

Practical "Introduction to Artificial Intelligence"

Prof. Dr. Gunter Grieser

Block 1: Prolog

Sheet 5: A Closer Look at Terms

Hints:

- *In Block 1 (Prolog) you do not have to submit your solutions to me. Just solve the exercises and discuss your problems and solutions. The aim of Block 1 is that you become familiar with the prolog programming.*
- *If you do not succeed with a task, just delay it and try it again later. Some constructs need time to settle in the brain and will become easier as you get more experienced.*

Preparation (at home):

Read Chapter 9 of LearnPrologNow!.

Exercise 5.1

Reproduce the examples from the chapters of LearnPrologNow! on your machine and solve the exercises.

Exercise 5.2

Define a predicate `count_leafs (Term, N)` that counts the number of atoms (leafs) in a term. For simplicity, first assume that all terms have arity 2 (resp. 0 for the leafs). Then extend it to arbitrary arity.

Exercise 5.3

Define a predicate `is_ground_term (Term)` that is true if Term does not contain any variable.

Exercise 5.4

Define `term_member (Term1, Term2)` that tests whether Term1 is a subterm of Term2.

Exercise 5.5

a) Define a predicate `subterm_left (Term1, Term2)` that deletes from Term1 all subterms but the leftmost one, i.e. `f (g (b, c) , g (d, e) , f)` is transformed into `f (g (b))`.

b) Change it, so that now the rightmost terms survive.

Excercise 5.6

Define a predicate `reverse_left(Term1, Term2)` that reverses the order of the arguments, i.e. `f(g(b,c), g(d,e), f)` is transformed into `f(f, g(e,d), g(c,b))`.

Excercise 5.7

Reconsider Excercise 2.3. Do you have some ideas how the program for that task can be written much more compact using the term and list predicates?

Excercise 5.8

Consider the following predicates `list0`, `list1` and `list2`. What are the differences?

```
list0([]).  
list0(._, Rest) :- list0(Rest).  
  
list1([]).  
list1([_|Rest]) :- list1(Rest).  
  
list2([]).  
List2([_|_]).
```

Excercise 5.9

Create a term as large as possible. How big is the maximal term until the memory is full?

Excercise 5.10

How much is unification with Occurs Check slower than unification without Occurs Check? Create terms of different size and meter the times.

Excercise 5.11

Consider the jug problem: You are at a river and have two empty jugs. One can hold 4 liters, the other 3. Neither jug has any measuring markings on it. How can you get exactly 2 liters?

- a) Define the search problem (see lecture slide 11 of deck 3), i.e. How do you model the problem?
- b) Define a predicate that searches for a solution for the above problem (4 liters / 3 liters / 2 liters).
 - Use the usual backtracking mechanism.
 - Use a depth limit, i.e. only search n steps deep (see lecture slide 74 of deck 3).
 - Your program shall return a sequence of actions that lead the goal.

- c) Extend your program so that it implements iterative deepening (see lecture slide 76ff. of deck 3).
- d) Extend your program so that
 - the target (was: 2 liters) can be given as parameter.
 - also the jug sizes (was: 4 and 3 liters) can be given as parameter.
- e) Define a predicate that solves the jug problem by breath first search.