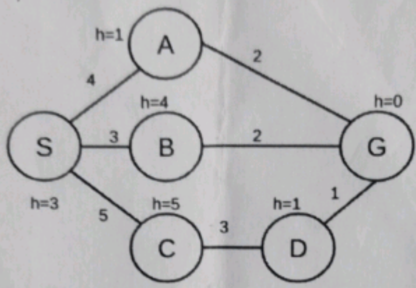
DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
Indian Institute of Technology Jodhpur

September 9, 2023

Artificial Intelligence (CSL7610/CS323)

Time: 1 hour | *Minor 1 Exam* | Maximum Marks: 30

1. Convert the city to city traversal problem given in the figure below to a search problem such that you can apply any uninformed/informed search. Here S is the start city, and G is the target city.

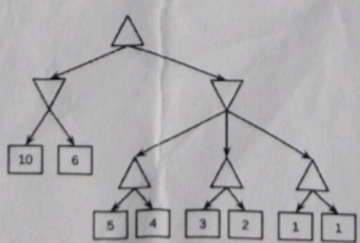


   a) Define all the components of a search problem for this case. [3 Marks]
   b) Apply BFS, DFS and UCS tree search to find the solution. Incase of ties follow alphabetical order. For each approach, show all the intermediate search trees, fringe list and other values as the nodes get expanded and also write the solution sequence of states [6 Marks]
   c) Apply Greedy and A* tree search to find the solution. Incase of ties follow alphabetical order. For each approach, show all the intermediate search trees, fringe list and other values as the nodes get expanded and also write the solution sequence of states. [4 Marks]
   d) Is the solution provided by A* for the above problem optimal? Explain in detail what causes the optimality or non-optimality in this case. [2 Marks]

2. a) Difference between an admissible and a consistent heuristic function? [1 Mark]
   b) Suppose A* uses f(x)=max(g(x),p(x)), where g(x) follows the usual definition. If the path to x from start state s follows s,c,d,x, then p(x)=(h(c)-h(s))+(h(d)-h(c))+(h(x)-h(d)), where h(.) is a consistent and admissible heuristic function. Suppose there is an optimal goal state A and a suboptimal goal state B. Prove that f(A)<f(B). [Donot prove using examples] [2 Marks]
   c) The current algorithm is the same as which other search algorithm and why? [1 Mark]

3. What is the shoulder problem in hill-climbing and how enforced hill-climbing avoids this. [2 Marks]

4. a) Where is alpha-beta pruning used and why? Apply alpha-beta pruning below. Show all the intermediate states and checks involved. Also show its negative effect here [6 Marks]



5. "Expectimax gives the best solution for the MAX player given an optimal MIN player." Explain why or why not? Modify an expectiminimax game tree to get a minimax solution without changing any of the nodes or actions? Show an example game before and after the modification. [3 Marks]

End of exam

CSL7610/CS323 AI Major Exam

Total Marks: 40

Duration: 120 mins

The questions have all the required information available.
Use of pencil is NOT allowed. If pencil is used to answer any question, that answer will not be checked.

1. **[8 marks]** Consider a Bayesian network representing the relationships between weather, traffic, and being late for work. The conditional probability tables are as follows:

W	P(W)
sunny	0.7
rainy	0.3

W	T	P(T|W)
sunny	+t	0.9
sunny	-t	0.1
rainy	+t	0.2
rainy	-t	0.8

W	T	L	P(L|T,W)
sunny	+t	+l	0.9
sunny	+t	-l	0.1
sunny	-t	+l	0.1
sunny	-t	-l	0.9
rainy	+t	+l	0.7
rainy	+t	-l	0.3
rainy	-t	+l	0.3
rainy	-t	-l	0.7

The nodes in the network are Weather (W), Traffic (T), and Late (L). Solve the room allotment problem with the following constraints:

a. Draw the Bayesian Network [2 Marks]

b. What is the probability that it is rainy, there is traffic (+t), and you are late for work (+l) [3 Marks]

c. What is the probability distribution over L with W=sunny, i.e. P(L,W=sunny) [3 Marks]

2. **[5 marks]** Let's consider a Bayesian network (BN) involving five variables: Health (H), Exercise (E), Diet (D), Stress (S), and Heart Disease (A). Suppose, all the conditional probability distributions have been provided to you, namely: P(H), P(E|H), P(D|H), P(S|D,E), P(A|S).

a. Draw the Bayesian Network [2 Marks]

b. What is the name of the algorithm used for checking for conditional independence in a BN [1 Mark]

c. If A=+a is known, check if E & D are conditionally independent using the above method. [2 Marks]

3. **[5 Marks]** Solve the following MDP (described in the table below) using Value Iteration:

Start State	Action	End State	Transition Prob	Reward
A	a	B	0.9	1
A	a	A	0.1	1
A	b	C	0.9	2
A	b	A	0.1	1
B	a	A	0.8	1
B	a	C	0.2	1.5
B	b	B	1	1
C	a	D	0.9	10
C	a	A	0.1	2
C	b	B	1	-1
D	a	D	0	0
D	b	D	1	0

There are 4 states A, B, C, D and 2 actions a and b. The discounting factor is 0.1. Repeat the iterations for either a maximum of 5 iteration steps or as long as the maximum difference between the previous and current expected value of any state is >0.1, whichever comes earlier. **Round off the expected V value obtained after every iteration to 1 decimal position.**

a. Draw the MDP state space with regular and Q nodes [2 Marks]

b. What is the final policy obtained? Show all the steps, and intermediate values. [3 Marks]

4. **[4 Marks]** Solve the following MDP. There are 5 states, A, B, C, D and two actions a and b. The policy is fixed with action a being recommended for all states. The transition probability and reward functions are not available, but the effects of 1 episode of the agent performing actions in the environment have been recorded. In this episode, A goes to B through action a, B goes to C through action a and C goes to D through action a. The rewards obtained are R1, R2, R3, i.e. Episode 1: (A,a,B,R1), (B,a,C,R2), (C,a,D,R3). No transition sample is available from D. Show that finding V(A), V(B), V(C), using Temporal Difference (TD) learning for this single episode is the same as using Direct Evaluation. Use discounting factor=0.1. Also in TD, at each timestep within an episode, the V(A) calculations only looks at the immediate next state after transition which is B in this case. Therefore, repeat the state-wise V calculations in TD till the previous calculation matches the current one. In TD, V(.) is initially 0 for all states. For Direct evaluation note that t starts from 0.

PV(91R)

5. **[2 Marks]** Explain under what common condition, likelihood-weighted sampling and rejection sampling are both same as prior sampling.

6. **[3 Marks]** In the following CSP variable value assignment table, **complete** the assignment of values (V1-9) to variables (R1-9) using backtracking, cycle consistency and MRV.

	V1	V2	V3	V4	V5	V6	V7	V8	V9	Order
R1	V1	x	x	x	x	x	x	x	x	1
R2	x	x	x	V4	x	x	x	x	x	2
R3	x	x	x	x	x	x	V7	x	x	3
R4	x	x		x	x		x	x		
R5	x	V2	x	x	x	x	x	x	x	4
R6	x	x	x	x	V5	x	x	x	x	5
R7	x	x	x	x	x	x	x	V8	x	6
R8	x	x		x	x		x	x		
R9	x	x		x	x		x	x		

In case of a tie between values follow the order V1>V2>V3>V4>V5>V6>V7>V8 and in case of tie between variables follow the order R1>R2>R3>R4>R5>R6>R7>R8>R9. **Some variables are already allotted. Don't change any existing entries.** Don't miss out on the Order column. x denotes values that cannot be assigned to current row (variable). The only constraint that you should look out for is that, adjacent variables should not have adjacent values, e.g., if R2 has V2 then R1 and R3 cannot have V1 or V3. Please Note that R1 and R9 are not adjacent variables and V1 and V9 are not adjacent values. One variable can have only one value and vice versa

7. **[4 Marks]** Verify if the query can be inferred from KB showing **all the steps and rules** using Resolution Refutation. (For inference only Resolution inference rule can be used, Don't use truth values or statements like if A is True then B will also be True) **[Solve any one of the two]**

a) A->(B->C)
(A v C) -> D
Query: C

b) A v C
A->B
C->D
¬B
Query: D

8. **[4 Marks]** Verify if the query can be inferred from KB showing **all the steps and rules** using only PL Natural Deduction (no resolution or truth value). (Don't use truth values or statements like if A is True then B will also be True) **[Solve any one of the two]**

a) P v (Q ∧ R)
(P v R) -> S
Query: S

b) A v (A ∧ B)
¬C->¬A
C->D
Query: D

9. **[5 Marks]** Suppose you have an array of size N=5 and you have to search for a number D. The starting state is the middle position (3). You can start checking from the starting state using DFS with the graph formulation. Each state can only move 1 step to the left or right direction. Once DFS has been completed, use the closed set position of each state to get the heuristic value, i.e., if the final closed set after DFS is S,A,C,B,D, then h(S)=5-1. The cost of moving left or right is 1. Now solve this problem using A* search. **Draw "only" the last tree and "other values" for both methods. Donot miss out nodes that cannot be expanded. Also, write down the final solution for DFS and A*.** Note for this question: In cases where multiple children of a node have to be inserted into the fringe simultaneously and they have the same overall cost depending on the algorithm, insert them in such a way that when removing them from the fringe, the removal is automatically in alphabetical order.

1	2	3	4	5
B	A	S	C	D

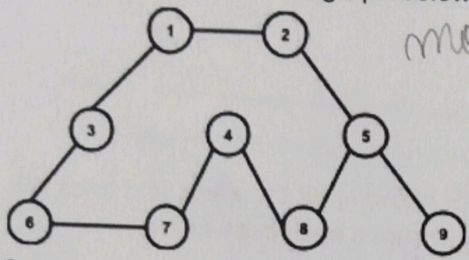
———END———

CSL7610 AI Minor 2

Total Marks: 30 **Duration:** 60 mins

Use of pencil is NOT allowed. If pencil is used to answer any question, that answer will not be checked.

1. **[8 marks]** Consider the following problem: 8 cricketers arrive at a hotel after their match and have to be allotted rooms for their stay. The cricketers include 3 batsmen: B1, B2, B3, 3 bowlers: W1, W2, W3 and 2 all rounders: A1, A2. There are 9 rooms in the hotel that are arranged as shown in the graph below.



more than 2 nodes

	B1	B2	B3	W1	W2	W3	A1	A2	Order
1	B1	X	X	X	X	X	X	X	1
2	X								
3	X								
4	X								
5	X								
6	X								
7	X								
8	X								
9	X								

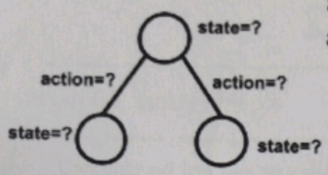
Solve the room allotment problem with the following constraints:
C1: The path between the room/nodes of the same types of cricketers should contain more than 1 node, excluding the start and end nodes, e.g., if W2 gets room 2, then no other bowler can be allotted rooms 1, 3, 5, 8, 9
C2: Room number of B2 > Room number of A1
C3: Room number of B3 > Room number of W2

Solve using CSP with backtracking **a) with only forward checking** and **b) with forward checking, arc consistency, and MRV**. In simple forward checking, assign cricketers to rooms in increasing order of the room/node number (1,2,..). In the table, add X to denote that the current room (row) cannot be allotted to this cricketer (column). If multiple cricketers are available to be allotted to a room, then try to allot that room in the order B1>B2>B3>W1>W2>W3>A1>A2. Fill the above table for both of the methods. If more than one room is equally eligible to be allotted next, follow the increasing order of the room number to allot first. **Stop at the first Failure and mention FAILURE**. Failure can be caused by not having any rooms for a specific cricketer or having more than one room that cannot be allotted to anyone. Room 1 is allotted to B1 by default. **A cricketer can be allotted only one room and one room can have only one cricketer. Order column should mention the order in which you tried to allot the room, even if you were not able to assign any cricketer to that room.**

2. **[7 Marks]** Consider a problem where you have to make two dishes D1 and D2. D1 requires the ingredients T1 and T2, whereas D2 requires the ingredient T3. Initially, none of the ingredients and dishes are available. Model this problem as a planning problem using the STRIPS formulation. The actions available are getT1, getT2, getT3, makeD1, makeD2. getT1 can get the ingredient T1 but this cannot be done if T3 or G1 is already present. getT2 can get the ingredient T2 but can only be performed if T1 is already present. getT3 can get the ingredient T3 but this cannot be done if T1 or G2 is already present. makeD1 requires T1 and T2 to be present and it consumes/removes T1 and T2 to produce D1. makeD2 requires T3 to be present, and it consumes/removes T3 to produce D3.

Continued next page...

D1 & D2



a) Represent all the actions and the starting state in STRIPS.

a) Draw the entire state space graph showing all possible states reachable from the starting state and subsequent states due to any valid action. The diagram should clearly show the STRIPS representation for each state. For showing actions, simply the action name can be mentioned.

Note: Use T1,T2,T3,D1,D3 as the corresponding propositions

3. **[5 Marks]** In each case, reduce to a single term using equivalence laws only and no truth table/resolution

a) $A \Rightarrow B \Leftrightarrow B \wedge A$

b) $((A \Rightarrow B) \wedge (C \Rightarrow B) \wedge C) \Rightarrow B$

4. **[5 Marks]** Verify if the query can be inferred from KB showing all the steps and rules (no resolution). Note: Follow the exact mechanism mentioned above each case below.

<u>using FOL Forward Chaining</u>

a) $\forall y\ \forall z\ P(y,z) \wedge T(y)$

$\forall x\ Q(x)$

$\forall x\ \forall y\ P(x,y) \wedge Q(y) \Rightarrow R(x) \wedge S(x,y)$

Query: S(Pam,Tam)

Assuming Pam,Tam are allowed values

<u>using FOL Backward Chaining</u>

b) $\forall x\ P(x) \wedge R(x) \Rightarrow Q(x)$

$\forall x\ S(x) \Rightarrow P(x)$

$\forall x\ P(x) \Rightarrow R(x)$

S(Pam)

Query: Q(Pam)

5. **[5 Marks]** Verify if the query can be inferred from KB showing all the steps and rules using only PL Natural Deduction (no resolution)

a) $A \wedge B \Rightarrow ((C \wedge D) \vee D)$

$A \wedge D \Rightarrow R$

$A \wedge B$

Query: R

b) $(\neg A \vee \neg C) \Rightarrow B$

$C \Rightarrow E$

$\neg B$

Query:E