Exam, TDDD08 Logic Programming

2016-10-22, 08:00 – 12:00, R41, U14, U15

Means of assistance (hjälpmedel):
• A sheet of notes – 2 sided A5 or 1 sided A4. The notes should be signed in the same way as the exam sheets and returned together with the exam.
• English dictionary

You may answer in English or Swedish.

Grade limits: 3: 17 p, 4: 23 p, 5: 29 p (for total 35 p).

Remember to give explanations for all answers!
Unexplained answers may be granted 0 points.
For instance, when you write a program you should explain the relations defined by the predicates of the program, and the role of each clause.

GOOD LUCK!

---

1. Determine which of the following pairs of terms are unifiable, and provide a most general unifier (mgu) in case there is one.

   a) $p(X, f(a, V), X)$
   $p(f(Y, b), Z, Z)$

   b) $[Y, a, a]$
   $[a, X|X]$

   c) $p(f(X), Y, f(Y), X)$
   $p(Z, g(Z), W, h(W))$

   d) $p(f(X, a), X, g(a))$
   $p(f(Z, Y), g(b), Z)$

   (4 points)

2. A programmer compiled a Prolog program intended to check whether a difference list is empty:

   $$empty\_dl(L\text{–}L).$$

   and provided a query $empty\_dl(t_1\text{–}t_2)$, where $t_1\text{–}t_2$ is a non-empty difference list. Prolog returned an answer (the query has not failed). Explain what happened; give terms t_1, t_2 giving such behaviour.

   What should be the result for this program and this query? (In other words: What are the SLD-derivations for this program and this query?) (3 points)

3. Write a program finding which elements (at the same positions) of the two lists are equal. Define predicate $ee/3$ which is true when its arguments are three lists, say $[t_1, \ldots, t_n], [s_1, \ldots, s_n], l$ such that the first and the second are of equal length, and the third, l, is the list of those elements t_i for which $t_i = s_i$.

   Employ $ee/3$ to write a predicate which checks that given two lists have exactly three equal elements at the same positions.

   Hint: use Prolog built-in `dif/2` to check inequality of terms. You are not allowed to use other built-in predicates of Prolog. (Thus in particular you cannot use negation.) Remember that s/n means that the symbol s takes n arguments. (4 points)

4. Assume that a logic program is given defining two predicates $c1/1$, $c2/1$. The predicates specify two disjoint sets of constants (for no term t both $c1(t)$ and $c2(t)$ hold). Let us call a constant satisfying $c2$ a *red* constant.

   Define a predicate $t/1$ specifying the terms built out of function symbols $f/1$, $g/2$ and the constants specified by $c1$, $c2$.

   Define a predicate $t1/1$ specifying those terms (from the terms described above) in which a red constant occurs.

   Define a predicate $tex1/1$ specifying those terms (from the terms described above) in which a red constant occurs exactly once.

   The task is to write a definite program defining the required predicates. You are not allowed to use Prolog built-ins, this includes negation. (4 points)

5. Consider the following definite program P:

$$
\begin{array}{l}
p(\,X, g(X)\,) \leftarrow q(\,X, Y\,), p(\,g(X), X\,).\\
q(\,Y, f(Z)\,) \leftarrow r(\,f(Y), Z\,).\\
q(\,V, g(W)\,) \leftarrow q(\,V, W\,).\\
r(\,f(X), g(X)\,).
\end{array}
$$

   Assume that the vocabulary $\mathcal{A}$ contains one constant a and two one-argument function symbols f, g. What is the Herbrand universe $\mathbf{U}_{\mathcal{A}}$ corresponding to $\mathcal{A}$?

   Find the least Herbrand model $\mathbf{M}_P$ of the program. Alternatively, find the set $PTR(P)$ of atomic logical consequences of the program.

   Give one ground atom which is a logical consequence of P, and one which is not; the predicate symbols of the atoms should not be r.

   Give a non-ground atom, with predicate symbol q, that is a logical consequence of P. (5 points)

6. For a chosen query A and a chosen subset $P' \subseteq P$ of the previous program construct two SLD-trees (using different selection rules) – one finite and one infinite. (2 points)

7. Consider a program INS (describing removing/adding an element from/to a list):

$$s(E, [E|L], L).$$
$$s(E, [H|L], [H|M]) \leftarrow s(E, L, M).$$

Is it correct w.r.t. the following specification?

$$S_1 = \left\{ s(t, [t_1, \ldots, t_n], l) \in \mathbf{B}_{\mathcal{A}} \;\middle|\; \begin{array}{l} n > 0,\ t = t_i \text{ for some } 1 \leq i \leq n, \\ l \text{ is the list } [t_1, \ldots, t_n] \text{ with } t_i \text{ removed} \end{array} \right\}$$

Prove that the program is correct w.r.t. a specification

$$S = \left\{ s(w, t, u) \in \mathbf{B}_{\mathcal{A}} \;\middle|\; \text{if } u \text{ is a list then } t \text{ is a list, and } |t| = |u| + 1 \right\},$$

where $|l|$ denotes the length of a list l, e.g. $|[t_1, \ldots, t_n]| = n$. (4 points)

8. Translate the following DCG into a Prolog program (using a standard approach).

```
p(0) --> [].
p(X) --> [a], p(X), [a].
p(s(X)) --> [b], p(X).
```

Show that $[a, b, a]$ is a member of the language of $p(s(0))$ by sketching a proof tree, or a successful SLD-derivation.

For which terms u the language of $p(u)$ is not empty? (3 points)

9. Consider the following general program P:

$$p(a).$$
$$p(f(f(X))) \leftarrow \neg p(X).$$

Draw SLDNF-forests for queries $p(Y)$, $p(f^2(a))$, $p(f^3(a))$. Make it clear, which trees are finitely failed and which leaves are floundered.

Construct the completion $comp(P)$ of the program (except for the equality axioms CET). Explain whether $\neg p(f^2(a))$ is a logical consequence of $comp(P)$.

Hint: It may be useful to remember that from CET it follows that $\forall X\, (a \neq f(X))$, and that $f(X) = f(Y)$ implies $X = Y$. (5 points)

10. Choose one case from the list below, and explain the notion(s).

    Your explanation should be short but precise, and should show that you understand the notions. The chosen notions should not be explained in your sheet of notes.

    (a) Logical consequence (of a set of formulae).

    (b) Herbrand interpretation, Herbrand model.

    (c) SLD-derivation, computed answer.

    (d) Soundness of SLD-resolution.

    (e) Difference list.

    (f) Closed world assumption (CWA).