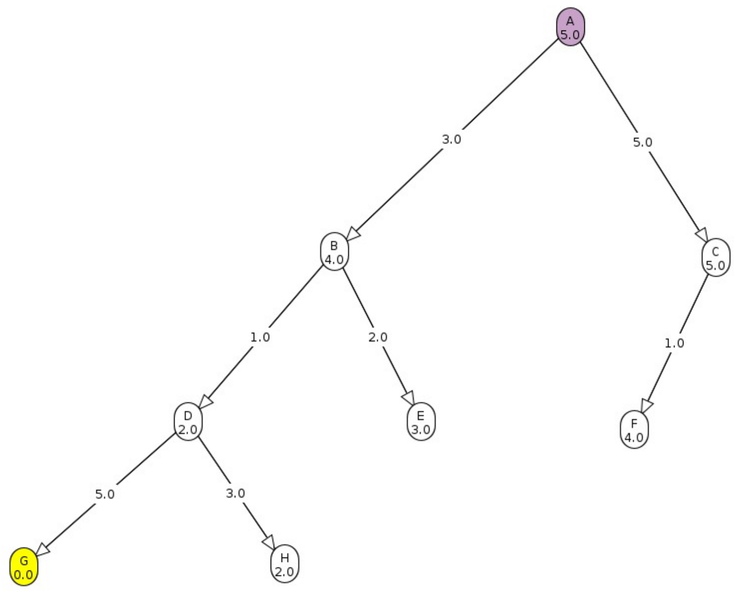
21st CSEC – Past Year Paper Solution (2020 – 2021 Semester 2)
CZ3005 - Artificial Intelligence

1 (**a**) i. True; UCS is optimal since a priority queue with path cost $g(n)$ is used to order the elements at every step. This ensures that the search expands nodes in the order of their optimal path cost regardless of how the tree looks.

ii. False; In general, the performance of informed search algorithms like A* depends on how well the evaluation function is designed. It is possible for uninformed search algorithms like DFS to be faster than A*. For example, given the tree below:



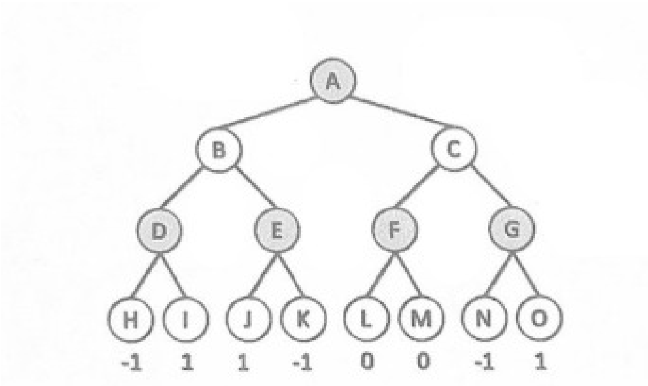
We can see that A* would expand more nodes than DFS. Hence the statement is false.

iii. True; A most constraining heuristic would reduce the branching factor of future choices by selecting the variable that is involved in the largest number of constraints. This contrasts with the least constraining heuristic which leaves maximum flexibility. Thus, this means that dead ends are more likely when variables are more constrained.

iv. False; Every finite game must have a Nash equilibrium. Finiteness is a necessary and sufficient condition for the existence of a Nash equilibrium.

v. True; The strategy for both player A and player B taking action 2 leads to a state wherein it is not possible for either A or B to improve their payout by changing actions. Thus, since neither player has any incentive to deviate from the equilibrium making this a pure strategy.

(b) The tree looks like:



(A, B, C, D, E, F, G) = (1, 1, 0, 1, 1, 0, 1), $\therefore$ the best move is to go to B
<u>*Editor's Note:*</u> *Since we are the "black" player, the objective is to maximise black*

(c) We assume that when a player who is bad at the game comments, he does not know what choices would maximize the reward. For example, when we are at state A, a noob player may lead us to go to C instead of B which is a suboptimal move. At this state C, the opposing player would choose to go to F based on the tree above. Thus, this leads us to get a reward of 0 instead of 1.

If at state A, he chooses B, the opposing player can choose either D or E with equal probabilities since they both have a value of 1. However at this point, since we are able to get either -1 or 1, the comments of the noob player may lead us to move to a state in which the reward is -1 instead of 1. Thus, this affects our play negatively since our reward is lessened.

Assuming the noob player is playing based on chance, the expected reward from listening to his comments is:

$$\frac{1}{4}(-1)+\frac{1}{4}(1)+\frac{1}{2}(0)=0$$

This is lower than the value at A that we have calculated based on the Minimax algorithm above.

2 (**a**) In order to conduct one round of value iteration, we need to calculate the action value of a state-action pair $Q^{\pi}(s,a)$ as well as the value of a state $V^{\pi}(s)$. The formulas are given by:

$$Q_{i+1}(s,a) = \sum_{s'} P(s'|s,a)[R(s,a,s') + \gamma V_i(s')]$$
$$V_{i+1}(s) = \max_a \sum_{s' \in S} P(s'|s,a)[R(s,a,s') + \gamma V_i(s')]$$
$$= \max_a Q_{i+1}(s,a)$$

The Q-table after one iteration would be:

Q_1	U	D	L	R
1	0	0	0	0
2	0.8(0) + 0.1(0) + 0.1(1) = 0.1	0.8(0) + 0.1(0) + 0.1(1) = 0.1	0	0.8(1) + 0.1(0) + 0.1(0) = 0.8
3	0.8(0.8(0.8)) + 0.2(0) = 0.512	0	0.1(0.8(0.8)) + 0.9(0) = 0.064	0.1(0.8(0.8)) + 0.9(0) = 0.064
4	0	0	0	0

Hence the V-table is:

$V_1(s) = \max_a Q_1(s,a)$
0
0.8
0.512
0

(**b**) Based on the observation from the updated table, the optimal policy is:

- 1 - N.A.
- 2 - R
- 3 - U
- 4 - N.A.

(**c**) Since V-learning is similar to Q-learning and we know that the formula for Q-learning is:

$$Q_{new}(S_t, A_t) \leftarrow Q_{old}(S_t, A_t) + \alpha(R_{t+1} + \gamma \max_a Q_{old}(S_{t+1}, a) - Q_{old}(S_t, A_t))$$

Thus, the formula for V-learning is:

$$V_{new}(S_t) \leftarrow V_{old}(S_t) + \alpha(R_{t+1} + \gamma V_{old}(S_{t+1}) - V_{old}(S_t))$$

After the episode above:

- $Q(2,\mathrm{R}) = 0 + 0.1(1 + 0.8(0) - 0) = 0.1$
- $V(2) = 0.1$
- $Q(3,\mathrm{U}) = 0 + 0.1(0 + 0.8(0.1) - 0) = 0.008$
- $V(3) = 0.008$

Hence the V table is:

$V_1(s)$
0
0.1
0.008
0

(d) In order to discourage such samples, we would initially set a baseline policy, we would then keep updating the policy slowly even if the action we have taken for each state has converged such that we would introduce some uncertainties which allows us to explore more of the environment.

We can also set a cutoff k, If a certain episode after k movements do not end at 4, we discard the sample and change the policy and conduct another trial. This allows us to quickly change policies, discard samples without the desired outcome and explore more of the environment.

3 **(a)** i. True

ii. True

iii. True

iv. True

v. True

(b) i. To prove the if:

$$\begin{aligned}(A \Rightarrow \neg B \vee C) &\equiv \neg A \vee (\neg B \vee C)\\ &\equiv (\neg A \vee \neg B) \vee (\neg A \vee C)\\ &\equiv \neg A \vee \neg B \vee C\\ &\equiv \neg(A \wedge B) \vee C\\ &\equiv A \wedge B \Rightarrow C\end{aligned}$$

To prove the reverse direction:

$$\begin{aligned} A \wedge B \Rightarrow C &\equiv \neg(A \wedge B) \vee C \\ &\equiv (\neg A \vee \neg B) \vee C \\ &\equiv (\neg A \vee C) \vee (\neg B \vee C) \\ &\equiv \neg A \vee \neg B \vee C \\ &\equiv \neg A \vee (\neg B \vee C) \\ &\equiv A \Rightarrow \neg B \vee C \end{aligned}$$

Thus, it is a valid statement

A	B	C	¬ B	¬ B ∨ C	A ⇒ ¬ B ∨ C	A ∧ B	A ∧ B ⇒ C	A ⇒ ¬ B ∨ C ⟺ A ∧ B ⇒ C
T	T	T	F	T	T	T	T	T
T	F	T	T	T	T	F	T	T
F	T	T	F	T	T	F	T	T
F	F	T	T	T	T	F	T	T
T	T	F	F	F	F	T	F	T
T	F	F	T	T	T	F	T	T
F	T	F	F	F	T	F	T	T
F	F	F	T	T	T	F	T	T

ii. Since the last columns is always True, we have proven that A ⇒ ¬ B ∨ C ⟺ A ∧ B ⇒ C

(c) i. Every student takes a course

ii. Not all students are smart

(d) Refer to below:

$$\forall x, \text{Student}(x) \wedge \text{Hardworking}(x) \wedge \text{Smart}(x) \Rightarrow \text{Highmark}(x) \tag{1}$$
$$\forall x, \text{Study}(x) \Rightarrow \text{Hardworking}(x) \tag{2}$$
$$\forall x, \text{Clever}(x) \Rightarrow \text{Smart}(x) \tag{3}$$
$$\text{Study(Andy)} \tag{4}$$
$$\text{Clever(Andy)} \tag{5}$$
$$\text{Student(Andy)} \tag{6}$$

From (3) and Universal-Elimination:

$$\text{Clever(Andy)} \Rightarrow \text{Smart(Andy)} \tag{7}$$

From (5, 7) and Modus Ponens:

$$\text{Smart(Andy)} \tag{8}$$

From (2) and Universal-Elimination:

$$\text{Study(Andy)} \Rightarrow \text{Hardworking(Andy)} \tag{9}$$

From (4, 9) and Modus Ponens:

$$\text{Hardworking(Andy)} \tag{10}$$

From (1) and Universal-Elimination:

$$\text{Student(Andy)} \wedge \text{Hardworking(Andy)} \wedge \text{Smart(Andy)} \Rightarrow \text{Highmark(Andy)} \tag{11}$$

From (6, 8, 10) and And-Introduction:

$$\text{Student(Andy)} \wedge \text{Hardworking(Andy)} \wedge \text{Smart(Andy)} \tag{12}$$

From (11, 12) and Modus Ponens:

$$\textbf{Highmark(Andy)} \tag{13}$$

4 **(a)** i. False; $\mu_{\mathbf{A}}(X) \geq 0$

ii. False; By definition, a fuzzy pseudo-partition must satisfy $\sum_{i=1}^{c} \mu_{i,i \in c}(X) = 1$. Thus, if an example i.e. a point along X do not belong to any cluster to any degree this means that the sum at that point is $0 \neq 1$. Hence, it is non pseudo-ly partitioned.

iii. True

iv. False; The number of rules in ANFIS is exponential to the number of partitions i.e. the number of linguistic labels

v. False; The order of TSK models refer to the order of the polynomial by which the rule y is defined.

(b) i. To formulate this problem as an FCSP, we need to define **V, A, X** and **C**.
In this problem, we will define **V** as the set of variables representing the elements of the logo. Hence $\mathbf{V} = \{V_D, V_I, V_S, V_L\}$. **A** is the set of linguistic labels. This would be {red, blue, green}.

The domain of discourse X can be the RGB values for a particular colour and we would have the membership function be a Gaussian membership function based on the values given by the RGB value.

The set of constraints **C** are:

- If V_D is **a** then V_I is not **a**
- If V_D is **a** then V_S is not **a**
- If V_I is **a** then V_D is not **a**
- If V_I is **a** then V_S is not **a**
- If V_S is **a** then V_D is not **a**
- If V_S is **a** then V_I is not **a**
- If V_S is **a** then V_L is not **a**
- If V_L is **a** then V_S is not **a**

ii. In order to solve an FCSP with DFS, we would need to make sure that we are able to transform the continuous values in the domain of discourse into discrete spaces. This means that given a value $x \in \mathbf{X}$, we need to be able to determine a label $\mathbf{a} \in \mathbf{A}$. This can be done by simply taking the label a for x for which $\text{argmax}_{a \in \mathbf{A}} \mu_a(x)$ is the max. Thus, in this way, our continuous space can now be treated as a discrete search space.

During exploration, we would already have initialised a value for V_i, a member of the set **V**. Since we are able to get a label for the value x which we have assigned to V_i. We would then need to generate the next branch in our search tree. Based on the set **C** which gives us the constraints for $V_j \in \mathbf{V}, j \neq i$, we are able to do forward checking and prevent an expansion of nodes which violates any constraints as we are given constraints of the form 'If V_i is **a** then V_j is not **a**'. We would then get values of a for which the constraints are not violated. If the constraints are violated, we do not expand the node, else we get values of x for which V_j is not **a**. Since it is a continuous space, there are many possible values of x, we take the arithmetic mean $\frac{a+b}{2}$ of the set of possible values of x which is denoted by $[a,b]$. This gives us a value of x for a particular **a**.

In order to know if a constraint have been violated, we would have to refer to **D** which is the defuzzification rule. As we assign values to the variables V, we substitute this into D and check if the output is NAN. Since D only returns NAN if its argument is a constant zero function, this means that the constraint has been violated based on the definition of $\sigma(V_i)$ given in the question. Thus, we would only backtrack and stop expanding a particular path if the value returned is NAN.

Editor's Note: Question 4b is open-ended and hence, there are many possible answers. The one provided above is simply the one that I had written during the exam.

Solver: Aurelio Jethro Prahara (aprahara001@e.ntu.edu.sg)