**Online Exam**: CSU33061 Artificial Intelligence I

*29 April 2022, 9:30-11:30* (+20 minutes for LENS students)

**Instructions:**

Answer two of the three questions, labelled Q1, Q2 and Q3.

Each question is worth 50 marks; two questions together are 100 marks.

Please upload a plaintext file or PDF. (PDF converters are available online.)

If you have any questions or encounter any issues with the test, please contact `Tim.Fernando@tcd.ie`.

**Tip**: As partial credit will be awarded where possible, try *not* to leave any part of a question you are answering blank. Write down some thoughts that show effort and understanding (even if the understanding is that the effort falls short). But to avoid penalties for overly long answers, please be concise.

**Declaration**: By taking this exam, you are declaring

> *I understand that this is an individual assessment and that collaboration is not permitted.*
>
> *I have not received any assistance with my work for this assessment.*
>
> *Where I have used the published work of others, I have indicated this with appropriate citation.*
>
> *I have not and will not share any part of my work on this assessment, directly or indirectly, with any other student.*
>
> *I have read and I understand the plagiarism provisions in the General Regulations of the University Calendar for the current year, found at* `http://www.tcd.ie/calendar`.
>
> *I have also completed the Online Tutorial on avoiding plagiarism, located at* `https://libguides.tcd.ie/plagiarism/ready-steady-write`.
>
> *I understand that by taking this exam, I am agreeing with the statement above.*

[end of Declaration]

**Question Q1** This question is about graph search, which assumes

(i) a binary predicate `arc/2` between nodes

(ii) a unary predicate `goal/1` on nodes

(iii) and a node `Start`.

A simple search for a `goal` node along `arcs` from `Start` is the query

```
search(Start)
```

where

```
search(Node) :- goal(Node).
search(Node) :- arc(Node,Next), search(Next).
```

(a) True or False: The search described above is depth-first.

[2 marks]

(b) True or False: For any graph, depth-first search will terminate if the graph has only finitely many arcs.

[4 marks]

(c) True or False: For any graph, breadth-first search will terminate if the graph has only finitely many arcs.

[4 marks]

(d) For this part, (d), of Question Q1, suppose that every arc has cost $> 0$.

(i) If $n$ is a node for which there is a path from $n$ to a goal node, does it follow that there is a path from $n$ to a goal node such that the cost of that path is less than or equal to the cost of any path from $n$ to a goal node? Briefly justify your answer.

[9 marks]

(ii) If $h$ is a heuristic function such that for every node $n$, $h(n)$ is less than or equal to the cost of any path from $n$ to a goal node, does it follow that whenever there is a path from $n$ to a goal node, A-star will find one with minimal cost? Briefly justify your answer.

[9 marks]

(e) The rest of Question Q1 concerns Assignment 1, where A-star is used to search propositional knowledge bases such as

```
q:- a.
q:- b,c.
b:- c.
c:- e,f.
e.
```

which is represented as a list KB of lists

```
[[q,a],[q,b,c],[b,c],[c,e,f],[e]]
```

for use in the clauses

```
arc([H|T],Node,Cost,KB) :- member([H|B],KB), append(B,T,Node),
                           length(B,L), Cost is 1+ L/(L+1).
heuristic(Node,H) :- length(Node,H).
goal([]).
```

You were asked to define a predicate

```
astar(+Node,?Path,?Cost,+KB)
```

that implements A*, returning a path from Node to the goal node [] with minimal cost, given KB.

(i) Is A-star admissible under the given arc costs and heuristic function. Briefly justify your answer.

[5 marks]

(ii) Give the first 7 nodes searched by the query

```
?- astar([q],Path,Cost,
         [[q,a],[q,b,c],[a,d],[b],[d,c],[c,d]]).
```

[7 marks]

(iii) Given a list *kb* encoding a propositional Prolog knowledge base as specified above, what if anything would change in our A-star search if instead of using *kb* we used the reverse of *kb*?

[10 marks]

**Question Q2** This question concerns Markov Decision Processes (MDPs) and Q-learning.

(a) What effect does raising the discount factor $\gamma$ have on the decision an MDP is designed to process?

[4 marks]

(b) Let us fix an MDP $\langle S, A, p, r, \gamma \rangle$ with $0 \leq \gamma < 1$, and agree that

- a *policy* is a function $\pi : S \to A$ specifying an action $\pi(s) \in A$ for every state $s$
- the *value* $V^\pi(s)$ *of* a state $s$ *under* a policy $\pi$ satisfies

$$V^\pi(s) \;=\; \sum_{s' \in S} p(s, \pi(s), s')(r(s, \pi(s), s') + \gamma V^\pi(s'))$$

- a policy $\pi$ is *optimal* if

$$V^\pi(s) \geq V^{\pi'}(s) \;\text{ for any } \pi' : S \to A \text{ and } s \in S.$$

(i) True or False: There can be at most one optimal policy (per MDP).

[4 marks]

(ii) True or False: If $\pi$ is optimal, then $V^\pi(s) = \max_{a \in A} Q^\pi(s, a)$ where

$$Q^\pi(s, a) \;=\; \sum_{s' \in S} p(s, a, s')(r(s, a, s') + \gamma V^\pi(s')).$$

[4 marks]

(iii) Assuming $\pi$ is optimal and $V^\pi(s_1) > V^\pi(s_2)$, does it follow that $r(s, a, s_1) \geq r(s, a, s_2)$ for all states $s$ and actions $a$?

[4 marks]

(iv) Assuming $\pi$ is optimal and $V^\pi(s_1) > V^\pi(s_2)$, does it follow that $r(s_1, a, s) > r(s_2, a, s)$ for some state $s$ and action $a$?

[4 marks]

(v) Suppose $\pi$ is optimal, $s$ is a state, and $a, a'$ are actions such that

$$r(s, a, s') > r(s, a', s') \;\text{ for all states } s'.$$

Is it possible that $\pi(s) = a'$? Briefly explain your answer.

[10 marks]

(c) What ingredients do an MDP and Q-learning have in common?

[5 marks]

(d) True or False: During Q-learning, the learning rate $\alpha$ should be decreased as the Q-table is updated. Briefly justify your answer.

[5 marks]

(e) Suppose it is known that doing any action $a$ at any state $s$ always leads to the same state $s'$; i.e., $s \xrightarrow{a} s'$ is deterministic in that there is *no* $s'' \neq s'$ such that $s \xrightarrow{a} s''$. Given this knowledge, suppose we were to proceed with Q-learning as follows:

(†) given a state $s$, we choose an action different from any action that we have already tried at $s$, and stop updating our Q-table once we have tried every action $a$ at every state $s$.

Is (†) justified by the aforementioned determinism of transitions $\xrightarrow{a}$ between states? Briefly explain your answer.

[10 marks]

**Question Q3** This question is about topics covered after Q-learning.

(a) Why is it the case that every set of definite clauses has a model, whereas a set of Horn clauses need not?

[10 marks]

(b) Recall that a *Constraint Satisfaction Problem* is a triple [Var, Dom, Con] consisting of a list Var $= [X_1, \ldots, X_n]$ of *variables* $X_i$, a list Dom $= [D_1, \ldots, D_n]$ of finite sets $D_i$ of size $s_i$, and a finite set Con of *constraints* that may or may not be satisfied by a node instantiating $X_i$ with a value in $D_i$ for a search space of size $\prod_{i=1}^{n} s_i$. Briefly explain what is gained by enlarging that search space to $\prod_{i=1}^{n}(s_i + 1)$ by allowing a variable to be un-instantiated.

[10 marks]

(c) Given a Constraint Satisfaction Problem [Var, Dom, Con] what more do we need to get a *belief network* over Var, Dom?

[10 marks]

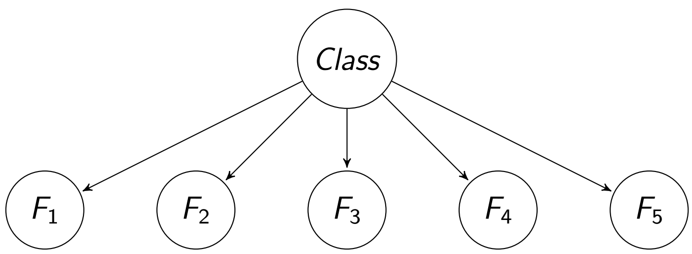
(d) Recall that a *naive Bayes classifier* assigns a class on the basis of features $F_{1:n} = F_1, \ldots, F_n$

$$P(Class|F_{1:n}) \;=\; \frac{P(F_{1:n}|Class)P(Class)}{P(F_{1:n})}$$

assuming $F_i$ are independent of each other given *Class*

$$P(F_{1:n}|Class) \;=\; \prod_i P(F_i|Class).$$

This can be pictured as




for $n = 5$.

Why is it *not* necessary to calculate $P(F_{1:n})$ to decide what value *Class* is most likely to have, given values to features $F_{1:n}$?

[5 marks]

(e) What is the Markov net corresponding to the naive Bayes classifier?

[15 marks]