Two hours

UNIVERSITY OF MANCHESTER
SCHOOL OF COMPUTER SCIENCE

Artificial Intelligence Programming

Date: Thursday 21st May 2009

Time: 14:00 – 16:00

Please answer any THREE Questions from the FOUR Questions provided

The use of electronic calculators is NOT permitted.
1. a) How will SWI Prolog respond in response to the following goals?

```
?- append(X, [c,d],[a,b,c,d]).
?- append(X, [e],[a,b,c,d]).
```

(2 marks)

b) To what values will the variables X and Y be bound on return from the following goals?

```
?- name(bag,X).
?- name(Y, [98, 101, 100]).
```

The ascii value of a is 97. (2 marks)

c) Write a Prolog predicate `pp1/2` which, if called with the first argument bound to an atom and the second uninstantiated, returns with the second argument bound to an atom having the same name as the first argument but with an additional `ed` on the end:

```
1 ?- pp1(whack,L).
   L = whacked ;
   No
```

The predicate should not resatisfy. (2 marks)

d) The past participles of most English verbs ending in an e require the addition of a d only. Write a predicate `pp2/2` to implement this feature. Use cut (!) to avoid duplicate tests. Again, the predicate should not resatisfy:

```
2 ?- pp2(whack,L).
   L = whacked ;
   No
3 ?- pp2(slice,L).
   L = sliced ;
   No
```

(4 marks)
e) Verbs ending in a consonant preceded by a vowel usually require the doubling of the final consonant, provided it is neither w or y. Add clauses to `pp2/2` to implement this feature. Again, use cut (!) to avoid duplicate tests. Say where the new clauses must be positioned in relation to the existing clauses.

4 ?- pp2(flog,L).
L = flogged ;
No
5 ?- pp2(flay,L).
L = flayed ;
No

You may wish to define auxiliary predicates for this problem. (10 marks)
2. a) Assuming the following semantically annotated context-free grammar rules

\[
\begin{align*}
\text{NP/ } \varphi (\psi) & \rightarrow \text{Det/ } \varphi, \text{N/} \psi \\
N/ \varphi & \rightarrow N/ \varphi \\
\text{Det/ } \lambda p \lambda q [\forall x (p(x) \rightarrow q(x))] & \rightarrow \text{every} \\
N/ \lambda x [\text{boy}(x)] & \rightarrow \text{boy}
\end{align*}
\]

show that the meaning of

\[
\text{every boy}
\]

is \( \lambda p[\forall x (\text{boy}(x) \rightarrow p(x))] \). \hspace{1cm} (6 marks)

b) Informally, what property (of properties) does this \( \lambda \)-term denote? \hspace{1cm} (2 marks)

c) Assuming the following semantically annotated context-free grammar rules

\[
\begin{align*}
\text{CP/ } \varphi (\psi) & \rightarrow \text{C-Specw/ } \varphi, \text{Cw/} \psi \\
\text{Cw/ } \lambda w [\varphi] & \rightarrow \text{C, IP/ } \varphi \\
\text{IP/ } \varphi (\psi) & \rightarrow \text{NP/ } \varphi, \text{I/} \psi \\
\text{I/ } \varphi & \rightarrow \text{I, VP/ } \varphi \\
\text{VP/ } \varphi (\psi) & \rightarrow \text{V/ } \varphi, \text{NP/} \psi \\
\text{NP/} \varphi & \rightarrow \text{RelPro/} \varphi \\
\text{RelPro/ } \lambda q \lambda p \lambda x [p(x) \land q(x)] & \rightarrow \text{who} \\
\text{V/ } \lambda s \lambda x [s(\lambda y [\text{love}(x,y)])] & \rightarrow \text{loves}
\end{align*}
\]

and a rule giving a wh-trace with index \( w \) the meaning \( \lambda p[p(w)] \), draw the (deep) structure of the CP

\[
\text{who every girl loves}
\]

and compute its meaning as a \( \lambda \)-term. \hspace{1cm} (10 marks)

d) Informally, what does this \( \lambda \)-term denote? \hspace{1cm} (2 marks)
3. a) Explain briefly the technique of *proof by contradiction.*

(2 marks)

b) Using the obvious non-logical signature of unary and binary predicates, write down sentences of first-order logic expressing the meaning of the following English sentences:

- No artist respects any artist
- Some beekeeper respects some beekeeper

(4 marks)

c) Similarly, write down a sentence of first-order logic expressing the meaning of the following English sentence:

- Some beekeeper is not an artist.

(2 marks)

d) Convert your answers to Part b) and the *negation of* your answer to Part c) into clausal form.

(6 marks)

e) Using resolution theorem-proving, show that the sentences in Part b) together entail the sentence in Part c).

(6 marks)
4.  a) Consider the Prolog predicate

\[
\text{app1}([X|L1],L2,[X|L3]) :- \text{app1}(L1,L2,L3).
\]
\[
\text{app1}([],L,L).
\]

How many calls to \text{app1} (counting the initial goal as the first) will be made on executing a goal of the form

\[- \text{app1}(L, M, L3) . \]

where the first argument \(L\) is instantiated to an \(n\)-element list, the second argument \(M\) is instantiated to a list, and the third argument is the variable \(L3\)? (2 marks)

b) Consider the Prolog predicate

\[
\text{rev1}([X|L1],L3) :- \text{rev1}(L1,L2),\text{app1}(L2,[X],L3).
\]
\[
\text{rev1}([],[]).
\]

How many calls to \text{app1} will be made altogether on executing a goal of the form \(\text{rev1}(L, R)\). where the first argument \(L\) is instantiated to an \(n\)-element list, and the second argument is the variable \(R\)? (4 marks)

c) Given the Prolog predicate

\[
\text{fn}(X/Y,Y/Z,X/Z).
\]

to what value will \(L\) be bound if the following goal is called?

\[
\text{fn}([a,b,c|U]/U,[1,2,3|V]/V,L/[]).
\] (2 marks)

d) Write a Prolog program to reverse a list in time roughly linear in the length of that list. (4 marks)

e) Write down the Prolog clause into which the Prolog pre-processor translates the grammar rules

\[
\text{s} \rightarrow \text{np}, \text{vp}.
\]
\[
\text{n} \rightarrow \text{[cat]}.
\] (4 marks)

f) Explain briefly the relationship between this translation and the clause in Part c). (2 marks)

g) From a programming point of view, what is problematic about the following Prolog grammar rule?

\[
\text{v_bar} \rightarrow \text{v_bar}, \text{adverb}.
\] (2 marks)

\text{END OF EXAMINATION}