Second Exam
Logic Programming, LVA 703113

March 4, 2016

Name: **Studentnumber**:

The exam consists of 6 exercises with a total of 100 points. Please fill out your name and credentials *before* you start the exam.

1	2	3	4	5	6	Sum

0–49: **5**	50–59: **4**	60–74: **3**	75–89: **2**	90–100: **1**

1. Consider the directed graph $G = (\{a, b, c, d, e, f, g, h\}, E)$ with the following set of edges:

$$E = \{(a, a), (a, b), (b, b), (b, c), (c, d), (d, e)\} .$$

   – Represent G in Prolog and implement a relation *connected*/2 that expresses that two nodes are connected in G. (4 pts)

   – Show that for any directed acyclic graph G the *height* of the SLD tree is bounded by the number of vertices in G. (6 pts)

   – Is the above estimate on the hight of the search tree also true for non-ground queries? (4 pts)

2. Consider the following implementations of *sublist*/2 and *subsequence*/2:

```
sublist(Xs, AsXsBs) :-
        append(_As, Xs, AsXs), append(AsXs, _Bs, AsXsBs).

subsequence([X|Xs],[X|Ys]) :- subsequence(Xs, Ys).
subsequence(Xs,[_Y|Ys]) :- subsequence(Xs, Ys).
subsequence([], _Ys).
```

   – Consider *sublist*/2. Explain why the given goal order is not ideal. (8 pts)

   – Consider the meaning of *sublist*/2 and *subsequence*/2. Is the meaning of these programs the same? Explain your answer. (8 pts)

3. Implement a predicate *isotree*($Tree_1$,$Tree_2$) which holds if $Tree_1$, $Tree_2$ are isomorphic binary trees. (5 pts)

   Hint: You can use any suitable representation of binary trees.

4. Consider the following grammar for propositional formulas over the atoms p, q, and r:

$$\begin{array}{ll} P \to \mathsf{true} \mid \mathsf{false} & P \to \neg P \\ P \to (P \wedge P) & P \to (P \vee P) \end{array}$$

   – Write a DCG that generates the languages by *directly* encoding the grammar and builds an expression tree for the formula parsed. (10 pts)

   – Improve your implementation by taking into account the following precedence of connectives $\neg > \wedge > \vee$, so that brackets can be dropped. Furthermore prevent the left-recursion in the grammar. (15 pts)

5. Implement the predicate *palindrome*(Xs), which holds if Xs represents $w \in \Sigma^*$, such that w is a palindrom.

```
?- palindrome([a,b,b,a]).
true
```

(20 pts)

6. Determine whether the following statements are true or false. Every correct answer is worth 2 points, every wrong answer -1 points. (The worst that can happen is that you get zero points for this exercise.) (20 pts)

statement	**yes**	**no**
An existential fact is a fact that contains existentially quantified variables.	☐	☐
A rule is a universally quantified logical formula of the form $A \leftarrow B_1, B_2, \ldots, B_n$, where A is a goal and for all $i = 1, \ldots, n$: B_i is a goal.	☐	☐
Data is structured in logic programs to obtain for example (i) better modularity or (ii) better organisation of the data.	☐	☐
Almost all, but not all basic elements of a relation database model can be expressed in Prolog.	☐	☐
Consider the standard implementation of *member*/2. Then any call to *member* terminates iff the second argument is a complete list.	☐	☐
A Prolog clause is called *iterative* if it has one recursive call and zero or more calls to system predicates that appear before the recursive call.	☐	☐
A cut fixes all choices between (and including) the moment of matching the rule's head with parent goal and the cut. If backtracking should reaches the cut, then the cut succeeds and the execution is continued with the clause *after* the clause containing the cut.	☐	☐
Constraint logic programs typically fall into two parts. In one part the pure relation is formuated, possible using constraints. In the second part the constraints are instantiated.	☐	☐
The predicate *bagof*($Template$,$Goal$,Bag) unifies Bag with the alternatives of $Goal$ that meet $Template$.	☐	☐
Time complexity of Prolog programs is measured via the number of inferences performed and real time consumption depends on the time used for unification attempts.	☐	☐