Prolog für Schüler

Tilman Brock

8. April 2002

Inhaltsverzeichnis

I. Theorie **3**

II. Praxis **4**

1. Stack-Strukturen **5**
1.1. Lagerverwaltung 5
1.1.1. Fakten 5
1.1.2. Regeln 6
1.2. Andere Anwendungen 7

2. Ringstrukturen **8**

3. Baumstrukturen **9**

4. Datenbank-Strukturen **10**

Teil I.

Theorie

Teil II.

Praxis

1. Stack-Strukturen

1.1. Lagerverwaltung

Mit Prolog lassen sich unter anderem sogenannte Stack-Strukturen gut beschreiben. Eine Stack-Struktur ist hierbei ein Etwas, das gestapelt ist. Man könnte also ein Lagerverwaltungssystem als Stack auffassen, denn in einem Lager gibt es (sehr vereinfacht gesehen) auch viele unterschiedliche Stäpel, auf denen Dinge liegen. Bildlich gesehen sähe das in Etwa so aus wie hier:

seife		
zigaretten	kuchen	
cola	mehl	
schnaps	fisch	
wein	fleisch	hosen
bier	saft	computer
a	**b**	**c**

1.1.1. Fakten

Die gezeigte Lagerstruktur liesse sich etwa wie folgt in Prolog abbilden:

```
stapel(a).       /* "a" ist ein Stapel */
auf(a,bier).     /* Auf "a" ist "bier" */
auf(bier, wein). /* Auf "bier" ist "wein" */
auf(wein, schnaps).
auf(schnaps, cola).
auf(cola, zigaretten).
auf(zigaretten, seife).

stapel(b).
auf(b, saft).
auf(saft, fleisch).
auf(fleisch, fisch).
auf(fisch, mehl).
auf(mehl, kuchen).

stapel(c).
auf(c, computer).
auf(computer, hosen).
```

Das Prädikat `stapel/1` beschreibt, was ein Stapel ist. Dieses Prädikat wird später bei den Regeln wichtig. Das `auf/2`-Prädikat beschreibt, was auf was liegt. `auf(a, bier)` besagt, dass auf a (der untersten Position von Stapel a) eine Kiste Bier liegt. `auf(bier, wein)` besagt, dass auf dieser

Kiste Bier eine Kiste Wein liegt usw.

1.1.2. Regeln

ueber

Mit dieser Wissensbasis muss man ja auch etwas anfangen können, zum Beispiel wäre es gut zu wissen, ueber welcher Kiste das Mehl liegt. Dazu schreibt man zuerst eine Regel `ueber/2`, die ergibt, ob ein bestimmter Block (z.B. `mehl`) über einer anderen Kiste liegt.

ueber(a,bier)	⇒	yes
ueber(wein, bier)	⇒	no

Gelesen wird diese Produktionsregel als „über a ist bier“ bzw. „über wein ist bier“. Letztere Aussage ist falsch, also soll Prolog `no` zurückgeben. Die Produktionsregel ueber/2 soll nach folgendem Muster arbeiten:
Ein Block liegt garantiert über einem anderen Block, wenn er direkt `auf` dem Anderen liegt.
Wenn der gesuchte Block nicht `auf` dem gegebenen Block liegt, soll der Block über dem gegebenen Block ermittelt werden, für den dann wieder getestet wird, ob der gesuchte Block daraufliegt.

```
/* Abbruchbedingung */
ueber(Block1, Block2) :-
  auf(Block1, Block2).

/* Rekursionsblock */
ueber(Block1, Block2) :-
  auf(Block1, BlockX),
  ueber(BlockX, Block2).
```

wo_ist

Die wahrscheinlich wichtigste Funktion in einem Lager ist die Beantwortung der Frage „Auf welchen verdammten Stapel habe ich das Bier gelegt?“. Mit dem Prädikat `wo-ist/2` ist die Antwort leicht zu finden und dem Besäufnis steht nichts mehr im Weg ;-) (ausser natürlich den Kisten, die über dem Bier liegen).

```
wo_ist(Was, Stapel)
```

Das Prädikat wo_ist soll für jeden in der Wissensbasis (siehe S. 5) vorhandenen Stapel prüfen, ob das erste Argument (Was) darauf liegt.

```
wo_ist(Was, Stapel) :-
  stapel(Stapel),
  ueber(Stapel, Was).
```

Zuerst wird die Variable `Stapel` mit einem Wert belegt (unifiziert), und zwar mit dem ersten Wert, der in einem `stapel`-Prädikat in der Wissensbasis auftaucht. Dann wird mit Hilfe des `ueber`-Prädikates bestimmt, ob das gesuchte `Was` auf dem betreffenden Stapel liegt. Gibt das `ueber` „yes“ zurück, so wird der Wert von `Stapel` zurückgegeben.
Die Eigenschaft des beschriebenen Lagers dass jeder Stapel mit einem Block mit dem Stapelbezeichner anfängt, macht dieses Prädikat so leicht zu Programmieren.

1.2. Andere Anwendungen

Die gezeigte Struktur lässt sich ohne großartige Anpassung auf viele andere Bereiche übertragen, zum Beispiel könnte man eine Hochhaus-Siedlung auch als Ansammlung von Stapeln von Wohnungen begreifen. Dazu könnte man als Ersatz für das Prädikat `stapel/1` aus Kapitel 1.1.1 ein Prädikat `haus/1` definieren.

2. Ringstrukturen

3. Baumstrukturen

4. Datenbank-Strukturen