

Assignment 3

Due: Sunday, 01.05.2011, 23:59:59 via SVN

Submit your solution for this assignment into your group's SVN as an **Eclipse Project** named "assignment03".
Programming exercises should be executable **Prolog files** in this project.
Other (theoretical) exercises should be submitted as **PDF files** in your project.

Task 1. *Declarative semantics* (4 Points)

Write down the Herbrand model for the following program:

```
likes(a,b).
likes(c,d).
likes(X,Y) :- likes(Y,X).
```

Task 2. *Operation semantics* (4 Points)

- Which answers will Prolog derive for the goal `likes(X, Y)` and the program from task 1?
- How often will Prolog succeed for the goal `likes(X, Y)`?
- Construct a non-terminating query for the program from task 1. A non-terminating query is a query that does search infinitely for answers without ever returning one.

Task 3. *Resolution of prolog programs* (2 Points)

Consider the following program and write down the resolution steps that lead to the first result for the query

$$?- \text{big}(X), \text{dark}(X).$$

Use the notation introduced in the lecture on slide 2-42: (goal, clause, mgu, resolvent).

```
big(bear).           % #1
big(elephant).      % #2
brown(bear).        % #3
black(cat).         % #4
gray(elephant).     % #5
dark(Z):-           % #6
    black(Z).
dark(Z):-           % #7
    brown(Z).
```

Task 4. *Resolution (3 Points)*

Given the following set of clauses

$$F = \{ [loves(hobbes, calvin)], [loves(hobbes, susi)], [loves(calvin, X) \vee \neg loves(X, susi)] \}$$

Formulate the query “Is there someone calvin loves?” as a first order formula. Deduct with the resolution principle the answer to this query. Which additional information is generated during this process and from which mechanism?

Task 5. *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))).
```

Task 6. *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.