Coláiste na Tríonóide, Baile Átha Cliath
Trinity College Dublin
Ollscoil Átha Cliath | The University of Dublin

Faculty of Engineering, Mathematics and Science

School of Computer Science & Statistics

Integrated Computer Science Programme **Semester 2 2019**
B.A. (Mod.) Computer Science & Business
B.A. (Mod.) Computer Science & Language
Year 3 Annual Examinations

Artificial Intelligence I

Tue, 23 April 2019 **RDS-SIM COURT** **14:00 – 16:00**

Dr Tim Fernando

Instructions to Candidates:

Attempt *two* questions. All questions carry equal marks. Each question is scored out of a total of 50 marks.

You may not start this examination until you are instructed to do so by the Invigilator.

Exam paper is not to be removed from the venue.

Materials permitted for this examination:

Non-programmable calculators are permitted for this examination — please indicate the make and model of your calculator on each answer book used.

1. Recall that we can encode a propositional knowledge base such as

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

as the list

```
[ [q,a], [q,b,c], [a,d], [b,c], [c], [d,a] ].
```

We can then explore various ways to prove an atom `X` from a knowledge base `KB` by different strategies for searching a path from `[X]` to `[]`, where

(i) a node is a list of atoms to prove

(ii) an arc is defined from a non-empty list with head `H` to a list in which `H` is replaced by subgoals that, under `KB`, yield `H`.

Depth-first search can be implemented as follows.

```
prove(X,KB) :- dfsearch([X],KB).

dfsearch(Node,_) :- goal(Node).
dfsearch(Node,KB) :- arc(Node,Next,KB), dfsearch(Next,KB).

goal([]).
```

(a) Define the predicate `arc/3` so that, for instance,

```
?- arc([q,d],Next,[[q,a],[q,b,c],[a,d],[b,c],[c],[d,a]]).
Next = [a,d] ;
Next = [b,c,d] ;
false.
```

[5 marks]

(b) How are the two queries below answered?

```
?- prove(b, [[q,a],[q,b,c],[a,d],[b,c],[c],[d,a]]).
?- prove(q, [[q,a],[q,b,c],[a,d],[b,c],[c],[d,a]]).
```

[6 marks]

(c) For search strategies other than depth-first, one approach is to maintain a list, called a **frontier**, of candidate nodes, explored from the head of the list on. Show how to define a predicate `frontierSearchBF/2` so that `proveBF(X,KB)` below uses breath-first search to check if `X` can be proved from `KB`.

```
proveBF(X,KB) :- frontierSearchBF([[X]],KB).
```

[9 marks]

(d) What is **A-star** and how is it different from breadth-first search? How can we modify `frontierSearchBF/2` to a predicate `astar/2` so that `proveAst(X,KB)` below uses A-star to check if `X` can be proved from `KB`.

```
proveAst(X,KB) :- astar([[X]],KB).
```

[20 marks]

(e) Yet another approach to checking if `KB` proves `X` proceeds from the goal node `[]`. For every integer $n \geq 0$, we compute the set G_n of nodes that can reach the goal node `[]` in n arcs but not less. Let G_0 consist solely of `[]`. And for $n \geq 0$, let G_{n+1} consist of all nodes L such that

(i) there is an arc from L to some node in G_n, and

(ii) for every non-negative integer $k \leq n$, L is *not* in G_k.

Given a knowledge base such as the list

```
[ [q,a], [q,b,c], [a,d], [b,c], [c], [d,a] ]
```

above, can we specify an integer $n \geq 0$ such that G_n is empty? Justify your answer, and explain why the question of specifying such an n is interesting.

[10 marks]

2. Let $\langle S, A, p, r, \gamma \rangle$ be a **Markov decision process** (MDP).

   (a) What decision is an MDP designed to analyse?

   [5 marks]

   (b) Compute the γ-**discounted value** $q_1(s_1, a_1)$ assuming

   $$\begin{aligned} S &= \{s_1, s_2\} \\ A &= \{a_1, a_2\} \\ \gamma &= \frac{1}{2} \end{aligned}$$

   and probabilities and immediate rewards given by Table a_1 and Table a_2 as follows: the pair

   $$p(s, a_i, s'), r(s, a_i, s')$$

   is the entry of Table a_i at row s, column s'.

a_1	s_1	s_2
s_1	.7, 3	.3, 0
s_2	.3, 0	.7, 1

Table a_1

a_2	s_1	s_2
s_1	.2, 4	.8, 2
s_2	.1, 1	.9, 6

Table a_2

   [20 marks]

   (c) Suppose that contrary to part (b), every action $a \in A$ in the MDP is **deterministic** in that there is a function $next : (S \times A) \to S$ such that for every state $s \in S$ and action $a \in A$,

   $$p(s, a, next(s, a))) = 1.$$

   Give a formula for the γ-discounted value $q(s, a)$ of s, a.

   [7 marks]

   (d) What is the **learning rate** α in Q-learning, and what value should α take assuming actions are deterministic according to part (c) above?

   [10 marks]

   (e) Should the **exploration-exploitation** tradeoff shift as we learn more about the values $Q(s, a)$? Explain.

   [8 marks]

3. (a) Recall that a **constraint satisfaction problem** [Var,Dom,Con] consists of a list Var of variables X_i, a list Dom of finite sets D_i (for variables X_i), and a list of constraints. A classic approach to solving [Var,Dom,Con] backtracks over a graph built from instantiating some variables X_i in Var with elements in D_i. What exactly are the nodes of this graph, and what are its arcs? What is the starting node, and which nodes are goal nodes?

[15 marks]

(b) Given a clause φ and a set KB of clauses, what does it mean for φ to be a **logical consequence of** KB (written $KB \models \varphi$)? You may assume a well-defined notion of truth of a clause at an interpretation.

[5 marks]

(c) What does it mean for a mechanical procedure $\vdash$ for logical consequence to be **sound**? What does it mean for $\vdash$ to be **complete**? Illustrate with an example of each.

[10 marks]

(d) Describe what a **Naive Bayes classifier** is, as a Bayes net (belief network)?

[10 marks]

(e) Given a graphical model, what is the **Markov blanket** of a node? What does this amount to in the case of an undirected graphical model (Markov network)?

[10 marks]