Übungsblock Prolog

Abfrage der Wissensbasis:

?- listing.	liefert alle Klauseln der Wissensbasis
?- listing(P)	liefert alle Klauseln, die Prädikat P definieren

```
?- listing(vater).
vater(karl,peter).
vater(karl,sabine).
vater(peter,doris).
…
```

Erweitern der Wissensbasis um neue Funktoren:

am Anfang:	asserta(klausel).
am Ende:	assertz(klausel).

Übungsblock Prolog

Löschen aus der dyn. Wissensbasis:

retract(K)	löscht erste Klausel, die mit K unifizierbar ist
Bsp:	?- retract(weiblich(X)). löscht erste Klausel weiblich(name)
abolish(N,S)	löscht alle Klauseln mit Prädikat N, die S-stellig sind
Bsp:	?- abolish(weiblich,1). löscht alle Klauseln weiblich(name)

Übungsblock Prolog

Weitere vordefinierte Prädikate:

true	immer genau einmal beweisbar
fail	nie beweisbar (negative Aussagen formulieren)
not X	beweisbar, wenn Argument X nicht beweisbar ist
X,Y	beweisbar, wenn X <u>und</u> Y beweisbar sind; kann in Anfrage oder rechter Regelseite stehen
X;Y	beweisbar, wenn X <u>oder</u> Y beweisbar sind; kann in Anfrage oder rechter Regelseite stehen

Übungsblock Prolog

Weitere vordefinierte Prädikate:

repeat — immer (unendlich oft) beweisbar
damit Schleifen formulierbar:
beweise q, solange b nicht erfüllt ist

?- p:-….,repeat,q,b,…

repeat beweisbar, beweise q, suche Beweis für b; wird b nicht erfüllt, suche Alternative für q, existieren keine Alternativen mehr, suche neuen Beweis für repeat; dieser existiert immer, also wird q neu bewiesen, ebenso b