ASTON UNIVERSITY

EXAMINATIONS 2012

CS2320

Introduction to Computational Intelligence

(FOUR questions)

Summer, 2012 2 hours

Answer ALL parts of ALL questions

Questions do not all have the same number of marks but the total is 100

THE USE OF CALCULATORS IS NOT ALLOWED

1) The **physical symbol system** theory of Newell and Simon states that all intelligent behaviour can be modelled by **symbolic expressions** that **designate** other symbolic expressions and **interpret** them if they are processes.

   a) Explain how designation and interpretation work for the following Lisp code

   ```
   (expt 2 3)
   ```

   (5 marks)

   b) Give FIVE reasons why the Lisp language is very well suited to programming artificial intelligence programs.

   (10 marks)

   c) Explain why the ability of computational intelligence software to find solutions to problems can be greatly influenced by the particular choice of **problem representation**. Your answer should refer to a suitable example such as calculating the number of matches in a knockout tennis tournament or the mutilated chessboard where dominos are required to cover all squares without going outside the board.

   (5 marks)

2) Consider the 8-puzzle problem with the following start and goal states:

8	1	3
	2	4
7	6	5

Start

1	2	3
8		4
7	6	5

Goal

   a) Draw the search tree created by the **breadth-first uninformed search** method and stop when the algorithm finds the solution. *Make sure that the first state you draw is the one with the blank tile in the top-left corner (i.e. you move the 8 down), because this makes it quicker to find the solution.*

   (10 marks)

   b) Consider the heuristic of "number of misplaced tiles" for estimating the distance from the goal state. Show the search tree generated by the **best-first informed** search method step-by-step, together with the open queue.

   (10 marks)

   c) Identify the main operational difference between the **best-first** and **A* informed** search algorithms and explain how they may produce different solution paths.

   (5 marks)

3) The following Lisp code sets up the rules and starting facts (working memory) for an expert system that identifies animals:

```
(setf *rules
   '((mammal ((hair y)(give-milk y)))
     (bird ((feathers y)(lay-eggs y)))
     (ungulate ((mammal y) (chew-cud y)))
     (zebra ((ungulate y)(black-and-white y)))
     (penguin ((bird y) (fly n) (swim y) (black-and-white y))))))

(setf *facts
   '((give-milk y)(hair y)(chew-cud y)(black-and-white y)))
```

The first rule can be interpreted as "If the animal has hair and gives milk then it is a mammal".

a) Explain how a **forward-chaining** reasoning process would use the rules and facts to identify the animal described by the facts at the start state. Your answer should show the sequence of rules leading to the goal, the consequent changes to the facts, and explain the following terms: **trigger** and **fire**.

(10 marks)

b) Suppose the mammal rule was set up as a variable called `mammal` as follows:

```
(setf mammal '(mammal ((hair y)(give-milk y)))
```

i) Write Lisp code for a function called `get-all-conditions` that takes the rule as a parameter and returns the conditions, as shown when called with the mammal rule:

```
>(get-all-conditions mammal)
((hair y)(give-milk y))
```

(5 marks)

ii) Write Lisp code for a function called `get-condition` that takes a rule and a property name as parameters and returns the matching condition or `NIL` if it does not exist *(you can use the function call from (i) in your answer, if appropriate, without redefining its code*:

```
>(get-condition 'hair mammal)
(hair y)
```

(5 marks)

iii) Write Lisp code for a function called `prove-condition` that takes a paired list of a property and its value such as `(hair y)` and the list of known facts. It returns the matching condition or `NIL` if the condition does not exist as shown in the following function call: *(note that the value, 'y' or 'n', of the property must match, not just the property name. You can call functions defined earlier in this question without giving their definitions again)*:

```
>(prove-condition '(hair y) *facts)
(hair y)
```

(8 marks)

4) Freemind is an open-source program for creating **mind maps** that was used during the laboratory classes for this module. One of the example knowledge domains used was choosing somewhere to live.

   i) Draw a mind map that represents what you think is the knowledge most relevant for helping somebody decide where to live. Your mind map should have at least FIVE concepts coming from the central node and a depth of THREE nodes (i.e. the path from the center node to the leaf node includes four nodes).

   (10 marks)

   ii) Explain how the properties of Freemind and mind-mapping in general help elicit knowledge from human experts and incorporate it into an intelligent knowledge-based system.

   (10 marks)

   iii) Explain the main properties of **semantic networks** and how they differ from mind maps. Include a description of how **intersection search** works to find relationships between two concepts or nodes.

   (7 marks)

END OF EXAMINATION PAPER