Second Exam
Logic Programming, LVA 703113

July 5, 2013

Name: **Studentnumber**:

The exam consists of 6 exercises with a total of 100 points. Please fill our 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 $G = (\{a, b, c, d, e, f, g\}, E)$ with the following set of edges:

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

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

   – Draw the SLD-tree T for the query connected(X,Y). (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)

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. Write a program to normalise products using incomplete datastructures.

```
?- normalise((a*b)*(c*d),N).
N = a* (b* (c* (d*1)))
```

(10 pts)

4. 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
```

(15 pts)

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

$$\begin{array}{ll} P \to \texttt{p} \mid \texttt{q} \mid \texttt{r} & P \to \texttt{\~{}}P \\ P \to (P \texttt{ \& } P) & P \to (P \texttt{ | } P) \\ P \to (P \texttt{ => } P) & \end{array}$$

   – 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 `~` > `&` > `|` > `=>`, so that brackets can be dropped. Furthermore prevent the left-recursion in the grammar. (15 pts)

6. Determine whether the following statements are true or false. Every correct answer is worth 2 points; wrong answers "earn" -1 points. (20 pts)

statement	**yes**	**no**
An existential fact is a fact that contains existentially quantified variables.	☐	☐
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.	☐	☐
?−**op**(180, xfy, [imp, =>]). asserts that the operators imp and => are binary right-associative operators.	☐	☐
The explicit constructor for difference structures should be removed, if time or space efficiency is an issue.	☐	☐
A meta-interpreter for a language is an interpreter for the language written in the language itself.	☐	☐
Prolog is the only language that allows the efficient manipulation of meta-interpreters.	☐	☐