UNIVERSITY OF DUBLIN
TRINITY COLLEGE

Faculty of Engineering and Systems Sciences

DEPARTMENT OF COMPUTER SCIENCE

B.A.(Mod.) Computer Science

Junior Sophister Examination **Trinity Term 2003**

3BA2 Artificial Intelligence

Tuesday, 27th May Goldsmith Hall 9.30 – 12.30

T. Fernando, P. Cunningham and M. Brady

Answer **five** questions, at least **one** from each section. Each question is worth 20 marks.

Section A

1. (a) What is the difference between finite-state machines and Turing machines? Is the Halting problem unsolvable for finite-state machines or for Turing machines?

   (b) What does it mean for either kind of machine to be non-deterministic? Why is non-determinism important in artificial intelligence (AI)? Why is determinism also important in AI?

   (c) For which is non-determinism more problematic: finite-state machines or Turing machines? Answer this question by describing a search strategy that works for one but not for the other.

2. (a) What is the difference between *don't care* non-determinism and *don't know* non-determinism? To which should we apply search? Suppose we were attempting to answer a query

   ```
   ?- q.
   ```

   using the rule

   ```
   q :- a,b.
   ```

In what sense is the choice between `a` and `b` a case of don't care non-determinism? In what sense is that choice a case of don't know non-determinism?

(b) What is heuristic search? How do heuristics differ from costs?

(c) What is A* search, and what does it mean for it to be *admissible*? State three conditions that together are sufficient to guarantee admissibility.

3. Consider a knowledge base consisting of the clauses

```
conn(X,X).
conn(X,Y) :- arc(X,Z), conn(Z,Y).

con2(X,X).
con2(X,Y) :- arc(Z,Y), con2(X,Z).
```

(a) What is Clark's completion of `conn`? What is Clark's completion of `con2`?

(b) Compare and contrast `conn` and `con2`. Give an interpretation where `conn(a,b)` is true but `con2(a,b)` is not. Give another interpretation where `con2(a,b)` is true but `conn(a,b)` is not.

(c) Expand the knowledge base above with at most two `arc` facts such that Prolog says `yes` to

```
?- conn(a,b).
```

but fails to say `yes` to

```
?- con2(a,b).
```

4. (a) How can reasoning with equality give rise to non-termination? Give a concrete example.

(b) What does unification have to do with the *Unique Names Assumption* (UNA), and how does UNA keep us from cases of non-termination, as in part (a)?

(c) Can we introduce equalities such as `clarkKent = superman` without violating UNA? Justify your answer.

5. (a) What is abduction, and what purpose does it serve? How does it differ from deduction (logical consequence)?

(b) How are integrity constraints and assumables used in abduction?

(c) What is the difference between abducing a goal and explaining it?

6. (a) Describe the architecture of a single layer neural network that can learn to associate an output vector $\vec{t}$ with a general input vector $\vec{s}$, given that it can be trained with a training set of pairs $\vec{s} : \vec{t}$.

(b) Describe an algorithm for training this network using the training set of pairs $\vec{s} : \vec{t}$. You may assume, in order to simplify the training, that the data is in bipolar format.

(c) Explain why this architecture can only learn classification problems where the classes are linearly separable.

Section B

7. Write a Prolog program to read the words in a text file, count the total number of words and the total number of distinct words. For example, if the file contained *the cat sat on the mat*, there are six words in total, but only five distinct words, since the word *the* occurs twice.

   To simplify matters, assume (rather unrealistically) that all letters are lowercase and there is no punctuation in the file.

   Hence, or otherwise, compute an efficient coding scheme to be used for compressing the file by encoding more frequently-occurring words using shorter codes than less frequent words.

8. "Prolog is a synthesis of a pure logic programming language and a normal procedural language." Discuss this statement, giving an account of the features of the language that might support or refute it. Give examples.