrücken instanziiert sind:

?- X is 9-5, Y is 20 mod 6, Z is X*Y.
X = 4
Y =2
Z = 8 ;
No

linker Term: Zahl oder Variable

Übungsblock Prolog

vordefinierte Operatoren:

+,- *,/,mod, -(unär), **(Potenz), // (Int.-Division)

vordefinierte Vergleiche:

<,>, =:= (oder ==), =\= (oder \=), =<, >=

Mathem. Funktionen:

abs, exp, log, sqrt, round, truncate, ceiling, floor, integer

Prädikate mit arithmetischen Ausdrücken:
Bsp: Fakultät

```
fak(0,1).
fak(N,F) :-  N1 is N-1,fak(N1,F1), F is F1*N.
```