

Written assignment 1

You should send me by electronic mail (henning@ruc.dk), you answer as *one* pdf or word file **before Monday, September 24, at 12.00**. Explain briefly for each question below, how you solved it, include your source text and a few test queries. (If you can't solve a question, explain then how you tried to solve it and where you think the bug may be). Answers must be individual, i.e., prepared by each student personally, but is a good idea to discuss technical problems with your fellow students.

The purpose of this assignment is to give you some experience with Prolog before we get to the more advanced applications. The following points are exposed in the exercises:

- learning to work with logical variables
- recursion as the only way of doing repetition,
- working with lists and structures,
- pattern matching (unification) instead of selectors and constructors.

NB: It is strongly suggested for *these exercises* that you avoid using standard predicates such as `append` and `member`. Solutions can be written directly in a quite elegant way without using those.

Exercise 1.

The exercise numbered 2.2 in [HC1]: “Logic programming as a framework ...”. A hint for the second question of that exercise: Remember that Prolog predicates (often) are reversible, which means that if you can add, you can also subtract ;-) For this particular exercise, it is acceptable if your implementation says “no” in some cases.

Exercise 2.

The job is to write predicates that can help animal psychologists to analyze the communication between cat and dog. We can assume that their respective vocabulary is defined by the following set of Prolog facts.

```
dogsound(wouf).  
dogsound(wrrouf).  
dogsound(grrrr).  
dogsound(grr).  
dogsound(aoaouuuuuu).  
dogsound(whimp).  
  
catsound(miaw).  
catsound(miiiiiaaaauuuuw).  
catsound(scratch).  
catsound(fshhhhhhhtt).
```

Question 2.1

Write a predicate, `dogtalk` that accepts lists of dog sounds (0, 1, or more) and only that (i.e., including the empty list). Examples:

```
?- dogtalk([wouf,wouf,wouf,grr]).  
yes  
?- dogtalk([wouf,wouf,wouf,grr,miiiiiaauuuuw]).  
no
```

Question 2.2

A dialogue between a dog and a cat can be represented as a list of dog and cat sounds. You should write a predicate, `separate`, which filters out the utterances of the two into separate lists. Example:

```
?- separate([miaw,wouf,wouf,wrrouf,grr,fshhhhhhhtt,  
              scratch,whimp,whimp],Ds,Cs).  
Cs = [miaw,fshhhhhhhtt,scratch],  
Ds = [wouf,wouf,wrrouf,grr,whimp,whimp] ? ;  
no
```

Question 2.3

This question concerns the power game between cats and dogs. You should write a predicate, `winner`, that determines who ended up being the strongest in a dialogue. We say that the dog is the winner if it gets the last word, unless this is `aoaouuuuuu` or `whimp`; otherwise the cat is the winner. Example:

```
?- winner([miaw,wouf,wouf,wrrouf,grr,fshhhhhhhtt,  
              scratch,whimp,whimp],W).  
W = cat ? ;  
no
```

Exercise 3

This is concerned with book, libraries and selection of books. A book is represented as a structure of the form

`book(author, title, no-of-pages, category)`

NB: Do not confuse “book” in this context with a predicate.
The following are examples of structures that represent books:

```
book(negnevitski, artificialIntelligence, 415, study)  
book(dumas, musqueteers, 777, justFun)
```

A library is represented as a list of such book representations; example:

```
[ book(sayers, someTitle, 288, detective),
  book(negnevitski, artificialIntelligence, 415, study),
  book(nerdson, smartAlgorithms, 465, study),
  book(sayers, someOtherTitle, 194, detective),
  book(dumas, musqueteers, 777, justFun)
]
```

For the examples below, we assume that library is given by a fact

```
test_data( ... ).
```

where “...” stands for the structure above.

Question 3.1

Explain in word what the following predicate is doing:

```
p([book(dumas,_,_,_) | _]).  
p([_|Rest]) :- p(Rest).
```

Question 3.2

Write a predicate which, when given a library, returns the list of all titles in the library. Example:

```
?- test_data(Lib), titles(Lib,Titles).  
Lib = ..... ,  
Titles =  
[someTitle,artificialIntelligence,smartAlgorithms,  
 someOtherTitle,musqueteers] ? ;  
no
```

Question 3.3

Write a predicate that extracts the list of all detective stories from a given library.
Example:

```
?- test_data(Lib), detectiveStories(Lib,Ds).  
Ds = [book(sayers,someTitle,288,detective),  
      book(sayers,someOtherTitle,194,detective)],  
Lib = ..... ? ;  
no
```

Question 3.4

Now we are going on vacation and we want to bring some books with us. However, we require that it

- must not include books for study
- must not comprise more than 800 pages.

Define a predicate of the following form:

`vacationReading(library, book-list-for-vacation, total-no-of-pages)`

For given library, the predicate should be able to generate all possible selections of reading for the vacation (as described above) in the form of a list of the selected books. In addition, the predicate indicates the total number of pages in the book selection.