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

Faculty of Engineering, Mathematics & Science

School of Computer Science & Statistics

Integrated Computer Science **Sample**
Computer Science & Business
Computer Science & Language
Mathematics

Artificial Intelligence 1

Mon, 29 Apr 2024 **RDS-SIM COURT** **14:00 – 16:00**

Dr Tim Fernando

Instructions to Candidates:

Answer both questions. Each question is 50 points (for a total of 100).

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

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.

Question Q1 This question is about searching a graph for a path from a given node `Start` to some goal node. The predicate `frontierSearch/1` below outlines a general approach on which to add definitions of `goal/1, arc/2` and `add2frontier/3`.

```
search(Start):- frontierSearch([Start]).
frontierSearch([Node|_]):- goal(Node).
frontierSearch([Node|More]):- findall(Next,arc(Node,Next),Children),
                              add2frontier(Children,More,New),
                              frontierSearch(New).
```

(a) Revise `frontierSearch(+Frontier)` to `froSearch(+Frontier,?Path)` so that `froSearch([[Start]],Path)` returns a `Path` from `Start` to a goal node (initializing the frontier to the list `[[Start]]` consisting of one path `[Start]`). As with the definition of `frontierSearch` above, leave the predicates `goal/1, arc/2` and `add2frontier/3` unspecified.

[5 marks]

(b) To define `add2frontier`, recall that A-star uses two functions, *cost* and *h*, to assign each arc a cost, and each node a heuristic estimate of its distance to a goal node.

(i) One way to define distance between nodes is the minimum cost of a path between the nodes. How is the cost of a path computed from costs of arcs?

[4 marks]

(ii) True or False: If the heuristic estimate of every node is 0, then A-star does a *min-cost* search.

[3 marks]

(iii) How can we define the functions *cost* and *h* so that on these functions, A-star does *depth-first* search?

[4 marks]

(iv) True or False: if A-star searches breadth-first then A-star is *admissible*.

[4 marks]

(v) True or False: If the heuristic function *h over*estimates the cost of a path to a goal node, then A-star is not admissible. Briefly explain your answer.

[7 marks]

(c) Recall from Homework 1 that a propositional Prolog knowledge base such as

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

can be represented as the list

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

and used as `KB` in `arc(Node,Next,KB)` to pick out arcs `(Node,Next)` along which Prolog tries to find a path to `[]` (the goal node), where

```
arc([H|T],Node,KB) :- member([H|B],KB), append(B,T,Node).
```

Thus, we can analyze the Prolog query `?-q` against the knowledge base above as a search for a path from the node `[q]` to `[]`. The arc from `[q]` to `[a]` fails to lead to a path to `[]` because

(†) there is no arc from `[a]` (i.e., no member of `KB` has head `a`)

whereas the arc from `[q]` to `[b,c]` leads to the path

```
[q], [b,c], [c,c], [c], []
```

which we can shorten to

```
[q], [b,c], [c], []
```

if we replace `arc(Node,Next,KB)` by `arc2(Node,Next,KB)` to reduce `Next` to a set (without repeating members) as in

```
arc2(Node,Next,KB) :- arc(Node,Nx,KB),
                      makeSet(Nx,Next).
makeSet([],[]).
makeSet(List,Set) :- setof(X,member(X,List),Set).
```

Line (†) above suggests revising `arc2(Node,Next,KB)` further to

```
arc3(Node,Next,KB) :- arc2(Node,Next,KB),
                      allHeads(Next,KB).
allHeads([],_).
allHeads([H|T],KB):- member([H|_],KB), allHeads(T,KB).
```

That is, `arc3(Node,Next,KB)` implies `Next` is a set of heads of members of `KB`.

(i) Which of (A), (B) and (C), if any, are true for all instantiations of `Node`, `Next` and `KB`?

(A) `arc3(Node,Next,KB)` implies `arc2(Node,Next,KB)`

(B) `arc2(Node,Next,KB)` implies `arc(Node,Next,KB)`

(C) `arc3(Node,Next,KB)` implies `arc(Node,Next,KB)`

[5 marks]

(ii) True or False: if `arc3(Node,Next,KB)` then there is a path from `Next` to `[]` along arcs satisfying `arc3`. Briefly explain your answer.

[5 marks]

(iii) True or False: if there is a path of length k from `Start` to `[]` along arcs satisfying the predicate `arc`, then there is a path of length $\leq k$ from `Start` to `[]` along arcs satisfying `arc3`. Briefly explain your answer.

[5 marks]

(d) 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.

[8 marks]

Question Q2

(a) What is the decision that an MDP is set up to analyze?

[4 marks]

(b) True or False: in a MDP $\langle S, A, p, r, \gamma \rangle$, an agent can do any action $a \in A$ at any state $s \in S$ to reach any state $s' \in S$ for an immediate reward of $r(s, a, s')$. Briefly explain your answer.

[4 marks]

(c) In an MDP $\langle S, A, p, r, \gamma \rangle$ with discount factor $\gamma = 0$, what action $a \in A$ should an agent do at a state $s \in S$?

[4 marks]

(d) The remainder of Q2 is about a baby variant of the grid from Homework 2, reduced from 5×5 to 3×3, and differing in other respects. To be precise, let us fix a discount factor $\gamma = .9$ and flesh out an MDP $\langle S, A, p, r, .9 \rangle$ where a state $s \in S$ is a (row,column)-pair of integers from $\{1, 2, 3\}$, an action $a \in A$ is one of: up, down, left, right

$$S = \{1, 2, 3\} \times \{1, 2, 3\}$$
$$A = \{\text{up, down, left, right}\}$$

(1,1)	(1,2)	(1,3)
(2,1)	(2,2)	(2,3)
(3,1)	(3,2)	(3,3)

and every immediate reward $r(s, a, s')$ is 0 unless s' is either (2,2) for a losing -1 or (3,3) for a winning $+1$

$$r(s, a, s') = \begin{cases} -1 & \text{if } s' = (2,2) \\ +1 & \text{if } s' = (3,3) \\ 0 & \text{otherwise} \end{cases}$$

	-1	
		$+1$

which leaves transition probabilities $p(s, a, s')$ to be specified below.

(i) Recall that the optimal γ-discounted reward $Q(s, a)$ can be approximated by value iterations $q_n(s, a)$ converging to it at the limit

$$Q(s, a) = \lim_{n \to \infty} q_n(s, a)$$

where $q_0(s, a)$ is the expected immediate reward for doing a at s, and for $n \geq 0$,

$$q_{n+1}(s, a) = \sum_{s'} p(s, a, s')(r(s, a, s') + \gamma \max_{a'} q_n(s', a')).$$

What is the formula for $q_0(s, a)$?

[5 marks]

(ii) Let us suppose that if s is either (2,2) or (3,3) then for every action a, $p(s, a, s) = 1$. What are $q_n((2, 2), a)$ and $q_n((3, 3), a)$ for every integer $n \geq 0$?

[9 marks]

(iii) Let us turn next to a state s different from (2,2) and (3,3). If s has a square in the direction of an action a from it (in the 3×3 grid), let s_a be that state; otherwise, let s_a be s. For example,

$$(1, 1)_{\text{right}} = (1, 2) \ \text{ but } \ (1, 1)_{\text{left}} = (1, 1)_{\text{up}} = (1, 1).$$

Now, suppose

$$p(s, a, s_a) = 1 \text{ for every } a \in A \text{ and } s \in S \text{ different from (2,2) and } (3, 3).$$

What are $q_n((2, 3),\text{left})$ and $q_n((2,3),\text{down})$ for every integer $n \geq 0$?

[9 marks]

(e) The notion of a random variable can be formalized using three distinct but related notions:

(i) the probability measure $\mu(S)$ of a set S of possible worlds

(ii) the probability $P(\alpha)$ of a proposition α

(iii) the probability distribution P_X of a variable X.

How do these three notions differ and what do they have in common? How are they inter-related?

[15 marks]