Exam, TDDD08 Logic Programming

2017-01-03, 14:00 – 18:00

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

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

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

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

   (4 points)

2. Provide an example of a query Q and a program $P = \{A\}$ (consisting of a single clause) such that the result of executing P with Q under Prolog is different from what is theoretically described by SLD-resolution. (2 points)

3. Write a definite clause program which compares lists. It should have answers of the form

   $$co([s_1, \ldots, s_m], [t_1, \ldots, t_n], [u_1, \ldots, u_k]),$$

   where $k = \max(m, n)$, and the elements u_i (for $i = 1, \ldots, k$) are

   $$u_i = \begin{cases} e & \text{when } 1 \leq i \leq \min(m, n) \text{ and } s_i = t_i \\ n & \text{when } 1 \leq i \leq \min(m, n) \text{ and } s_i \neq t_i \\ 1 & \text{when } n < i \leq m \\ 2 & \text{when } m < i \leq n \end{cases}$$

(Continued on the next page)

Use Prolog built-in `dif`/2 to check inequality of terms.
Other built-ins and constructs of Prolog, like negation, are forbidden.

Note that $1 \leq i \leq \min(m, n)$ means that both s_i, t_i exist, and $n < i \leq m$ or $m < i \leq n$ means that only one of s_i, t_i exists.

(3 points)

4. (a) Write a program which, given a non-empty list of numbers, finds the greatest element of the list. To compare numbers, you can use built-in predicates `>`/2 and `=<`/2 of Prolog.

   Comments. Other built-ins and constructs of Prolog, like negation, are forbidden. Do not bother about possible run-time errors when `>` or `=<` get a wrong kind of arguments. Do not bother about checking if the input is of required kind (if not, the program should produce no answer, and may result in a run-time error).

   (b) Consider nested lists of numbers. Write a program finding the greatest number in such a nested list. Decide how to deal with the case when there are no numbers (like $[\,[\,],[\,]\,]$) and thus no greatest number. You may use built-in `number`/1 of Prolog, which recognizes that its argument is a number.

   Comments. The conditions from problem 4a apply. A nested list of numbers can be defined recursively as the empty list, or a list with each element being a number or a nested list of numbers.

   You may begin with solving a simpler problem, excluding empty lists (as list elements, and at the top level).

   (5 points)

5. Consider the following definite program P:

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

   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 an example of a ground atom which is a logical consequence of P; the 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.

(Continued on the next page)

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.

(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 PREFIX:

$$p([\,], L).$$
$$p([H|T], [H|L]) \leftarrow p(T, L).$$

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

$$S_1 = \{\, p([t_1, \ldots, t_m], [t_1, \ldots, t_n]) \in \mathbf{B}_{\mathcal{A}} \mid 0 \leq m \leq n \,\}.$$

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

   (b) Using a standard method, prove that the program is correct w.r.t. a specification

$$S = \left\{\, p(u, t) \in \mathbf{B}_{\mathcal{A}} \;\middle|\; \text{if } u, t \text{ are lists then } |u| \leq |t| \,\right\}$$

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

8. Consider the DCG:

```
p(a,[]) --> [a].              p(b,[]) --> [b].
p(a,T) --> [a], p(a,T).       p(b,T) --> [b], p(b,T).
p(a,[c|T]) --> [a], p(b,T).   p(b,[c|T]) --> [b], p(a,T).
```

   (a) Translate the DCG into a Prolog program (using a standard approach). It is sufficient to present only those clauses that are used in the next task.

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

   (c) For which terms t, u is the language of $p(t, u)$ not empty?

(3 points)

9. Consider the following general program P:

$$p(X, Y) \leftarrow m(X, Y).$$
$$p(X, Y) \leftarrow \neg\, m(X, Y).$$
$$m(H, [H|L]).$$
$$m(E, [H|L]) \leftarrow m(E, L).$$

Draw SLDNF-forests for queries $p(a, b)$ and $p(a, Y)$. Make it clear, which trees are finitely failed, which leaves are floundered, which branches are successful derivations, and what are their answers. (If a tree has many answers, make explicit at least two of them.)

Construct the completion $comp(P)$ of the program (except for the equality axioms CET). Explain whether $p(a, Y)$ is a logical consequence of $comp(P)$. Compare this with the corresponding SLDNF-forest.

(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) Definite logic program.

    (b) Interpretation, model.

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

    (d) Incorrectness diagnosis.

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

    (f) Negation as finite failure.

    (2 points)