The University of British Columbia
Department of Computer Science
Midterm Examination 1 — Fall 2016

Computer Science 312
Functional and Logic Programming

Question 1 [10 marks]

Consider the following knowledge base, *KB*:

$p :- v.$ $\quad$ $t :- m.$
$p :- q,\ r.$ $\quad$ $t :- n.$
$q :- s,\ t.$ $\quad$ $n.$
$v :- w.$ $\quad$ $r :- s.$
$s.$

(a) [4 marks] v is not a logical consequence of *KB*. Explain what this means and show why v isn't a logical consequence of *KB*.

(b) [6 marks] p is a logical consequence of *KB*. Give a successful top-down derivation for the query `?- p.`

Answer clause	Clause resolved

Question 2 [12 marks]

(a) [6 marks] Suppose we want to represent the defaults: Birds fly. Emus are birds that don't fly. Penguins are birds that don't fly. Things on planes fly, unless the plane is broken. Planes are not broken by default.

Write a program, using negation as failure and only propositional atoms (no arguments to predicates) that has the following behaviour:

- It should answer "false" to the query `?- flies.`
- If someone were to add the fact *bird*, it should answer "true" to the query `?- flies.`
- If someone were to add just the fact *emu*, it should answer "true" to the query `?- bird.` and answer "false" to the query `?- flies.`
- If someone were to add just the fact *penguin*, it should answer "true" to the query `?- bird.` and answer "false" to the query `?- flies.`
- If someone we to add the facts *emu* and *on_plane*, it should answer "true" to the query `?- flies.`
- If someone we to add the facts *emu* and *on_plane* and *plane_broken*, it should answer "false" to the query `?- flies.`

You do not need to worry about dynamic declarations.

(b) [6 marks] Consider the logic program with negation as failure:

a :- b. b :- \+ d.
a :- \+ c. b :- c.
d. c :- f.
e :- \+ f.

Give the set of all atoms and negations of atoms that are a logical consequence (i.e., the atoms and their negations that would be produced by the bottom-up proof procedure for negation as failure). You do not need to give the derivation.

Question 3 [10 marks]

Suppose that times are represented as $am(H,M)$ for the time M minutes after hour H in the morning or as $pm(H,M)$ for M minutes after hour H in the afternoon. For example, $am(11,30)$ is 11:30 in the morning, $pm(1,30)$ is 1:30 in the afternoon, and $pm(12,30)$ is halfway between these times. Write a predicate *next_hour*$(T1,T2)$ that is true when time $T2$ is exactly one hour after $T1$. It does not need to wrap over midnight. You can only use the built-in predicates $<$ (which compares two arithmetic expressions) and *is* (where `V is E` is true if arithmetic expression `E` evaluates to number `V`). You can assume that $T1$ does not contain variables when called.

An example of its use is:

```
?- next_hour(am(10,23),T).
T = am(11, 23) .
?- next_hour(am(11,23),T1), next_hour(T1,T2).
T1 = pm(12, 23),
T2 = pm(1, 23) .
```

Question 4 [10 marks]

(a) [6 marks] Write a program *del*1(E,L,R) which is true when R is a list with the same elements as list L (in the same order) but with one instance of E removed. For example, it should have the following behaviour:

```
?- del1(a,[a,v,a,t,a,r],R).
R = [v, a, t, a, r] ;
R = [a, v, t, a, r] ;
R = [a, v, a, t, r] ;
false.
?- del1(a,[f,u,n],R).
false.
```

(b) [4 marks] What are all of the answers to the query

```
?- del1(2,L,[a,b,c,a]).
```

(Note that you should be able to do this, even if you cannot do part (a).)