Exam, TDDD08 Logic Programming

2016-08-19, 14:00 – 18:00, G36

Person on duty (*jour*): Włodek Drabent, by phone

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) $p(X, f(a), Y)$ and $p(Y, f(X), f(Z))$

   b) $p(f(X,Y), Y)$ and $p(Z, f(Z,X))$

   c) $[X, a]$ and $[Y, Z|Y]$

   d) $p(X, f(a), f(Y))$ and $p(Y, f(X), Z)$

   (4 points)

2. A programmer compiled a program

   ```
   p :- q( t1 ).
   q( t2 ).
   ```

   where t_1 and t_2 are two non-unifiable terms (without any common variables). In spite of this the Prolog system answered "yes" for the goal `?-p`. Explain this. Give an example pair of such terms.

   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 logic program defining a predicate $oe/2$ (oe for "odd equal") which is true when its arguments are lists, say $[t_1, \ldots, t_n]$ and $[s_1, \ldots, s_m]$, and $t_{2i-1} = s_{2i-1}$ for any $i > 0$ such that $2i - 1 \leq n$ and $2i - 1 \leq m$ (in other words: the odd numbered elements present in both lists are equal).

   Your program should be a definite clause program without Prolog built-ins. (Thus in particular you cannot use negation.)

   Remember that by a list we mean a term of the form $[t_1, \ldots, t_n]$ (equivalently $[t_1|[t_2|\cdots[t_n|[\,]]\cdots]]$). So for instance $[1, 2|3]$ is not a list.

   (3 points)

4. Consider binary ordered trees, in which each leaf contains a value (and non-leaf nodes do not contain values). The values in the leaves may be arbitrary terms. Design a representation of such trees as terms. Design also a representation of trees which, additionally, contain a value in each non-leaf node.

   Assume that a predicate $\mathtt{lt}/2$ defines an ordering on terms (so $\mathtt{lt}(t_1, t_2)$ means that term t_1 precedes t_2 in the ordering). Write a program transforming a tree of the first kind into a tree of the second kind, in which each non leaf-node N has the smallest value from the leaves below N.

   (4 points)

5. Consider the following definite program P:

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

   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 two examples of ground atoms which are logical consequences of P; the predicate symbols of the atoms should be q and r, and each atom should not be an instance of a unary clause of the program.

   Give an example of a ground atom which is not a logical consequence of P, its predicate symbol should be q or r.

   Give an example of a non-ground atom which (1) is a logical consequence of P, and (2) is not an instance of any unary clause of the program; or explain that such atom does not exist.

   (5 points)

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

7. Consider the well-known program APPEND:

$$a([\,], L, L,).$$
$$a([H|K], L, [H|M]) \leftarrow a(K, L, M).$$

Is APPEND correct w.r.t. the specification

$$S_0 = \{\, a([e_1, \ldots, e_m], [e_{m+1}, \ldots, e_n], [e_1, \ldots, e_n]) \in \mathbf{B}_\mathcal{A} \mid 0 \leq m \leq n \,\} \ ?$$

($\mathbf{B}_\mathcal{A}$ is the Herbrand base.) A brief explanation is sufficient here.

Prove that – in any answer $a(l_1, l_2, l_3)$ of the program – if both l_2, l_3 are lists then l_2 is not longer than l_3. More formally: Prove that the program is correct w.r.t. the specification

$$S_0 = \{\, a(l_1, l_2, l_3) \in \mathbf{B}_\mathcal{A} \mid \text{if } l_2, l_3 \text{ are lists then } |l_2| \leq |l_3| \,\},$$

where $|l|$ is the length of a list l. Use a standard method. (4 points)

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

```
p( n ) --> [].
p( a ) --> [a].
p( b ) --> [b].
p( t(L,E,R) ) --> p( L ), [E], p( R ).
```

Show that $[a, b]$ is a member of the language of $p(t(a, b, n))$ 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:

$$p(0).$$
$$p(s(s(X))) \leftarrow \neg p(s(X)), \neg p(X).$$
$$q \leftarrow p(Y)$$

Draw SLDNF-forests for queries $p(s^3(0))$ and q. (Choose a convenient selection rule.) 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). Explain whether $\neg p(s(0))$ is a logical consequence of $comp(P)$.
(5 points)

10. Choose two cases from the list below, and explain the notions.

    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) Herbrand interpretation.

    (b) Query, general query.

    (c) Proof tree

    (d) Computed answer substitution.

    (e) Specification, and completeness of a program with respect to a specification.

    (f) Incorrectness diagnosis

    (g) Finitely failed SLD-tree.

    (h) General program (called also normal program)

    (i) Constraint predicate (called also interpreted predicate).

    (2 points)