CSE-505 Computing with Logic

Final Exam

Dec 20, 2007

Max: 45 points Duration: 2 hours 30 minutes

- WRITE YOUR NAME and USB ID NUMBER legibly and sign in the space below.
- Write your answer for each question in the space provided.
- You are allowed one sheet of handwritten notes.
- The exam has 8 questions, over 11 pages (including this one).

Name: ______________________ **USB ID:** ______________

Signature: ______________________

Question	Max.	Score
1.	5	
2.	5	
3.	8	
4.	9	

Question	Max.	Score
5.	5	
6.	4	
7.	3	
8.	6	
Total:	45	

1. [5 points] Write a Prolog predicate `isPermutation(L1, L2)` that, given two lists `L1` and `L2` succeeds if and only if `L1` is a permutation of `L2`; i.e., `L2` can be constructed by reordering the elements of `L1`. For example,

   - the following queries succeed:
     - `isPermutation([a,b,c], [c,a,b])`
     - `isPermutation([a,b,c], [a,b,c])`
     - `isPermutation([a,b,a], [a,a,b])`
     - `isPermutation([a,b,c,e,a,d], [a,c,a,b,e,d])`
   - the following queries fail:
     - `isPermutation([a,b,c], [a,c])`
     - `isPermutation([a,b,a], [b,b,a])`
     - `isPermutation([a,b,c,e,a,d], [a,b,c,d,e,f])`

   Note: You may assume that in every query to `isPermutation`, the two arguments are given ground lists (i.e. not variables).

2. [5 points] Write a Prolog predicate `evens` such that `evens(L, X)` will succeed if and only if `X` occurs in an even position in `L`. For instance, `evens([a,b,c,d], b)` and `evens([a,b,c,d], d)` will both succeed, while `evens([a,b,c,d], a)` and `evens([a,b,c,d], c)` both fail. Moreover, `evens([a,b,c,d], X)` should first succeed with `X=b`, and upon backtracking with `X=d` (and no more answers).

   Note: *`evens` should be written without using arithmetic. For full credit, `evens` should be written without using any helper predicates.*

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

$$\begin{aligned}
&even(0).\\
&even(s(s(X))) \leftarrow\ even(X).\\
&odd(s(0)).\\
&odd(s(s(X)) \leftarrow\ odd(X).\\
&strange(X) \leftarrow\ even(X),\ odd(X).
\end{aligned}$$

(a) What is the (Clark's) completion of the above program?

(b) Is there a finitely failed SLD tree for the goal $\leftarrow strange(X)$? Explain.

[CONTINUED ON NEXT PAGE]

[CONTINUED FROM PREVIOUS PAGE]

(c) Consider the statement of *completeness* of negation as finite failure:

if $comp(P) \models \forall(\neg A)$ then there exists a finitely failed SLD tree of A.

In light of the program in the previous page, and the query in part (b) of this question, is the above statement true, i.e. is negation as finite failure complete? Briefly explain.

4. [9 points total] Consider the following program:

```
sg(X,X).
sg(X,Y) :- e(X,U), sg(U,V), e(V,Y).
```

where `e/2` represents the edge relation of a tree.

We'll first consider a complete binary tree with seven nodes labelled from 1 through 7 such that (A) 1 is the root of the tree, and (B) there is an edge from node i ($i > 1$) to node i `div` 2 (i.e. the quotient of dividing i by 2).

Note that all edges flow from a child to its parent in the tree. This tree will be represented by an edge relation with facts of the form `e(`i`, `i` div 2)` for all $2 \leq i \leq 7$.

(a) [3 points] When tabled resolution is used, what will be the entries in the call table made when resolving the query `sg(7,A)`? Briefly explain.

(b) [3 points] When tabled resolution is used, what will be the entries in the answer table for call `sg(7,A)`? Briefly explain.

[CONTINUED ON NEXT PAGE]

[CONTINUED FROM PREVIOUS PAGE]

(c) [3 points] Now consider the edge relation `e/2` representing an arbitrary complete balanced binary tree with $2^h - 1$ nodes. What is the maximum number of distinct calls to `sg` that will be made when resolving a query of the form `sg(`i`,A)` for some *leaf node* i using tabled resolution? Briefly explain.

5. [5 points] Consider the following Prolog program:

```
p(X, Y) :- q(X,X), !, r(X,Y).
p(X, Y) :- q(X,Z), r(Z,Y), !.
p(X, Y) :- q(X,Y).

q(a,b).
q(b,c).
q(c,c).

r(b,d).
r(c,c).
r(c,d).
```

(a) How many answers are there for the query `p(b,X)`? Explain briefly.

(b) How many answers are there for the query `p(X,Y)`? Explain briefly.

6. [4 points] Compute all the stable models and the well founded model for the following general logic program:

$$
\begin{array}{lcl}
q & \leftarrow & p, \neg r \\
q & \leftarrow & r \\
p & \leftarrow & q, r \\
p & \leftarrow & \neg r \\
r & \leftarrow & s, \neg v \\
s & \leftarrow & v \\
v & \leftarrow & s
\end{array}
$$

7. [3 points] Consider the following program:

$$
\begin{array}{lcl}
p & \leftarrow & \neg q, \neg r \\
q & \leftarrow & \neg p, s \\
r & \leftarrow & \neg v \\
s & \leftarrow & s, \neg v \\
v & \leftarrow & \neg s
\end{array}
$$

Given that the well-founded model for the above program is $\{p, \neg q, \neg r, \neg s, v\}$, list all the stable models of the program. Justify.

8. [6 points] Consider the following WAM instructions for building a term on the heap.

```
put_structure a/0, X5
put_structure h/2, X4
set_value X5
set_variable X6
put_structure g/1, X2
set_value X4
put_structure f/3, X1
set_value X2
set_variable X3
set_value X3
```

For each of the following registers, write the term in heap represented by the register when the above instructions are executed.

Register	**Term represented by Register**
X5	
X4	
X2	
X1	

END OF EXAM