CSE-505 Computing with Logic

November 3, 2008 **Mid-Term Exam** Max: 40 points

Fall 2008 Duration: 1h 25m

---

1. [4 points] Formalize the following sentences of natural language as formulas in predicate logic:

   (a) No man is superior to any other.

   (b) Every student of CS who is not specializing in Logic must take a course in Automata Theory.

   (c) The village barber shaves all men in the village who do not shave themselves.

   (d) Every statement must be followed by a semicolon except if it is the last statement in a block.

2. [6 points] Consider the following definite logic program:

   ```
   p(f(f(f(a)))).
   p(X) :- p(f(X)).
   p(g(X)) :- p(f(X)).
   p(h(X)) :- p(g(f(X))).
   ```

   Using the immediate consequence (T_P) operator, find the least Herbrand model for the above program. Show each step of the iteration sequence used to compute the least model.

3. [8 points] Consider the following definite logic program:

   ```
   r(X,Y) :- a(X,Y).
   r(X,Y) :- r(X,Z), r(Z,Y).

   a(1,2).
   a(1,3).
   a(2,4).
   a(3,3).
   a(3,5).
   a(4,5).
   a(4,6).
   ```

   (a) [4 points] Show the SLD derivation tree for the query `r(2,A)` showing all derivations (successful, failed, or infinite). If the tree has infinite paths, indicate these paths and why they are infinite.

   (b) [4 points] Show the SLD derivation tree for the query `r(3,B)` showing all derivations (successful, failed, or infinite). If the tree has infinite paths, indicate these paths and why they are infinite.

4. [4 points] For each of the following claims, state whether the claim is true or false. In each of the claims, G denotes a definite goal, and P is a definite logic program. Briefly justify your answers.

   (a) If G has a finite SLD derivation with computed answer substitution θ, then $P \models \forall(G\theta)$.

   (b) If a is an atom in the least Herbrand model of P, then *every* SLD derivation of a (i.e. derivations with a as the goal) is finite.

[MORE QUESTIONS ON NEXT PAGE; PLEASE TURN OVER]

5. [6 points] Write a Prolog predicate `pick_odd(L, E)` that returns the elements that occur at odd positions in `L` one by one, upon backtracking. That is, returns the first element of list `L`, and upon backtracking the third element, the fifth element and so on.

   For instance, `pick_odd([1,2,4,8,16,32,64], X)` should first return `X=1`. Upon backtracking (i.e. if the user types ";" after that answer), it will return `X=4`; upon further backtracking it will return `X=16`; and upon further backtracking `X=64`.

   Write this predicate without using any arithmetic and without using any other helper predicates.

6. [6 points] Write an Prolog predicate `min` that takes two arguments and works as follows. Given a list of numbers `L`, `min(L, Min)` succeeds with `Min` bound to the smallest element of `L`. The predicate fails if `L` is empty.

   For instance, `min([1,2], M)` succeeds with `M=1`; `min([4,2,3], M)` succeeds with `M=2`; and `min([], M)` fails.

   You may use helper predicates for defining `min`.

7. [6 points] Consider the grammar for $\mathcal{L} = \{a^n b^n \mid n \geq 0\}$ given in DCG form below:

```
s -->  [a], s, [b].
s -->  [].
```

   (a) [2 points] Write the Prolog program generated from the above DCG.

   (b) [4 points] How long does it take to parse a string of length n using this DCG?