

# Assignment 4

Due: Friday, 3.6.2016, 15:59 via Git

For help, contact [alp-staff@lists.iai.uni-bonn.de](mailto:alp-staff@lists.iai.uni-bonn.de) (staff only) or  
[alp-course@lists.iai.uni-bonn.de](mailto:alp-course@lists.iai.uni-bonn.de) (staff and participants).

Start working on the exercises early enough so that you can contact your tutor in time if you have problems. Don't expect your tutor to be available at midnight or during weekends!

Submit your implemented predicates as a file named "assignment04/solutions.pl" in the Git repository of your group.

Add a file "assignment04/testRuns.txt" showing the console output of a session in which you test that **each** solution works for the provided input data and some queries that represent sensible test cases.

If no input data is provided in the text of the task, create some sensible input data. If input data is represented as facts, include them into the "solutions.pl" file and add suitable comments.

## Task 1. *SLD-Resolution* (11 + 5 Points)

Write down the derivation steps that the following program performs for finding all results for the given goal. Use the notation introduced in the lecture (see slides 4-31, 4-35, 4-37, 4-47, 4-54, 4-56). For each derivation step, show the number of the used clause, the computer unifier and the number of the choicepoint clause. Number each clause relative to the predicate it is part of, not relative to the entire program (e.g. the second clause of predicate b/2 is #2, not #4).

a) ?- a(Result).

a(1).

a(X) :- b(X,Y), c(Y).

b(1, a).

c(b).

b(2, b).

c(c).

b) ?- p([2, 3, 4], Res).

p([X|Y], [A, Z|B]) :- q(X, Z), Y = [A|B].

q(2, 10).

## Task 2. *Classification and Negation (3 Points)*

Assume we have a database of results of tennis games played by members of a club. The results are represented as facts for the predicate `beat/2`, meaning that the player mentioned in the first argument has beaten the player in the second argument:

```
beat( tom, jim ).      % tom has beaten jim
beat( ann, tom ).     % ann has beaten tom
beat( pat, jim ).     % pat has beaten jim
```

Your task is to define a predicate “`category(Player,Category)`” that classifies the players into three categories:

- 1) **winner**: A player who won all his or her games.
- 2) **fighter**: A player who won some games and lost some.
- 3) **loser**: A player who lost all his or her games.

For instance, “?- `category(tom, fighter)`.” should succeed.

## Task 3. *Double negation (2 Points)*

Given the following program discuss and argue whether `r/1` and `s/1` are equivalent or not. Provide a simple definition of `p/2` on which you can demonstrate your arguments.

**Tip:** Consider the possible invocation modes of `r/1` and `s/1`.

```
r(X) :- p(a,X) .
s(X) :- not(not(p(a,X))) .
```