

CSE-505 Computing with Logic

Fall 2007

Mid-Term Exam

Duration: 1h 20m

Oct 18, 2007

Max: 35 points

1. [3 points] Formalize the following sentences of natural language as formulas in predicate logic:

- (a) Every natural number has a successor.
- (b) Logic plays an important role in all areas of computer science.
- (c) The renter of a car pays the deductible in case of an accident.

2. [8 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))) .
```

- (a) [2 points] What is the Herbrand Universe of the above program?
 - (b) [6 points] 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:

```
ps(X1,X5) :- ps(X1,X2), pa(X2,X3), ps(X3,X4), pb(X4,X5) .
ps(X1,X1) .
pa(X1,X2) :- occurs(X1,X2,a) .
pb(X1,X2) :- occurs(X1,X2,b) .
occurs(1,2,a) .
occurs(2,3,a) .
occurs(3,4,b) .
occurs(4,5,b) .
```

Give an SLD derivation for the query `ps(2,4)` using the above program.

Show each step of the derivation, and the computed substitutions at each step.

4. [8 points] Write a Prolog predicate `substitute(L1, X, S, L2)` that, given a list `L1`, an element `X` and a list `S`, returns a list `L2` where every occurrence of `X` in `L1` is substituted by the sequence of elements in `S`. For example:

- `substitute([a,b,c,a,b], a, [d,e], Z)` succeeds with answer `Z = [d,e,b,c,d,e,b]`.
- `substitute([a,b,c,a,b], a, [], Z)` succeeds with answer `Z = [b,c,b]`.
- `substitute([a,c,d,a,b], d, [e,f], Z)` fails.

For full credit, queries to `substitute` must run in linear time.

5. [8 points] Consider the language given by the following grammar:

$$\begin{aligned} Pgm &\longrightarrow Seq \\ Seq &\longrightarrow Stmt \ Seq \\ Seq &\longrightarrow \epsilon \\ Stmt &\longrightarrow \text{assign} \\ Stmt &\longrightarrow \text{begin } Seq \ \text{end} \end{aligned}$$

In the above grammar, **begin**, **end** and **assign** are terminal symbols; *Pgm*, *Stmt*, and *Seq* are nonterminals; and *Pgm* is the start symbol of the grammar.

As usual, we will use lists of terminal symbols to represent strings. For instance, [**begin**, **assign**, **end**, **assign**, **assign**] is a string in the above language, and [**begin**, **assign**, **end**, **begin**, **assign**, **assign**] is *not* a string in the above language.

- (a) [2 points] Write a Prolog predicate **prog(L)** that, given a list **L** determines whether or not **L** represents a string in the above language.
- (b) [6 points] The “nesting level” of an **assign** is said to be the number of **begins** not matched by **ends** occurring before **assign**. For instance, in [**begin**, **assign**, **end**, **assign**, **assign**], the nesting level of the first **assign** is 1, while the nesting level of the second and third **assigns** is 0.

Write a Prolog predicate **nest(L, N)** that, given a list **L** representing a string in the above language, returns in **N** the maximum nesting level of an **assign** in **L**. The query **nest(L,N)** fails if **L** is *not* in the above language.