Two hours

UNIVERSITY OF MANCHESTER
SCHOOL OF COMPUTER SCIENCE

Symbolic AI

Date: Monday 28th May 2012

Time: 14:00 - 16:00

Please answer any THREE Questions from the FOUR Questions provided

Use a SEPARATE answerbook for EACH Question

For full marks your answers should be concise as well as accurate.
Marks will be awarded for reasoning and method as well as being correct.

This is a CLOSED book examination

The use of electronic calculators is permitted provided they are
not programmable and do not store text

1. a) Using the Prolog predicates `call/1`, `!/0` ('cut') and `fail/0`, define the standard Prolog predicate `not/1`. (3 marks)

   b) What is the result of executing the Prolog query `not(X = a)`? (1 mark)

   c) In the context of Prolog, what is meant by the term *negation as failure*? Explain in what sense negation as failure is not true logical negation. (3 marks)

   d) What is the *unifier* of two expressions containing variables? Explain the term *most general unifier* (mgu). (4 marks)

   e) Find the most general unifiers of the following pairs of terms, τ_1 and τ_2, or write 'no unifier' if they do not unify. (The symbols x, y and z are variables.)

      i) $\tau_1 = r(a,y) \quad \tau_2 = r(x, f(x))$
      ii) $\tau_1 = r(f(x,x),x) \quad \tau_2 = r(y, g(y,y))$
      iii) $\tau_1 = t(x, g(x,x), z) \quad \tau_2 = t(f(a), y, h(y,y))$

      (3 marks)

   f) The default Prolog unification algorithm does not carry out the 'occurs check'. In which of the cases in Question 1e will a call of the form τ_1=τ_2 yield the wrong answer? (1 mark)

   g) Explain informally the difference between calling the Prolog goals `X = Y` and `X == Y`, where `X` and `Y` may be instantiated at the time of calling. Give (not necessarily ground) instantiations for which the first call succeeds, and the second fails. (3 marks)

   h) Explain informally the difference between calling the Prolog goals `X = Y` and `not(not(X = Y))`, where `X` and `Y` may be instantiated at the time of calling. (2 marks)

2. a) In the context of first-order theorem-proving, write the *resolution* rule. (3 marks)

   b) Similarly, write the *factoring* rule. (3 marks)

   c) Write the following formulas in clause form:

      i) $\forall x (p(x) \rightarrow \exists y (q(y) \wedge r(x,y)))$
      ii) $\forall x (p(x) \rightarrow \forall y (q(y) \rightarrow s(x,y)))$

   (4 marks)

   d) Write the *negation* of the following formula in clause form:

$$\forall x (p(x) \rightarrow \exists y (q(y) \wedge s(x,y)))$$

   (2 marks)

   e) Using resolution theorem-proving, show that the formulas given in Question 2c entail the (unnegated) formula given in Question 2d (8 marks)

3. a) What is a ‘context-free grammar’? (2 marks)

b) Explain why the context-free grammar given below fails to to account properly for the grammaticality or ungrammaticality of (1)–(4). (4 marks)

```
s ==> [np, vp].                  word('i', pronoun).
np ==> [det, noun].              word('a', det).
np ==> [pronoun].                word('girl', noun).
vp ==> [verb, np].               word('boy', noun).
                                 word('see', verb).
                                 word('sees', verb).
```

(1)	I see a girl
(2)	a girl see a boy
(3)	a girl sees a boy
(4)	a girl sees I

c) Rewrite this grammar either using features or by expanding the set of non-terminal symbols so that it allows the grammatical sentences from (1)–(4) and rules out the ungrammatical ones. (6 marks)

d) Explain how shift-reduce parsing works, and show how it would lead to an analysis of *‘I like Mary but I hate Susan’* given the set of categorial descriptors below. (6 marks)

```
I = np                           but = (s\s)/s
like = (s\np)/np                 hate = (s\np)/np
Mary=np                          Susan = np
```

e) How would you change the basic shift-reduce algorithm to enable it to cope with sentences like *‘I like Mary but Susan I hate’*. How would this change apply if you were trying to analyse this sentence? (2 marks)

4. a) State the principle of compositionality, and discuss the advantages and disadvantages of using compositional rules to translate from natural language to a meaning representation couched in a formal language. (5 marks)

   b) Use the semantically annotated grammar given below to construct a 'quasi-logical form' for the sentence *'Hob believes a witch killed a cow'*. (7 marks)

```
s(VP:NP) ==> np(NP), vp(VP).
np(DET:N) ==> det(DET), noun(N).
vp(V:NP) ==> tverb(V), np(NP).
vp(V:S) ==> sverb(V), s(S).

np(hob) ==> [hob].
noun(lambda(X, witch(X))) ==> [witch].
noun(lambda(Y, cow(Y))) ==> [cow].
det(lambda(P, lambda(Q, Q:(qq(exists(X::{P:X}, X)))))) ==> [a].
tverb(lambda(U, lambda(V, [kill(V, U)]))) ==> [killed].
sverb(lambda(A, lambda(B, [believe(B, A)]))) ==> [believe].
```

   c) What would you have to do to convert the QLF that you have obtained to a full logical form? Show all the full logical forms that you would obtain from this QLF. (5 marks)

   d) Which, if any, of the logical forms that you have constructed entail the existence of a witch and a cow? (3 marks)

END OF EXAMINATION