Exam, TDDD08 Logic Programming

2017-10-20, 08:00 – 12:00

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

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

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

   d) $p(X, [Y|Y])$
   $p(f(Y), [\,[\,]\,|Y])$

   (4 points)

2. Write a program defining the following predicates:

   ep/1 describing the lists which contain a pair of two consecutive equal elements.

   noep/1 describing the lists which do not contain a pair of two consecutive equal elements.

   ep1/1 describing the lists which contain exactly one pair of two consecutive equal elements.

   Your program should be a definite clause program. Negation, `( -> ; )`, and other Prolog built-in predicates should not be used, except for `dif/2` for inequality of terms. A list like $[a, a, a]$ should be treated as containing two pairs of consecutive elements.

   (3 points)

3. Consider the terms built out of constants a, b, c and two-argument function symbols f, g. Define a predicate $t/1$ specifying the set of such terms.

   Define a predicate $rep/2$ describing the replacement of each g in such a term by (a two-argument function symbol) h.

   Define a predicate $tr/2$ which translates a term satisfying $t/1$ into the list of symbols representing the term. For instance, the list corresponding to $f(a, b)$ is $[f$, '(', a, ',', b, ')']. (For simplicity, you may assume that the term to be translated does not contain g).

   The task is to write definite clause program defining the predicates. Negation and Prolog built-in predicates should not be used, except for `append`/3.

   For the full score your procedure *tr*/2 should efficiently work as a Prolog program; in particular it should avoid the inefficiency due to multiple traversing the same fragments of lists by `append`/3.

   Hints: Begin with an obvious, inefficient version of *tr*/2. Use difference lists for efficiency.

   (5 points)

4. Consider a program P_1:

$$p(f(Y), Z). \qquad q(g(Y), Z) \leftarrow p(f(Y), Z).$$

   Does there exist a query Q such that for Q and P_1 Prolog provides a wrong result due to omission of the occur-check? Give such a query or explain that it does not exist. (2 points)

5. Consider the following definite program P:

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

   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 an example of a ground atom which is a logical consequence of P, but is not an instance of $r(g(Z), f(Z))$.

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

   Give an example of a non-ground atom which is a logical consequence of P, but is not an instance of $r(g(Z), f(Z))$.

   (5 points)

6. For a chosen query Q 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 P_3 :

$$m(\,X, a\,).$$
$$m(\,f(X,Y), f(X',Y')\,) \leftarrow m(X,X'), m(Y,Y').$$

   (a) Is the program correct with respect to the following specification?

$$S_0 = \left\{\, m(t,u) \in \mathbf{B}_{\mathcal{A}} \;\middle|\; t,u \in T_{fa} \,\right\},$$

   where $\mathbf{B}_{\mathcal{A}}$ is the Herbrand base, and T_{fa} is the set of terms built out of constant a and 2-argument function symbol f.
   A brief explanation is sufficient here.

   (b) Let $|t|$ stands for the number of (occurrences of) constants in a term t, for instance $|a| = 1$, and $f(a,a) = 2$.
   Using a standard method, prove that the program is correct w.r.t. a specification

$$S = \left\{\, m(t,u) \in \mathbf{B}_{\mathcal{A}} \;\middle|\; |t| \geq |u| \,\right\}.$$

(4 points)

8. Consider the DCG:

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

   (a) Translate the DCG into a Prolog program (using a standard approach).
   (b) Show that $[a,b]$ is a member of the language of $p(s(0))$ by sketching a proof tree or a successful SLD-derivation.
   (c) For which ground terms u the language of $p(u)$ is not empty?

(3 points)

9. Consider the following general program P_4:

$$p(f^2(X)) \leftarrow \neg p(f(X)), \neg p(X).$$

We abbreviate $f(f(X))$ by $f^2(X)$, $f(f(f(X)))$ by $f^3(X)$ and so on.

   (a) Draw SLDNF-forests for queries $p(f^3(a))$ and $p(Y)$, under selection rules chosen by you. Make it clear, which trees are finitely failed, which leaves are floundered, which branches are successful derivations, and what are their answers.

(b) Construct the completion $comp(P_4)$ of the program (except for the equality axioms CET). Explain whether $p(f^2(a))$ is a logical consequence of $comp(P_4)$. Does the SLDNF-tree for $p(Y)$ provide a corresponding result?

Hint: Use the fact that $\neg p(a)$ and $\neg p(f(a))$ are logical consequences of $comp(P_4)$; you are not required to show this.

(5 points)

10. Choose two cases from the list below, and explain the notions(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) Most general unifier.
(b) Interpretation, Herbrand interpretation.
(c) Proof tree.
(d) Soundness of SLD-resolution.
(e) Computed answer substitution.
(f) Constraint domain.
(g) Negation as finite failure.
(h) Stratified program.
(i) Nonmonotonic reasoning.

(2 points)