DUBLIN INSTITUTE OF TECHNOLOGY

DT228 BSc. (Honours) Degree in Computer Science

Year 3

WINTER EXAMINATIONS 2016/2017

INTRODUCTION TO ARTIFICIAL INTELLIGENCE [CMPU3051]

MR RICHARD LAWLOR

MONDAY 9TH JANUARY 1.00 P.M. – 3.00 P.M.

TWO HOURS

INSTRUCTIONS TO CANDIDATES
ANSWER **FOUR** QUESTIONS OUT OF **FIVE**.
ALL QUESTIONS CARRY EQUAL MARKS.

1 (a) Describe the architecture of a rule-based expert system.

(9 marks)

(b) Explain why the use of an expert system shell can dramatically reduce the development time of an expert system.

(6 marks)

(c) With the aid of a simple example, outline the Prolog inference mechanism, making reference to unification and backtracking.

(10 marks)

2 (a) Outline three sources of uncertain knowledge.

(6 marks)

(b) Give three reasons why Bayesian reasoning is difficult to use in an expert system. Also, briefly describe three advantages in using certainty factors as opposed to Bayesian reasoning. Briefly mention two well-known historical expert systems, one of which used Bayesian reasoning and the other certainty factors.

(10 marks)

(c) Given the forecasting rules below, show how they would fire to forecast tomorrow's weather when provided with the following information: there is rain today and the rainfall is low with a certainty factor of 0.8, and also it is cold with a certainty factor of 0.9. This formula may be useful:

$$cf(cf_1, cf_2) = cf_1 + cf_2 \times (1 - cf_1)$$

Rule: 1
if today is rain
then tomorrow is rain {cf 0.5}

Rule: 2
if today is dry
then tomorrow is dry {cf 0.5}

Rule: 3
if today is rain
and rainfall is low
then tomorrow is dry {cf 0.6}

Rule: 4
if today is rain
and rainfall is low
and temperature is cold
then tomorrow is dry {cf 0.7}

Rule: 5
if today is dry
and temperature is warm
then tomorrow is rain {cf 0.65}

Rule: 6
if today is dry
and temperature is warm
and sky is overcast
then tomorrow is rain {cf 0.55}

(9 marks)

3 (a) Represent the following knowledge in Prolog.

$\forall X \bullet \text{mammal}(X) \wedge \text{wagsTail}(X) \Rightarrow \text{dog}(X).$
$\forall X\ \forall Y \bullet \text{child}(X) \wedge \text{dog}(Y) \Rightarrow \text{likes}(X,Y).$
$\forall X\ \forall Y \bullet \text{child}(X) \wedge \text{spoilt}(X) \wedge \text{likes}(X,Y) \Rightarrow \text{has}(X,Y).$

(5 marks)

(b) What is the meaning and difference if any in the following two Prolog clauses?

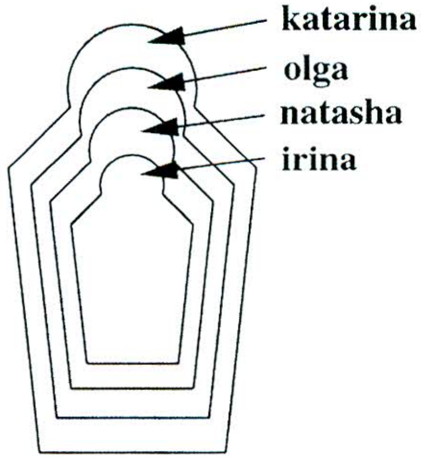
```
a :- not( city(dublin)).
a :- city(X), X \== dublin.
```

(5 marks)

(c) Write a Prolog predicate **fac(N, X)** to compute the factorial of N.

(5 marks)

(d) Given the information expressed in the Russian doll diagram, write a knowledge base using the predicate **directlyIn/2** which encodes which doll is directly contained in which other doll. Then, define a recursive predicate **in/2** , that tells us which doll is (directly or indirectly) contained in which other dolls.

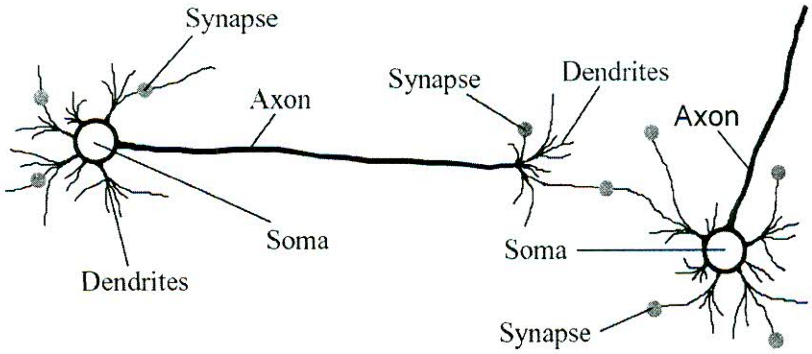


(5 marks)

(e) Write a list predicate **addLists(L1, L2, L)** where addLists([1,3,0], [2,7,6], L) gives L = [3,10,6].

(5 marks)

4 **(a)** Below is a simplified diagram of neuron connections in a brain. Provide a simple description of each of the terms in the diagram.



(5 marks)

(b) What is a perceptron? Draw a diagram for a single-layer two-input perceptron. Explain what is meant by linear separability. How many categories can it classify inputs into? Mention any issues involved in using a perceptron to compute logical AND, OR and XOR.

(7 marks)

(c) Draw a multilayer perceptron diagram that would be appropriate for an XOR gate. Name and draw the activation function which is suitable for a multilayer perceptron.

Comment briefly on its learning algorithm.

(8 marks)

(d) Illustrate the single perceptron training algorithm by computing the weight adjustments for the first two epochs in the table below.

(5 marks)

Perceptron Training for logical OR

Threshold $\theta = 0.2$, learning rate $\alpha = 0.1$.

Epoch	Inputs		Desired output	Weights		Weighted sum	Actual Output	Error	Weight adjustments	
	x_1	x_2	Y_d	w_1	w_2	X	Y	e	Δw_1	Δw_2
1	0	0	0	0.3	-0.1					
	0	1	1							
	1	0	1							
	1	1	1							
2	0	0	0							
	0	1	1							
	1	0	1							
	1	1	1							
3	0	0	0							

5 (a) In early AI research general problem solving methods were investigated which made the assumption that many problem domains could be characterised as a finite "*state space*" in which the problem solver could "operate" by moving from one state to another, making an allowed move.

Discuss this from the perspective of the *farmer-wolf-goat-cabbage* (fwgc) problem and include in your answer the *state space* graph for this problem.

(5 marks)

(b) Show how a fwgc state could be represented in Prolog and write some code for generating allowed moves.

(5 marks)

(c) Write a Prolog predicate solve/3 which could be used to find a solution path for the fwgc or any other similar problem.

(5 marks)

(d) Explain how an 8-puzzle state could represented in Prolog and use your scheme to represent the following state.

1	2	3
8		5
7	4	6

(5 marks)

(e) Could the code from part (c) be used to solve the 8-puzzle and if so would there be any major limitation were it to be applied to the 8-puzzle? If such a limitation exists in *solve/3*, describe a way (code not necessary) of overcoming it.

(5 marks)