

Programmiersprachen II – Wintersemester 2011/2012
Übungsblatt 4
Besprechung der Aufgaben in der nächsten Übung am 02.02.2012.

1 Abgabe

Das Bearbeiten der Aufgaben ist für die Vorlesung Programmiersprachen 2 *nicht* verpflichtend. Eine freiwillige Abgabe ist möglich unter: John.Witulski@uni-duesseldorf.de. Termin für die Besprechung ist Donnerstag der 02.02.2012 14:30 in 25.12.02.55.

2 Aufgaben

Aufgabe 1 (Einsteins Rätsel)

Betrachten ein Rätsel mit folgenden Einschränkungen:

1. Es gibt fünf Häuser mit je einer anderen Farbe.
2. In jedem Haus wohnt eine Person anderer Nationalität.
3. Jeder Hausbewohner bevorzugt ein bestimmtes Getränk, raucht eine bestimmte Zigarettenmarke und hält ein bestimmtes Haustier.
4. Keine der fünf Personen trinkt das gleiche Getränk, raucht die gleichen Zigaretten oder hält das gleiche Tier wie seine Nachbarn.

Des weiteren erhalten Sie folgende Hinweise:

1. Der Brite lebt im roten Haus.
2. Der Schwede hält einen Hund.
3. Der Däne trinkt gern Tee.
4. Das grüne Haus steht direkt links neben dem weissen Haus.
5. Der Besitzer des grünen Hauses trinkt Kaffee.
6. Die Person, die Pall Mall raucht, hält einen Vogel.
7. Der Mann, der im mittleren Haus wohnt, trinkt Milch.
8. Der Besitzer des gelben Hauses raucht Dunhill.
9. Der Norweger wohnt im ersten Haus.
10. Der Marlboro-Raucher wohnt neben dem, der eine Katze hält.
11. Der Mann, der ein Pferd hält, wohnt neben dem, der Dunhill raucht.
12. Der Winfield-Raucher trinkt gern Bier.

13. Der Norweger wohnt neben dem blauen Haus.
14. Der Deutsche raucht Rothmans.
15. Der Marlboro-Raucher hat einen Nachbarn, der Wasser trinkt.

a) Schreiben Sie ein Programm, das untersucht, welche Person den Fisch als Haustier hat.

Aufgabe 2 (Constraints)

Setzen Sie sich nochmal mit dem Kapitel 10.5 aus dem Skript auseinander, insbesondere mit den Prolog-Builtin `block`. Dieses Builtin ist ein spezielles Feature von SICStus Prolog.

Wichtig: Überprüfen Sie, ob das Prolog-Builtin `term_expansion/2` unter Ihrer Prolog-Distribution verfügbar ist. Falls Sie unbedingt SICStus Prolog benutzen möchten, achten Sie darauf, dass es dort nur `term_expansion/6` gibt. Nutzen Sie in diesem Fall die SICStus-Dokumentation.

a) Machen Sie sich zuerst mit den Prolog-Builtin `term_expansion/2` vertraut: Dieses Prädikat wird automatisch aufgerufen, wenn Sie ein Programm consultieren. Dabei handelt es sich um ein Hook. D.h. eine Schnittstelle, mit der sich ein Programm verändern kann. Als erstes Argument erhält es einen Term. Das zweite Argument ist der neue Term, welcher den ersten überschreibt, sofern das Prädikat erfolgreich ist.

Beispiel (Programmauszug):

```
fib(0,0) :- !.
fib(1,1) :- !.
fib(N, M) :-
    N1 is N - 1, fib(N1, M1),
    N2 is N - 2, fib(N2, M2),
    M is M1+M2.

term_expansion((Head :- Body), (Head :- Body)) :-
    Head =.. [Functor|Args],
    atom_concat('runtimecheck_', Functor, RuntimeFunctor),
    RuntimeHead =.. [RuntimeFunctor|Args],
    RuntimeBody = (statistics(cputime,CpuTime1),
        call(Head),
        statistics(cputime,CpuTime2),
        CpuTime is CpuTime2 - CpuTime1,
        nl,write(time_spent:CpuTime),write(' sec'),nl),
    assert((RuntimeHead :- RuntimeBody)).
```

```
% Console:
?- fib(30,N).
N = 1346269 ?
yes
```

```
?- runtimecheck_fib(30,N).
time_spent:1.26563 sec
N = 1346269 ?
yes
```

b) Versuchen Sie nun, `block` in ihre Prolog Distribution zu implementieren. Gehen Sie davon aus, dass es für ein Prädikat jeweils nur ein `block` geben wird. Nutzen Sie die Co-Routine `when/2` zur Implementierung.

So könnte eine Anwendung aussehen:

```
:- block(fib(-,?)).  
fib(0,0).  
fib(1,1).  
fib(N, M) :- N > 1,  
N1 is N - 1, fib(N1, M1),  
N2 is N - 2, fib(N2, M2),  
M is M1+M2.
```

```
% Console:  
?- fib(N,M), N=10.  
M = 55  
N = 10 ?  
yes
```