Two hours

UNIVERSITY OF MANCHESTER
SCHOOL OF COMPUTER SCIENCE

Symbolic AI

Date: Wednesday 31st May 2017

Time: 09:45 - 11:45

Please answer any THREE Questions from the FOUR Questions provided.

This is a CLOSED book examination

The use of electronic calculators is NOT permitted

1. Prolog

a) Explain what is meant by *negation-as-failure* in logic programming. What result is returned by the following call in Prolog?

```
1 ?- A \= a.
```

(3 marks)

b) Define `not/1` using `call/1`, `!/0` (cut) and `fail/0`. (3 marks)

c) What happens on calling the goal `p`, after the following program has been loaded into Prolog?

```
p:- not(p).
```

(2 marks)

d) In the context of logic programming, what is meant by *unification*? (2 marks)

e) What happens when (SWI) Prolog is called with the goal

```
X = f(X).
```

and why does this represent a departure from the rule that `=/2` implements unification?

(2 marks)

f) Explain the meaning of the pre-defined Prolog predicate `==/2`. What answers are given to the following two queries? Explain your answer.

```
A = 6, B = 6, A == B.
A == B, A = 6, B = 6.
```

(2 marks)

g) Write a predicate `myU/3` which, when called with the first two arguments instantiated to lists containing no duplicate terms, binds the third to a list containing the union of these sets of terms, with any duplicates removed. Non-ground terms, however, should not be unified. As an example:

```
1 ?- myU([A,a,b,C],[b,a,C,D],L).
L = [A, b, a, C, D].
```

(4 marks)

h) Using the pre-defined predicates `not/1` and `=/2`, define a predicate `canU(T1,T2)` which succeeds if `T1` and `T2` can be unified at the time of calling, but without actually unifying them. Thus:

```
1 ?- canU(A,B), A = 6, B = 7.
A = 6,
B = 7.

2 ?- A= 6, B= 7, canU(A,B).
false.
```

(2 marks)

2. Planning and logic

In AI, the term *blocks world* denotes an imaginary situation where blocks a, b, c, ... are stacked in neat, vertical piles on an infinite table; a manipulator is able to pick up one block at a time (if there is nothing on top of it) and place it either on top of another pile of blocks or on the table (thus starting a new pile). The aim is generally to find a sequence of actions that will achieve a given goal, e.g. a is directly on top of b and b is directly on top of c.

a) Suppose a situation in the blocks world is represented in Prolog as a list of atoms, e.g.

```
[on(a,b), on(c,table), on(b,table), clear(a), clear(c)]
```

with the obvious interpretation. (A block is 'clear' if there is nothing on top of it.) Let a collection of goals be represented in the same way. Suppose the action of moving block `X` from `Y` (a block or the table) to `Z` (a block or the table) is represented by the Prolog term `move(X,Y,Z)`. Define in SWI Prolog a predicate `satisfied(Situation,Goals)` which determines whether a set of goals is satisfied in a given situation. Define also a predicate `effect(Sit1,Act,Sit2)` which binds `Sit2` to the result of performing the action `Act` in situation `Sit1`, failing if the action cannot be performed. You may use any SWI Prolog primitives.
(6 marks)

b) Explain how planning in such a situation can be represented as the problem of searching a node-labelled tree. Make clear what information needs to be stored at nodes in order to (i) prevent looping; (ii) recover a plan when a situation is found satisfying the goals. (4 marks)

c) In the *situation calculus*, the blocks world may be modelled using a signature of predicates $\text{on}(x,y,s)$, $\text{clear}(x,s)$, where s is a variable representing situations, together with the function $\text{move}(x,y,z)$ mapping blocks (or the table) to actions, and a binary function $\text{do}(a,s)$ indicating the situation which results from performing action a in situation s (and representing some unspecified object if the action cannot be performed). Using the situation calculus, write axioms expressing the effects of the move-action. (4 marks)

d) Again using the situation calculus, write axioms expressing the *non*-effects of the move-action. (4 marks)

e) Explain how, in the context of the above representation scheme, finding a plan to achieve a given collection of goals in the blocks world can be understood as a theorem-proving task. (2 marks)

3. Natural language semantics

   a) Consider the semantically annotated context-free grammar

$$\begin{array}{l} \text{IP}/\phi(\psi) \rightarrow \text{NP}/\phi \;\; \text{I}'/\psi \\ \text{I}'/\phi \rightarrow \text{I} \;\; \text{VP}/\phi \\ \text{VP}/\phi(\psi) \rightarrow \text{V}/\phi \;\; \text{NP}/\psi \\ \text{NP}/\phi(\psi) \rightarrow \text{Det}/\phi \;\; \text{N}/\psi \\ \text{I} \rightarrow \_\,s \\ \text{V}/\lambda s \lambda x[s(\lambda y[\text{love}(x,y)])] \rightarrow \textit{love} \end{array}$$

   Add appropriate lexical rules to represent the meanings of the nouns *boy* and *girl* and the determiners *some* and *every*. (6 marks)

   b) Assuming a rule of verb-movement (which you may state informally) produce a derivation of the meaning of the string

   Every boy loves some girl

   showing the resulting phrase-structure. You should show your working and indicate clearly the process of verb-movement. (6 marks)

   c) Using the same signature as in parts (a) and (b), write a first-order formula expressing the meaning of the sentence

   Every boy who some girl loves loves every girl.

   (2 marks)

   d) Expand your grammar to account for such sentences as that in part (c), assuming a rule of wh-movement, which you should explain informally. You need not derive the answer given to part (c), but you should draw the phrase-structure of the subject noun-phrase, indicating the effect of wh-movement and the functioning of the associated trace. (6 marks)

4. Resolution theorem-proving

a) Translate the following English sentences into first-order logic, using the signature of unary predicates *a* (artist), *b* (beekeeper) and the binary predicate *h* (hates):

No artist hates every beekeeper
Some beekeeper hates every artist.

(4 marks)

b) Translate the following English sentence into first-order logic, using the same signature:

Some beekeeper is not an artist.

(2 marks)

c) Put the answer to part (a) and the negation of the answer to part (b) into clausal form, introducing Skolem constants and functions as necessary. (4 marks)

d) Demonstrate, using resolution theorem-proving, that the sentence in (b) follows from the sentences in (a). Show clearly all steps of the derivation. (10 marks)

END OF EXAMINATION