CSE-505 Computing with Logic

Fall '06

Mid-Term Exam

Duration: 1h 20m | Nov 1, 2006 | Max: 35 points

---

1. [9 points] Consider the following formulas in first-order predicate logic:

$$\begin{aligned}\mathcal{F}_1: &\quad \forall X.p(X,X) \wedge \forall X,Y.(p(X,Y) \Rightarrow p(f(X),Y))\\ \mathcal{F}_2: &\quad \forall X.q(X,a,X) \wedge \forall X,Y,Z.(q(X,Y,Z) \Rightarrow q(f(X),f(Y),Z))\\ \mathcal{F}_\ni: &\quad \forall X,Y,Z.q(X,Y,Z) \Rightarrow p(X,Z).\end{aligned}$$

   (a) [3 points] Construct an interpretation $\Im$ with $|\Im| = \mathbf{N}$ (i.e., the domain of the interpretation is the set of natural numbers) such that $\Im$ is a model for the above $\mathcal{F}_1$. Be sure to convincingly show that your interpretation is indeed a model for the formula.

   (b) [3 points] Construct an interpretation $\Im$ with $|\Im| = \mathbf{N}$ such that $\Im$ is a model for the above $\mathcal{F}_2$. Be sure to convincingly show that your interpretation is indeed a model for the formula.

   (c) [3 points] Is $\mathcal{F}_3$ a logical consequence of the first two formulas? i.e., is $\mathcal{F}_1 \wedge \mathcal{F}_2 \models \mathcal{F}_3$? Explain.

2. [6 points] Let $G = (V, E)$ be an undirected graph with vertices V and edges E. Let e be a binary predicate such that $e(v_1, v_2)$ is true if and only if $(v_1, v_2) \in E$, that is, there is an edge between v_1 and v_2 in E.

   (a) [2 points] Let p be a unary predicate such that $p(v)$ is true if and only if v is a vertex in the graph. Write a Horn Clause formula of the form $\forall X.(\mathcal{F} \rightarrow p(X))$ that relates the two predicates p and e.

   (b) [4 points] A graph is said to be connected if there is a path between every pair of vertices. Write a formula in first order logic such that it is true whenever e that represents a connected graph.

3. [7 points] Consider the following definite logic program:

```
s([], _).
s([X|Xs], [X|Ys]) :- s(Xs, Ys).
```

   (a) [2 points] Show the sequence of resolution steps in *some* successful SLD derivation for the query `s([a,b], Q)`.

   (b) [2 points] Give the computed answer substitutions for *every* successful SLD derivation for the query `s(Q, [c,d])`.

   (c) [3 points] Show the SLD for the query `s(Q,[c,d])` showing all derivations (successful, failed, or infinite). If the tree has infinite paths, indicate these paths and why they are infinite.

4. [8 points]

   (a) [4 points] Consider the following definite logic program:

   ```
   p(X,Y) :- e(X,Y).
   p(X,Y) :- e(X,Z), p(Z,Y).
   e(1,2).
   e(2,3).
   e(3,4).
   e(4,5).
   ```

   How many iterations does it take to compute the least Herbrand model of the above program using the T_P operator? Explain.

   You need not show every step of the computation. You need to only show how many steps it takes.

   (b) [4 points] Consider the following definite logic program:

   ```
   q(X,Y) :- e(X,Y).
   q(X,Y) :- q(X,Z), q(Z,Y).
   e(1,2).
   e(2,3).
   e(3,4).
   e(4,5).
   ```

   How many iterations does it take to compute the least Herbrand model of the above program using the T_P operator? Explain.

   You need not show every step of the computation. You need to only show how many steps it takes.

5. [5 points] Write a predicate `double(L)` that, given a list `L` succeeds if and only if every element in `L` occurs at least twice.

   For instance, `double([])`, `double([a,a,b,c,b,c,b])` and `double([a,a,a])` all succeed, while `double([a,b]` and `double[a,a,b,a,c]` fail.

   You may use any define and use helper predicates to define `double`. You may also use, as a helper predicate, anything that is fully defined in the book or in a standard Prolog library; in that case, be sure to say precisely where I can find a reference to this predicate and what it does.

END OF EXAM