NAMIBIA UNIVERSITY
OF SCIENCE AND TECHNOLOGY

Faculty of Computing and Informatics

Department of Computer Science

QUALIFICATION: Bachelor of Computer Science	
QUALIFICATION CODE: 07BACS	**LEVEL:** 7
COURSE: Artificial Intelligence	**COURSE CODE:** ARI711S
DATE: JUNE 2023	**SESSION:** 1
DURATION: 3 HOURS	**MARKS:** 100

FIRST OPPORTUNITY EXAMINATION QUESTION PAPER	
EXAMINER	MR STANTIN SIEBRITZ
MODERATOR:	MS PAULINA SHIFUGULA

THIS EXAM PAPER CONSISTS OF 7 PAGES
(Including this front page)

INSTRUCTIONS

1. This paper contains 5 questions.
2. Answer ALL questions on the examination booklet provided.
3. Marks/scores are indicated at the right end of each question.
4. Calculators are permitted.
5. NUST examination rules and regulations apply.

QUESTION 1: SEARCH **[20]**

State whether the following statements are either TRUE or FALSE.

1.1 Uniform-cost search will never expand more nodes than A*-search. (2)

1.2 The heuristic $h(n) = 0$ is admissible for every search problem. (2)

1.3 The heuristic $h(n) = 1$ is admissible for every search problem. (2)

1.4 If $h_1(s)$ and $h_2(s)$ are two admissible A* heuristics, then their average $h_3(s) = \frac{1}{2} h_1(s) + \frac{1}{2} h_2(s)$ must also be admissible. (2)

1.5 Depth-first search will always expand more nodes than breadth-first search. (2)

1.6 Depth-first graph search is guaranteed to return an optimal solution. (2)

1.7 Breadth-first graph search is guaranteed to return an optimal solution. (2)

1.8 Uniform-cost graph search is guaranteed to return an optimal solution. (2)

1.9 Greedy graph search is guaranteed to return an optimal solution. (2)

1.10 A* graph search is guaranteed to return an optimal solution. (2)

QUESTION 2: ADVERSARIAL SEARCH **[20]**

Using the α – β pruning technique, solve the adversarial game depicted in Figure 1. Indicate the α – β values at each node and a prune by drawing an [X] on the branch.

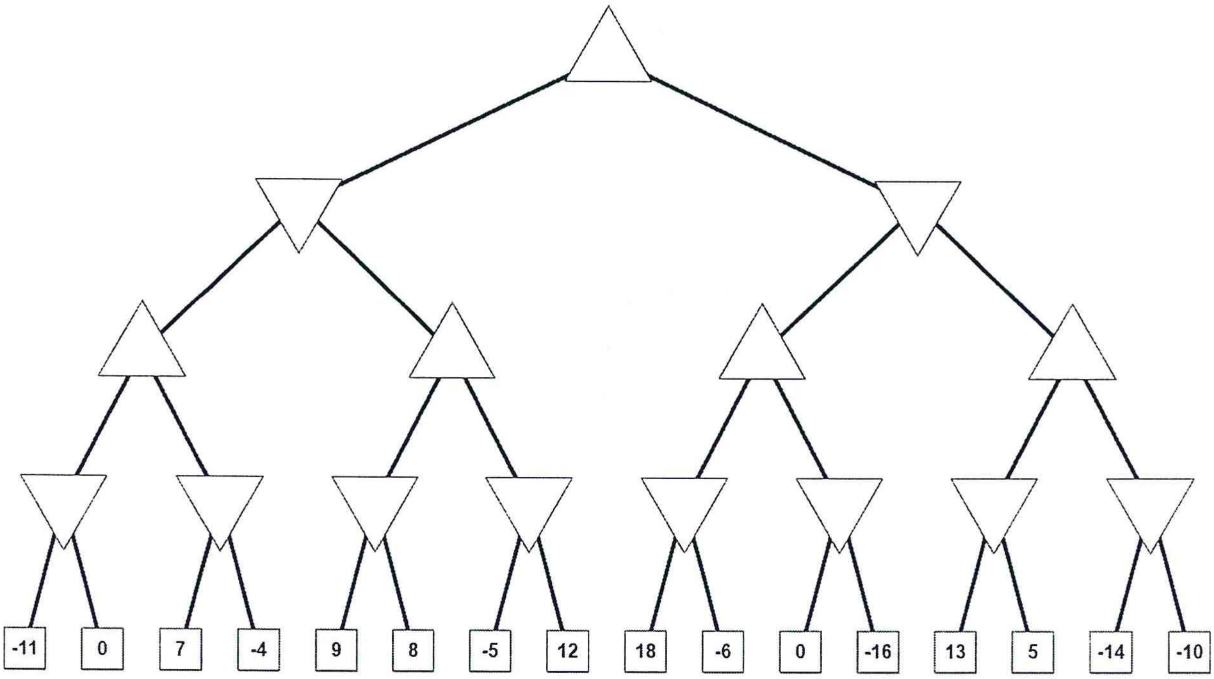


Figure 1: Adversarial Search Problem

QUESTION 3: CSP **[20]**

ARI711S Exams is coming up, and the ARI711S staff have yet to moderate the exam in order to produce the memorandum. There are a total of 6 questions on the exam and each question will cover a topic. Here is the format of the exam:

- Question 1: Search
- Question 2: Games (Adversarial Search)
- Question 3: CSPs
- Question 4: MDPs
- Question 5: True/False
- Question 6: Short Answer

There are 7 people on the course staff: Stantin, Luciano, Paulina, Judy, Kyle, Michael, and Nick.

Each of them is responsible to work with Prof Jose Quenum on one question (but a question could end up having more than one staff member, or potentially zero staff assigned to it). However, the staff are pretty quirky and want the following constraints to be satisfied:

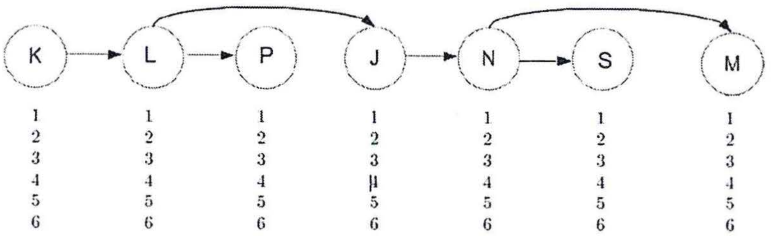
(i) Luciano (L) will not work on a question together with Judy (J).
(ii) Kyle (K) must work on either Search, Games or CSPs
(iii) Michael (M) is very odd, so he can only contribute to an odd-numbered question.
(iv) Nick (N) must work on a question that's before Michael (M)'s question.
(v) Kyle (K) must work on a question that's before Luciano (L)'s question
(vi) Stantin (S) does not like grading exams, so he must work on True/False.
(vii) Judy (J) must work on a question that's after Nick (N)'s question.
(viii) If Stantin (S) is to work with someone, it cannot be with Nick (N).
(ix) Nick (N) cannot work on question 6.
(x) Paulina (P) cannot work on questions 4, 5, or 6
(xi) Luciano (L) cannot work on question 5.
(xii) Luciano (L) must work on a question before Paulina (P)'s question.

3.1 Model this problem as a constraint satisfaction problem (CSP). Variables correspond to each of the staff members, J, P, N, L, M, S, K, and the domains are the questions 1, 2, 3, 4, 5, 6. After applying the unary constraints, what are the resulting domains of each variable? (4)

3.2 If we apply the Minimum Remaining Value (MRV) heuristic, which variable should be assigned first? (2)

3.3 Realizing this is a tree-structured CSP, we decide not to run backtracking search, and instead use the efficient two-pass algorithm to solve tree-structured CSPs. We will run this two-pass algorithm **after** applying the **unary constraints** from part 3.1.

Below is the linearized version of the tree-structured CSP graph for you to work with.



3.3.1 **First Pass: Domain Pruning.** (7)

Pass from *right* to *left* to perform Domain Pruning. Write the values that remain in each domain for each variable.

3.3.2 **Second Pass: Find Solution.** (7)

Pass from *left* to *right*, assigning values for the solution. If there is more than one possible assignment, choose the *highest* value.

QUESTION 4: MDP **[30]**

Consider the blocks world. The blocks can be on a table or in a box. Consider three generic actions: a_0, a_1, and a_2 described as follows:

a_0: when applied to a block, will keep it in the box;
a_1: when applied to a block, will move it on the table;
a_2: when applied to two blocks, will move the first one on top of the second one.

Consider the following four states in the system:
S_0: all blocks are in the box, no block is on the table;
S_1: only block B is on the table; all other blocks are in the box;
S_2: both blocks B and C are on the table, with C on top of B;
S_3: blocks B, C and D are on the table, with D on top of C and C on top of B.

Furthermore, additional information is provided in Table 1, where each state has a reward, possible actions and a transition model for each action. Note that for a given action, the probability values indicated in its transition model all sum up to 1.

State	Reward	Action	Transition
S_0	10	a_{0b}	$(1, S_0)$
		a_{1b}	$(0.25, S_0)$; $(0.75, S_1)$
S_1	20	a_{0c}	$(1, S_1)$
		a_{1c}	$(0.5, S_1)$; $(0.25, S_4)$; $(0.25, S_2)$
		a_{2c}	$(0.6, S_1)$; $(0.4, S_2)$
S_2	30	a_{0d}	$(1, S_2)$
		a_{1d}	$(0.5, S_2)$; $(0.25, S_5)$; $(0.25, S_3)$
		a_{2d}	$(0.6, S_2)$; $(0.4, S_3)$
S_3	100	-	-

Table 1: Additional Information

Assuming we model this problem as Markov Decision Process (MDP) and consider a discount value $\Upsilon = 0.4$

4.1 Provide the Bellman equation variations for policy extraction. (5)

4.2 Extract the policy of each of the states S_0, S_1 and S_2 at k=2 (third iteration) using the policy extraction algorithm. Note that although the states S_4 and S_5 have not been defined, they should be assumed in the system with a reward of 40 each. (20)

4.3 Consider the following policy, $\pi_0 = \{S_0 \rightarrow a_{0b}, S_1 \rightarrow a_{1c}, S_2 \rightarrow a_{2d}\}$. Is π_0 optimal? Explain. (5)

QUESTION 5: REINFORCEMENT LEARNING **[10]**

Recall that reinforcement learning agents gather tuples of the form $< s_t; a_t; r_{t+1}; s_{t+1}; a_{t+1} >$ to update the value or Q-value function. In both of the following cases, the agent acts at each step as follows: with probability 0.5 it follows a fixed (not necessarily optimal) policy π and otherwise it chooses an action uniformly at random.

Assume that in both cases updates are applied infinitely often, state-action pairs are all visited infinitely often, the discount factor satisfies $0 < \Upsilon < 1$, and learning rates are all decreased at an appropriate pace.

5.1 The Q-learning agent performs the following update: (5)

$$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha[r_{t+1} + \gamma \max_a Q(s_{t+1}, a) - Q(s_t, a_t)]$$

Will this process converge to the optimal Q-value function? If yes, write "Yes". If not, give an interpretation (in terms of kind of value, optimality, etc.) of what it will converge to, or state that it will not converge.

5.2 Another reinforcement learning algorithm is called SARSA, and it performs the update: (5)

$$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha[r_{t+1} + \gamma Q(s_{t+1}; a_{t+1}) - Q(s_t, a_t)]$$

Will this process converge to the optimal Q-value function? If yes, write "Yes". If not, give an interpretation (in terms of kind of value, optimality, etc.) of what it will converge to, or state that it will not converge.

END OF EXAMINATION PAPER