Queen Mary University of London

First/Masters Degrees in Engineering by Course Units

May 2015 14:30 - 17:00

ECS629U/759P Artificial Intelligence Duration: 2 hours 30 minutes

YOU ARE NOT PERMITTED TO READ THE CONTENTS OF THIS QUESTION PAPER UNTIL INSTRUCTED TO DO SO BY AN INVIGILATOR.

Answer ALL Questions **THIS PAPER INCLUDES SOLUTIONS**

ONLY NON-PROGRAMMABLE CALCULATORS ARE PERMITTED IN THIS EXAMINATION. PLEASE STATE ON YOUR ANSWER BOOK THE NAME AND TYPE OF MACHINE USED.

COMPLETE ALL ROUGH WORKINGS IN THE ANSWER BOOK AND CROSS THROUGH ANY WORK WHICH IS NOT TO BE ASSESSED.

IMPORTANT NOTE:
THE ACADEMIC REGULATIONS STATE THAT POSSESSION OF UNAUTHORISED MATERIAL AT ANY TIME WHEN A STUDENT IS UNDER EXAMINATION CONDITIONS IS AN ASSESSMENT OFFENCE AND CAN LEAD TO EXPULSION FROM QMUL.

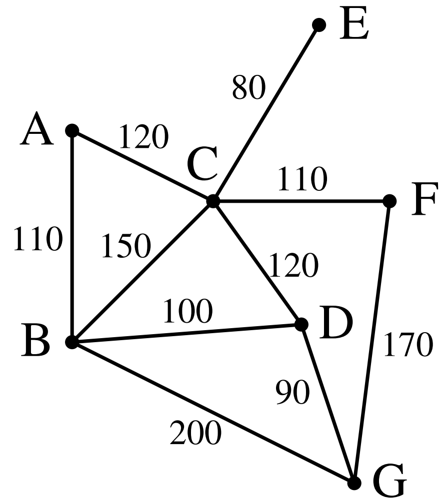
PLEASE CHECK NOW TO ENSURE YOU DO NOT HAVE ANY NOTES, MOBILE PHONES OR UNATHORISED ELECTRONIC DEVICES ON YOUR PERSON. IF YOU HAVE ANY THEN PLEASE RAISE YOUR HAND AND GIVE THEM TO AN INVIGILATOR IMMEDIATELY. PLEASE BE AWARE THAT IF YOU ARE FOUND TO HAVE HIDDEN UNAUTHORISED MATERIAL ELSEWHERE, INCLUDING TOILETS AND CLOAKROOMS IT WILL BE TREATED AS BEING FOUND IN YOUR POSSESSION. UNAUTHORISED MATERIAL FOUND ON YOUR MOBILE PHONE OR OTHER ELECTRONIC DEVICE WILL BE CONSIDERED THE SAME AS BEING IN POSSESSION OF PAPER NOTES. MOBILE PHONES CAUSING A DISRUPTION IS ALSO AN ASSESSMENT OFFENCE.

EXAM PAPERS CANNOT BE REMOVED FROM THE EXAM ROOM.

Examiners:
Dr S Dixon
Dr S Riis

Question 1

Consider the problem of finding the cheapest flight routes between pairs of cities, where a map as show to the right is given. Vertices represent cities, and are labelled by upper case letters. Edges represent direct flights between pairs of cities, and are labelled with the price of the flight (in pounds). Where there is a direct flight between a pair of cities, it operates in both directions and it has the same price in each direction. The required utility function considers only the cost, ignoring the number of separate flights, timetabling of flights, reputation of airline, etc.



a) Using depth-first search, what are the first 3 nodes expanded to find a path from F to B? (There is more than one correct answer. Just give one possible answer.)

[2 marks]

`Any one of: FCA, FCB, FCD, FCE, FCF, FGB, FGD or FGF`

b) Why is depth-first search not suitable for this problem? How can depth-first search be modified to make it suitable for this problem?

[4 marks]

`DFS is unsuitable due to cycles in the graph, which could lead to an infinite loop in DFS. This problem can be avoided by keeping a list of expanded nodes and only expanding nodes which have not already been expanded.`

c) What are the first 3 nodes expanded to find a path from F to B using breadth-first search?

[2 marks]

`Either of: FCG or FGC`

d) What advantages does breadth-first search have over depth-first search in general (i.e. not only for route-finding problems)? (Explain any terms you use.)

[4 marks]

Breadth-first search is complete (it always finds a solution if there is one) and optimal (in the sense that it finds the solution with the smallest number of steps).

e) What disadvantages does breadth-first search have for this route-finding problem?

[4 marks]

BFS has high memory and computational costs. It also does not optimise the utility function, so its solution is not guaranteed to have the least cost; instead it will have the minimum number of flights needed.

f) Execute uniform cost search to find a path from G to E. Show your working including the order of node expansion and the agenda at each step. What is the path and its cost found by this algorithm?

[6 marks]

Node Expanded	Agenda
Initial State	{(G,0)}
G	{(D,90),(F,170),(B,200)}
D	{(F,170),(B,190),(C,210)}
F	{(B,190),(C,210)}
B	{(C,210),(A,300)}
C	{(E,290),(A,300)}
E	Solution Found

The minimum cost path from G to E is to fly via D and C at a total cost of £290.

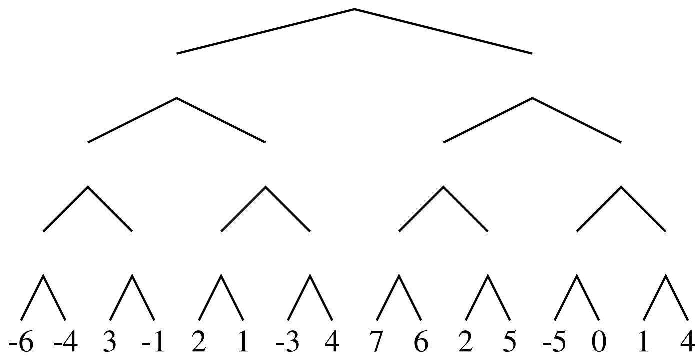
g) Would A* be a better algorithm for this problem? Why or why not?

[3 marks]

No. Since the prices of flights are arbitrary, it is not possible to design a non-trivial admissable heuristic function for this problem.

Question 2

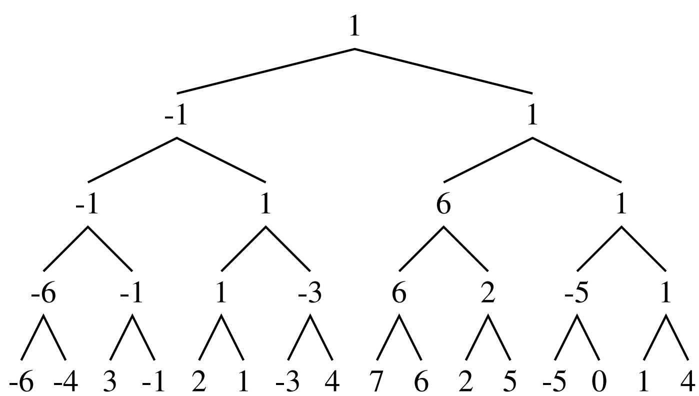
A two-player game has the following game tree for the last four moves. MAX tries to find the goal with the largest value and MIN the smallest one. It is MAX's turn to play.



a) Use the minimax algorithm to calculate the value at each node of the tree. (You will need to copy the whole tree into your answer book.)

[6 marks]

`Lose half a mark for each error.  Errors are penalised once only.`



b) What sequence of moves (left or right branch) is selected by each player at each level?

[2 marks]

`MAX chooses the right branch, then MIN chooses the right branch, MAX chooses the right branch and finally MIN chooses the left branch.`

c) Simulate the execution of α-β pruning on the tree, searching from left to right. Which branches are pruned and why? Refer to nodes by (*depth, position*), where *depth* is counted from the root (depth 1) and *position* from the left (also starting at 1). E.g. the fourth leaf, -1, is at node (5,4).

[6 marks]

```
At node (3,2), beta = -1, so after the left subtree is evaluated to 1, the right
subtree is pruned and 1 is returned.
At node (4,6), alpha = 6, so after node (5,11) evaluates to 2, the right subtree
(node (5,12)) is pruned and 2 is returned.
At node (4,7), alpha = -1, so after the left branch evaluates to -5, the right
branch (node (5,14)) is pruned and -5 is returned.
```

d) Neural Networks: What types of functions can be represented by an artificial neural network with an input and output layer only? Give an example of a function that can not be learnt by such a network.

[4 marks]

```
A single-layer network can only learn linearly separable functions.  One function
that can not be represented is exclusive-or.
```

e) Explain the backpropagation algorithm for training a multilayer neural network.

[7 marks]

```
Repeat until convergence:
    Select a training example
    Compute activations of all nodes using current weights
    For each node in the output layer
        Calculate delta values:  observed error × derivative of activation fn
    For remaining network layers (working backwards from output)
        Calculate deltas for each node:  sum of errors at the next layer,
                weighted by connection strengths, and
                scaled by derivative of activation function
    For each weight, increment by delta value × input × learning rate
```

Question 3

a) Give the formula for entropy H(S) as used in decision tree learning, and define each variable in the formula.

[4 marks]

$$H(S) = -\sum_{i=1}^{n} p_i \log_2(p_i)$$

where S is a sample of n items, p_i is the relative frequency of item i in S.

b) You are trying to understand your neighbour's cycle riding habits, and you hypothesise that her choice of riding or not riding to work is based on the weather. Based on the following 7 observations of the weather and your neighbour's choice of transport, construct a decision tree classifier using the ID3 algorithm. Show all working and draw the resulting decision tree.

Rain	Wind	Snow	RideBike
Y	Y	N	N
N	Y	N	Y
N	N	Y	Y
N	Y	Y	N
Y	Y	Y	N
Y	N	N	N
N	N	N	Y

[13 marks]

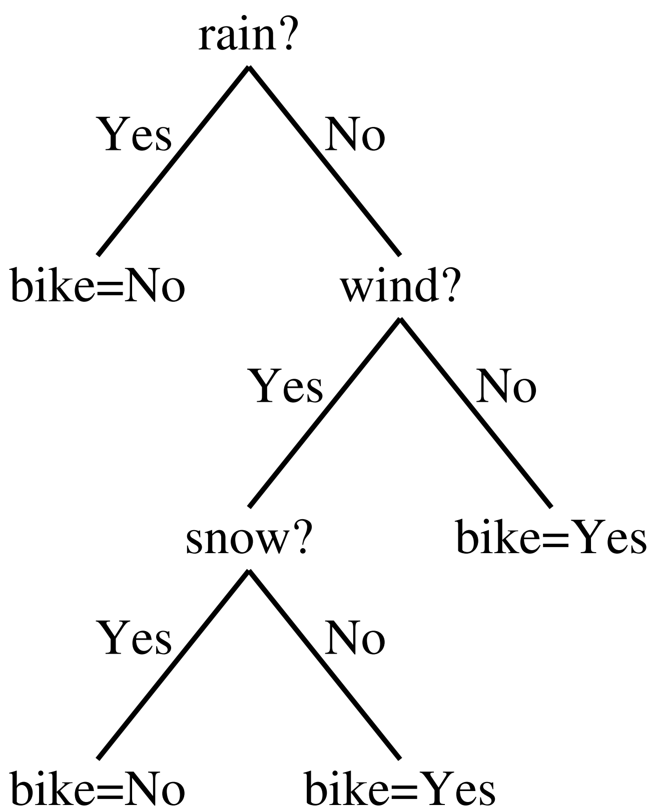
$$\begin{aligned}
H(S) &= -\tfrac{4}{7}\log_2(\tfrac{4}{7}) - \tfrac{3}{7}\log_2(\tfrac{3}{7}) \\
&= 0.985 \\
E(\text{rain} = \text{Y}) &= -\tfrac{0}{3}\log_2(\tfrac{0}{3}) - \tfrac{3}{3}\log_2(\tfrac{3}{3}) \\
&= 0 \\
E(\text{rain} = \text{N}) &= -\tfrac{3}{4}\log_2(\tfrac{3}{4}) - \tfrac{1}{4}\log_2(\tfrac{1}{4}) \\
&= 0.811 \\
G(\text{rain}) &= 0.985 - \tfrac{3}{7}E(\text{rain} = \text{Y}) - \tfrac{4}{7}E(\text{rain} = \text{N}) \\
&= 0.522 \\
E(\text{wind} = \text{Y}) &= -\tfrac{1}{4}\log_2(\tfrac{1}{4}) - \tfrac{3}{4}\log_2(\tfrac{3}{4}) \\
&= 0.811 \\
E(\text{wind} = \text{N}) &= -\tfrac{2}{3}\log_2(\tfrac{2}{3}) - \tfrac{1}{3}\log_2(\tfrac{1}{3}) \\
&= 0.918 \\
G(\text{wind}) &= 0.985 - \tfrac{4}{7}E(\text{wind} = \text{Y}) - \tfrac{3}{7}E(\text{wind} = \text{N}) \\
&= 0.128 \\
E(\text{snow} = \text{Y}) &= -\tfrac{1}{3}\log_2(\tfrac{1}{3}) - \tfrac{2}{3}\log_2(\tfrac{2}{3}) \\
&= 0.918 \\
E(\text{snow} = \text{N}) &= -\tfrac{2}{4}\log_2(\tfrac{2}{4}) - \tfrac{2}{4}\log_2(\tfrac{2}{4}) \\
&= 1 \\
G(\text{snow}) &= 0.985 - \tfrac{3}{7}E(\text{snow} = \text{Y}) - \tfrac{4}{7}E(\text{snow} = \text{N}) \\
&= 0.020
\end{aligned}$$

So first split is on the attribute *rain*. [5]

`For rain=Y, all examples are negative (bike=N).`
`For rain=N,` $H(S) = 0.811$ `(see above):`

$$
\begin{aligned}
E(\text{wind} = \text{Y}) &= -\tfrac{1}{2}\log_2(\tfrac{1}{2}) - \tfrac{1}{2}\log_2(\tfrac{1}{2}) \\
&= 1 \\
E(\text{wind} = \text{N}) &= -\tfrac{2}{2}\log_2(\tfrac{2}{2}) - \tfrac{0}{2}\log_2(\tfrac{0}{2}) \\
&= 0 \\
G(\text{wind}) &= 0.811 - \tfrac{2}{4}E(\text{wind} = \text{Y}) - \tfrac{2}{4}E(\text{wind} = \text{N}) \\
&= 0.311 \\
E(\text{snow} = \text{Y}) &= -\tfrac{1}{2}\log_2(\tfrac{1}{2}) - \tfrac{1}{2}\log_2(\tfrac{1}{2}) \\
&= 0.918 \\
E(\text{snow} = \text{N}) &= -\tfrac{2}{2}\log_2(\tfrac{2}{2}) - \tfrac{0}{2}\log_2(\tfrac{0}{2}) \\
&= 1 \\
G(\text{snow}) &= 0.811 - \tfrac{2}{4}E(\text{snow} = \text{Y}) - \tfrac{2}{4}E(\text{snow} = \text{N}) \\
&= 0.311
\end{aligned}
$$

`So the second split can use either` *`snow`* `or` *`wind`*`; we choose` *`wind`*`. [3]`
`Then for wind=N, we have only one class, bike=Y, but for wind=Y, we have to split on the final attribute` *`snow`* `to separate the final two examples. [2]`
`The resulting decision tree is [3 for tree]:`



c) Explain what overfitting is, and when it can cause problems with ID3.

[4 marks]

```
Overfitting is when a learned decision process matches too closely to its training
data, [1]
and thus it fails to generalise (classification performance drops on test
sets). [1]
It can occur when too few examples are associated with leaf nodes [1]
or where there is noise in the data. [1]
[Other valid explanations accepted.]
```

d) Explain the function of training, testing and validation sets and their relationship to overfitting with ID3.

[4 marks]

```
The labelled data is split into three sets.
The training set is used for standard decision tree learning [1]
Then the validation set is used for model selection (e.g. pruning the tree) to
address overfitting [1]
Usually this step is repeated with different training/validation splits
(cross-validation) to get a more robust estimate of accuracy [1]
The testing set is used to estimate how accurately the tuned result will work in
the future. [1]
```

Question 4

Consider the following situation:
Sam is a man with blue eyes and blonde hair. He is 1.8m tall. He likes dogs. Sally is a woman with green eyes and auburn hair. She is 1.6m tall. She owns a dog. Sally likes everyone who likes her dog.

a) Write down the ontology required to express this situation in predicate calculus, identifying the three different kinds of syntactic object that can appear in an ontology, and saying which is which. You may assume real numbers and associated tests (i.e., you do not need to write them down).

[5 marks]

```
Predicates:  Man, EyeColour, HairColour, Height, Likes, Dog, Woman, Owns [2]
Constants:  Sam, Blue, Blonde, Sally, Green, Auburn [2]
Functions:  (None) [1]
```

b) Translate the situation into predicate calculus using your ontology.

[8 marks]

Man(Sam)
EyeColour(Sam, Blue)
HairColour(Sam, Blonde)
Height(Sam, 1.8)
$\forall x.\texttt{Dog}(x) \rightarrow \texttt{Likes(Sam,}\ x\texttt{)}$
Woman(Sally)
EyeColour(Sally, Green)
HairColour(Sally, Auburn)
Height(Sally, 1.6)
$\exists d.\texttt{Dog}(d) \wedge \texttt{Owns(Sally,}\ d\texttt{)}$
$\forall x.\forall d.\texttt{Dog}(d) \wedge \texttt{Owns(Sally,}\ d\texttt{)} \wedge \texttt{Likes}(x,d) \rightarrow \texttt{Likes(Sally,}\ x\texttt{)}$

c) Give a resolution proof, showing and explaining your working, of the claim that Sally likes Sam.

[12 marks]

In resolution, use the negated goal:
$\neg$Likes(Sally, Sam) [1 mark]

Convert non-normal clauses to CNF. First, remove $\rightarrow$: [1 mark]
$\forall x.\neg$Dog(x) $\vee$ Likes(Sam, x)
$\exists d.$Dog(d) $\wedge$ Owns(Sally, d)
$\forall x.\forall d.\neg$(Dog(d) $\wedge$ Owns(Sally, d) $\wedge$ Likes(x, d)) $\vee$ Likes(Sally, x)

Then reduce scope of negations: [1 mark]
$\forall x.\neg$Dog(x) $\vee$ Likes(Sam, x)
$\exists d.$Dog(d) $\wedge$ Owns(Sally, d)
$\forall x.\forall d.\neg$Dog(d) $\vee$ $\neg$Owns(Sally, d) $\vee$ $\neg$Likes(x, d) $\vee$ Likes(Sally, x)

Then standardise variables apart: [1 mark]
$\forall x.\neg$Dog(x) $\vee$ Likes(Sam, x)
$\exists d.$Dog(d) $\wedge$ Owns(Sally, d)
$\forall y.\forall z.\neg$Dog(z) $\vee$ $\neg$Owns(Sally, z) $\vee$ $\neg$Likes(y, z) $\vee$ Likes(Sally, y)

Then Skolemise existentials and drop universals: [1 mark]
$\neg$Dog(x) $\vee$ Likes(Sam, x)
Dog(D) $\wedge$ Owns(Sally, D)
$\neg$Dog(z) $\vee$ $\neg$Owns(Sally, z) $\vee$ $\neg$Likes(y, z) $\vee$ Likes(Sally, y)

Split into disjunctive clauses: [1 mark]
$\neg$Dog(x) $\vee$ Likes(Sam, x)
Dog(D)
Owns(Sally, D)
$\neg$Dog(z) $\vee$ $\neg$Owns(Sally, z) $\vee$ $\neg$Likes(y, z) $\vee$ Likes(Sally, y)

Resolution proof: derive contradiction: [6 marks]

1. Man(Sam)
2. EyeColour(Sam, Blue)
3. HairColour(Sam, Blonde)
4. Woman(Sally)
5. EyeColour(Sally, Green)
6. HairColour(Sally, Auburn)
7. Height(Sam, 1.8)
8. Height(Sally, 1.6)
9. ¬Dog(x) ∨ Likes(Sam, x)
10. Dog(D)
11. Owns(Sally, D)
12. ¬Likes(Sally, Sam)
13. ¬Dog(z) ∨ ¬Owns(Sally, z) ∨ ¬Likes(y, z) ∨ Likes(Sally, y)

Resolve 12 with last literal in 13, unifying y with Sam:

14. ¬Dog(z) ∨ ¬Owns(Sally, z) ∨ ¬Likes(Sam, z)

Resolve 10 with first literal in 14, unifying z with D:

15. ¬Owns(Sally, D) ∨ ¬Likes(Sam, D)

Resolve 11 with first literal in 15:

16. ¬Likes(Sam, D)

Resolve 9 with 16:

17. ¬Dog(D)

Resolve 10 with 17:

18. ⊥

We have reached a contradiction, which completes the proof of our goal, Likes(Sally,Sam).

End of Paper