Universität Dortmund
Fachbereich Informatik
Ralf Klinkenberg
SoSe 2002

Dortmund, 24.07.2002

Übungen zum Programmierkurs Prolog

Blatt 3

Abgabe bis zum jeweils nächsten Tag zu Beginn der Übung per EMail oder Ausdruck

Aufgabe 3.1:

a) Erstellen Sie eine Datenbasis mit Fakten zu folgender Relation:

```
vorlesung(Vorlesung,zeit(Tag,Von,Bis),Prof,ort(Gebaeude,Raum))
```

Verwenden Sie als Wertebereich für die `Von`- und `Bis`-Variablen die natürlichen Zahlen.

b) Geben Sie Prolog-Regeln für folgende Relationen an:

```
dauer(Vorlesung,Laenge)
beschaeftigt(Prof,Tag,Zeit)
belegt(Raum,Tag,Zeit)
konflikt(Zeit,Raum,Vorlesung1,Vorlesung2)
```

Verwenden Sie als Wertebereich für die `Zeit`-Variablen die natürlichen Zahlen, d.h. Instanzen der entsprechenden Prädikate geben Informationen zu Zuständen während einer vollen Stunde an (z.B. kann man eine (Uhr)zeitangabe von 10 als 10:00–10:59 Uhr interpretieren).

Anstelle des Wertebereichs der natürlichen Zahlen können Sie hier auch strukturierte `zeit`-Terme wie in Teilaufgabe a) verwenden und dann auf das Argument `Tag` in den entsprechenden Prädikaten `beschaeftigt` und `belegt` verzichten. Die obige Darstellung ist allerdings vermutlich einfacher zu handhaben, wenn Sie z.B. ein Hilfsprädikat wie `inbetween(Von,Bis,Zeit)` aus der heutigen Vorlesung verwenden (siehe auch die Datei `inbetween.pl` unter Hilfsmittel zu Aufgabenblatt 2 auf der WWW-Seite zur Vorlesung oder im Verzeichnis `/home/pkpro/pkpro000/pub/`).

c) Formulieren Sie folgende Anfragen:

- Welche Kurse finden Montags um 10 Uhr statt?
- Wer hält in Hörsaalgebäude I, Hörsaal 2 Vorlesungen?
- Wann hält Professor Unrat Vorlesungen?

Aufgabe 3.2:

Die *Fibonacci*-Zahlen werden bekanntlich durch die folgende Rekursionsgleichung definiert. Für eine natürliche Zahl n soll $f(n)$ die n-te Fibonacci-Zahl angeben:

$$f(n) = \begin{cases} 1 & : falls \quad n = 0 \\ 1 & : falls \quad n = 1 \\ f(n-1) + f(n-2) & : falls \quad n > 1 \end{cases} \tag{1}$$

a) Implementieren Sie diese Rekursionsgleichung als Prolog-Prädikat.

b) Implementieren Sie eine iterative Lösung mittels Akkumulatoren.

c) Machen Sie Ihre Lösung deterministisch.

Fortsetzung auf der Rückseite

Aufgabe 3.3:

Ändern Sie die Prädikate `gcd`, `fact` und `fact2` aus der Vorlesung (siehe Folien oder WWW-Seite zur Vorlesung – Datei `arithmetik.pl` unter Hilfsmittel zum Übungszettel 2 – oder kopieren Sie sich diese Datei aus dem Verzeichnis `/home/pkpro/pkpro000/pub/`) so ab, daß diese deterministisch werden. Dazu müssen Sie gegebenenfalls auch die Hilfsprädikate ändern.