The National Higher School of Artificial Intelligence
Introduction to Artificial Intelligence
Final Exam 30/05/2024
Duration: 2 hour 15 minutes

Last Name	
First Name	
Group #	

Exercise 1		**/ 4**
Exercise 2		**/ 5**
Exercise 3		**/ 7**
Exercise 4		**/ 5**
TOTAL		**/21**
		Normalised /20

1. **ANSWER ON THESE EXERCISE STATEMENT PAGES. IF YOU DO NOT HAVE ENOUGH, SPACE WRITE ON THE BACK OF THE SAME PAGE.**
2. **Use a PEN NOT a PENCIL everywhere. (A penalty will be applied if you use a pencil.)**
3. **DO NOT** USE ANY RED OR PINK (OR RELATED) COLOUR IN YOUR ANSWERS**!**
4. **FILL IN ALL THE IDENTIFICATION BOXES OTHERWISE PAGES MAY GET MIXED UP.**
5. **WRITE NEATLY SO AS NOT TO BE PENALISED!**

Exercise 1: **4 points**

During a ceremony, you need to select exactly five pieces of music to play. Your music list is made up of a set of ten musical pieces {M_1, ..., M_{10}}. Six of these are in English (M_1, …, M_6) and four in Arabic (M_7, …, M_{10}). The pieces are categorised into musical styles: Chaabi={M_1, M_2, M_7}, Tergui={M_3, M_8}, Sahraoui={M_4,M_5,M_9} and Chaoui={M_6, M_{10}}.

The person in charge of the ceremony imposes certain constraints that you must respect:
- Two consecutive pieces should not be in the same language and style;
- One piece of each style must be played;
- Piece M_{10} must be played.
- The ceremony should end with a Tergui piece.

1. Formalise this problem in a CSP framework by giving the necessary variables, domains and constraints. The constraints must be written in First-Order Logic.
2. Assuming we assign: $P2=M10$ and $P5=M3$. What would be the solution if we apply the steps of the algorithm "Backtracking search with forward checking".

Solution:

a) Problem Formalization: 2.75pts

Variables 0.25 pt

The variables represent the positions of the musical pieces to be played. There are five variables $P1$, P2,$P3$,$P4$ and $P5$ where each variable represents the musical piece played at that position.

Domains 0.25 pt

The domain of each variable is the set of available musical pieces:
$D(Pi)=\{M1,M2,M3,M4,M5,M6,M7,M8,M9,M10\}$

Constraints 0.5 pt

1. No two consecutive pieces in the same language:
$\forall i\in\{1,2,3,4\}$, language$(Pi)\neq$language$(Pi+1)$

Where language($M1$)=english,…, language ($M6$)= english, langue($M7$)=arabic,…, language ($M10$)=arabic.

2. No two consecutive pieces of the same style: 0.5 pt
$\forall i \in \{1,2,3,4\}$,style($Pi$)≠style($Pi$+1)
Where style($M1$)=,…,style($M10$)=chaoui

3. At least one piece of each style: 0.5 pt
$\exists i \in \{1,2,3,4,5\}, Pi \in$ rock
$\exists i \in \{1,2,3,4,5\}, Pi \in$ jazz
$\exists i \in \{1,2,3,4,5\}, Pi \in$ techno
$\exists i \in \{1,2,3,4,5\}, Pi \in$ alternative

4. Special request from the responsible: 0.5 pt
$\exists i \in \{1,2,3,4,5\}, Pi = M10$

5. Finish with a Tergui piece: 0.25 pt
$P5 \in \{M3, M8\}$

b) Algorithm: *"Backtracking search with forward checking"* 1.25 pts
1) We use unary constraints to restrict domains variables.
2) We use binary constraints for forward checking.
3) During a complete assignment, test the n-ary constraints.

So the remaining assignments are :

The result of unfolding the CSP algorithm would be presented as a CSP Tree or per table by providing steps and explanations (0.5pts).

One of the possible solutions is explained as follows:
• $P1$, $P3$, P4 must respect language and style constraints and include at least one piece of each style.

Therefore,
• $P1$=$M1$ (Chaabi, English)
• $P2$=$M10$ (Chaoui, Arabic)
• $P3$=$M4$ (Sahraoui, English)
• $P4$=$M7$ (Chaabi, Arabic)
• $P5$=$M3$ (Tergui, English)

This solution respects all the constraints:

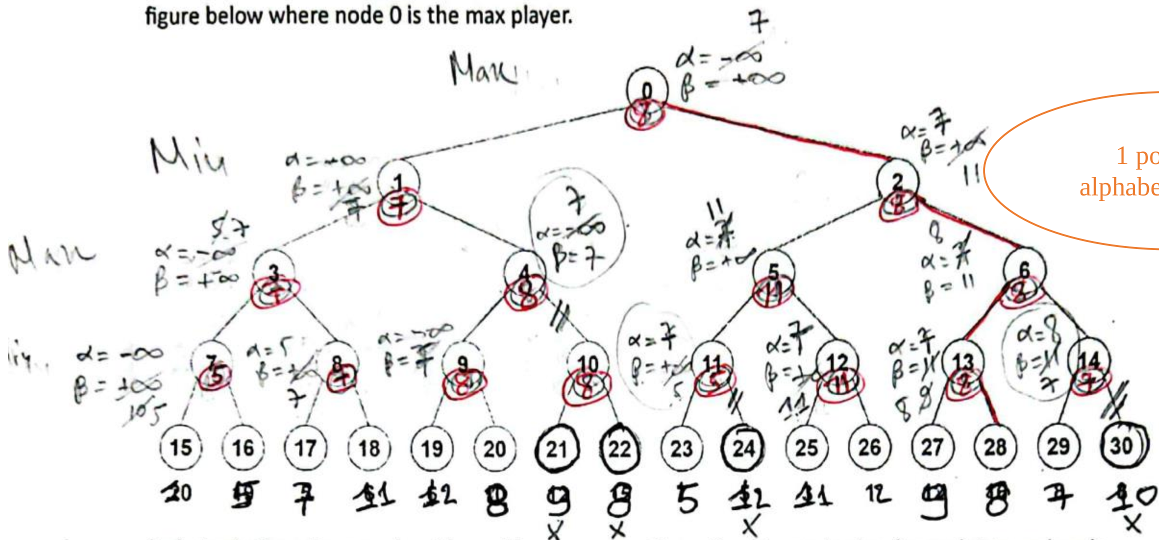
• No two consecutive pieces in the same language.
• No two consecutive pieces of the same style.
• At least one piece of each style.
• The piece M10 is played.
• The last piece is a Tergui piece.

Conclusion
The solution found: $P1$=$M1$, $P2$=$M10$, $P3$=$M4$, $P4$=$M7$, $P5$=$M3$ (0.75 pts)

This sequence respects all the constraints specified in this context.

Exercice 2 : (5 points)

Part (I). Consider an adversarial game with two players acting independently and the winner is unique. Suppose the min-max algorithm is used to play this game and it has produced a search tree shown in the figure below where node 0 is the max player.



1 point for alphabetapruning

(I.a). Apply the min-max algorithm without pruning. Show the returned value for each internal node (node 0 to 14).

Node	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
Value	8	7	8	7	8	11	8	5	7	8	8	5	11	8	7

1 point for minmax

(I.b). Indicate the optimal path as a linked list of nodes (e.g., 0-1-3-7-15) the agent will choose.

Optimal path	0	2	6	13	28

0.5 point for optimal path

(I.c). List all of the nodes that will be pruned in ascending order (you don't need to draw the tree; just list the nodes that will be pruned). Note that this question may require some time to answer thoroughly!

Pruned nodes ids: 21, 22, 24, 30 (note that pruning 21 and 22 are that same as pruning node 10)

part 2 : a) S : 1.2

1 point (0.25 each)

P : 0,4 Q : 1,2 R : 1

b) x_1 x_2 x_3

0.5 point

c) N , O

1 point (0.5 each)

Exercise 3: **7 points**

Part I: Suppose there are three chairs in a row, labeled **L(left)**, **M(middle)**, **R(right)** and three people **A**, **B**, and **C**. Everyone has to sit down but, unfortunately, **(1)** A doesn't want to sit next to B, **(2)** A doesn't want to sit on the left chair, **(3)** C doesn't want to sit to the (adjacent) right of B, **(4)** B doesn't want to sit in the middle, and **(5)** Each person sits on exactly one chair.

1- Formulate these constraints in propositional logic using only variable $\mathbf{X_{p,c}}$ to mean that person **p** sits on chair **c**. Express the constraints in CNF.

2- Using the Resolution method, check whether A can sit on the Right chair.

Solution:

Variables

$X_{A,L}$: Person A sits in the left chair.
$X_{A,M}$: Person A sits in the middle chair.
$X_{A,R}$: Person A sits in the right chair.
$X_{B,L}$: Person B sits in the left chair.
$X_{B,M}$: Person B sits in the middle chair.
$X_{B,R}$: Person B sits in the right chair.
$X_{C,L}$: Person C sits in the left chair.
$X_{C,M}$: Person C sits in the middle chair.
$X_{C,R}$: Person C sits in the right chair.

Constraints

1. A doesn't want to sit next to B:

C1 : $\sim X_{A,L} \lor \sim X_{B,M}$,
C2 : $\sim X_{A,M} \lor \sim X_{B,L}$,
C3 : $\sim X_{A,M} \lor \sim X_{B,R}$,
C4 : $\sim X_{A,R} \lor \sim X_{B,M}$

2. A doesn't want to sit in the left chair:

C5 : $\sim X_{A,L}$

3. B doesn't want to sit in the middle chair

C6 : $\sim X_{B,M}$

4. C doesn't want to sit to the (adjacent) right of B:

C7 : $\sim X_{B,L} \lor \sim X_{C,M}$,
C8 : $\sim X_{B,M} \lor \sim X_{C,R}$,

5. Each person sits in exactly one chair

C9 : $X_{A,L} \lor X_{A,R} \lor X_{A,M}$,
C10 : $X_{B,L} \lor X_{B,R} \lor X_{B,M}$,
C11 : $X_{C,L} \lor X_{C,R} \lor X_{C,M}$,
C12 : $\sim X_{A,L} \lor \sim X_{A,M}$,
C13 : $\sim X_{A,L} \lor \sim X_{A,R}$,
C14 : $\sim X_{A,M} \lor \sim X_{A,R}$,
C15 : $\sim X_{B,L} \lor \sim X_{B,M}$,
C16 : $\sim X_{B,L} \lor \sim X_{B,R}$,
C17 : $\sim X_{B,M} \lor \sim X_{B,R}$,
C18 : $\sim X_{C,L} \lor \sim X_{C,M}$,
C19 : $\sim X_{C,L} \lor \sim X_{C,R}$,
C20 : $\sim X_{C,M} \lor \sim X_{C,R}$,

CNF Form (1pt)

[C1 Λ C2 Λ C3 Λ C4 Λ
C5 Λ
C6 Λ
C7 Λ C8 Λ
C9 Λ C10 Λ C11 Λ C12 Λ C13 Λ C14 Λ C15 Λ C16 Λ C17 Λ C18 Λ C19 Λ C20 Λ C21]

III) Resolution: (2pt)

We will apply resolution to these clauses iteratively.
Initial Clauses
C1 to C20

Steps of the Resolution Method:
C21 = $\sim X_{A,R}$

Step 1:

Resolve C21 $\land$ C9

C21:$\sim X_{A,R}$

C9:($X_{A,L}$ **V** $X_{A,R}$ **V** $X_{A,M}$)

New clause:

C22: ($X_{A,L}$ V $X_{A,M}$)

Step 2:

Resolve C4 $\land$ C22

C5: $\sim X_{A,L}$

C22: ($X_{A,L}$ **V** $X_{A,M}$)

New clause:

C23: $X_{A,M}$

Step 3:

Resolve C2 $\land$ C10

C2: ($\sim X_{A,M}$ V $\sim X_{B,L}$)

C10: ($X_{B,L}$ **V** $X_{B,R}$ **V** $X_{B,M}$)

New clause:

C24: ($\sim X_{A,M}$ V $X_{B,M}$ V $X_{B,R}$)

Step 4:

Resolve C3 $\land$ C24

C3: ($\sim X_{A,M}$ V $\sim X_{B,R}$)

C24: ($\sim X_{A,M}$ **V** $X_{B,M}$ **V** $X_{B,R}$)

New clause:

C25: ($\sim X_{A,M}$ V $X_{B,M}$)

Step 5:

Resolve C6 $\land$ C25

C6: ($\sim X_{B,M}$)

C25: ($\sim X_{A,M}$ V $X_{B,M}$)

New clause:

C26: $\sim X_{A,M}$

Step 6:

Resolve C14 C26

C14: $X_{A,M}$

C26: $\sim X_{A,M}$

New clause:

C27 : {}

<u>Part II:</u>

1. Translate the following sentences into for the First-Order Logic sentences with an appropriate encoding:

a) There are three children Ali, Boudour and Laith.
(Hint: you must explicitly ensure that different names refer to distinct individuals).
b) Every child is wearing either a hat, coat or a scarf.
c) Ali is wearing a red hat.
d) Boudour is wearing a green coat.
e) Laith is wearing a blue scarf.

f) Ali has a favourite colour, which is either red, green, or blue.
g) Laith's favourite colour is green.
h) Boudour's favourite colour is red.

Solution

1) First-order logic encodings for the given statements:

a) $\exists x\ \exists y\ \exists z\ (x \neq y \wedge x \neq z \wedge y \neq z \wedge \text{Child}(x) \wedge \text{Child}(y) \wedge \text{Child}(z) \wedge \text{Name}(x,\text{Ali}) \wedge \text{Name}(y,\text{Boudour}) \wedge \text{Name}(z,\text{Laith}))$
0.75pt

b) $\forall x\ (\text{Child}(x) \rightarrow (\text{Wearing}(x,\text{Hat}) \vee \text{Wearing}(x,\text{Coat}) \vee \text{Wearing}(x,\text{Scarf})))$ 0.25pt

c) Wearing(Ali,RedHat) 0.25pt

d) Wearing(Boudour,GreenCoat) 0.25pt

e) Wearing(Laith,BlueScarf) 0.25pt

f) ([FavoriteColor(Ali,Red) ∧¬FavoriteColor(Ali,Green) ∧ ¬FavoriteColor(Ali,Blue)]
∨[FavoriteColor(Ali,Green) ∧¬FavoriteColor(Ali,Red) ∧
¬FavoriteColor(Ali,Blue)]∨[FavoriteColor(Ali,Blue) ∧¬FavoriteColor(Ali,Green) ∧
¬FavoriteColor(Ali,Red)]) 0.75pt

g) FavoriteColor(Laith,Green) 0.25pt

h) FavoriteColor(Boudour,Red) 0.25pt

2) To infer whether "No child (Laith and Boudour)is wearing anything that has his/her favorite color" using forward chaining, we need to derive this statement from the given premises. We can attempt to do this by considering each child and his/her favorite color: 1pt

From g, e:
favorite color (Laith, Green) ∧ Wearing(Laith, Blue)
→ ¬wearing (Laith, Green) ... (1)
From h, d:
favorite color (Boudour, Red) ∧ wearing (Boudour, Green)
→ ¬wearing (Boudour, Red) ... (2)
From (1) and (2) we get:
¬Wearing (Laith, Green) ∧ ¬wearing (Boudour, Red)
So, we can conclude that no child (Laith and Boudour) is wearing anything that has their favorite color.

Exercise 4: **(5 points)**
Consider a maze with only four possible positions, numbered 1 through 4 as shown in the following diagram. Position 2 is the start position (denoted S in the diagram below), while positions 1, 3, and 4 each contains a goal (denoted G_1, G_2, and G_3 in the diagram). Search terminates when the agent finds a path that reaches all three goals, using the smallest possible number of steps. (The only allowed moves are vertically or horizontally, where allowed. The black cells are not reachable.)

1 G_1	2 S	3 G_2
(black)	4 G_3	(black)

a) Define a notation for the state of this agent. How many distinct non-terminal states are there? **(1 point)**

Answer:

The state can be defined by a pair of variables: (P,G) where $P \in \{1, 2, 3, 4\}$ specifies the current position, and $G \in \{\emptyset, G_1, G_2, G_3, G_1G_2, G_1G_3, G_23,\}$ specifies which of the goals have been reached. There is only one position (2) that can be reached without touching any goal. After touching G_1, there are two positions that can be reached without touching another goal (1 and 2). After touching G_1, and G_2, there are three positions that can be reached without touching G_3, (1, 2, and 3). Generalizing, there are a total of 1 + (3*2) + (3*3) = 16 non-terminal states.

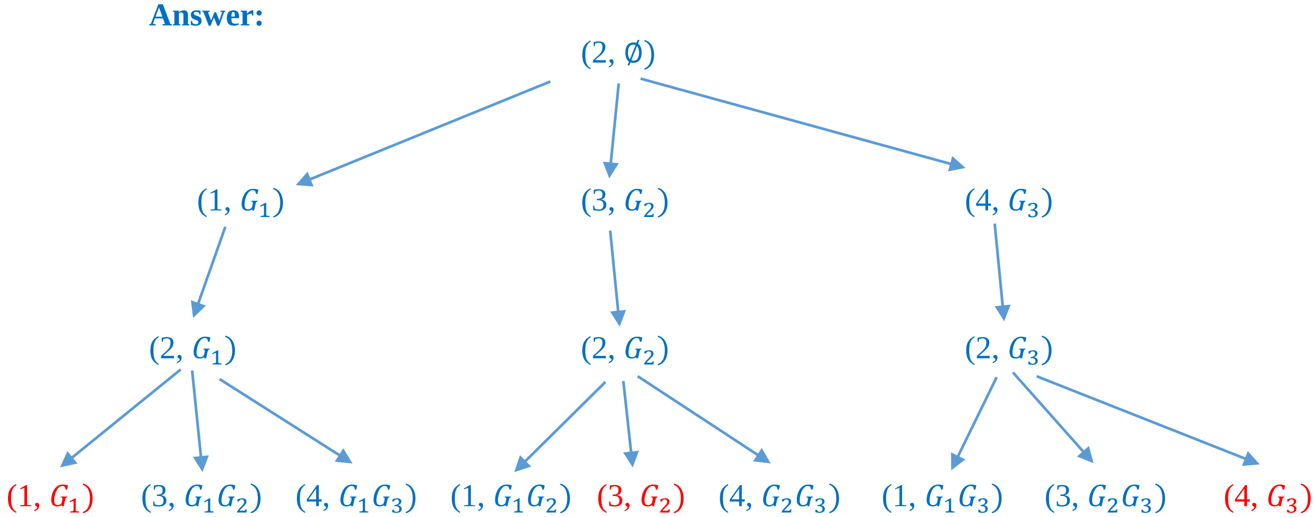
These states are: $(2, \emptyset), (2, G1), (2, G2), (2, G3), (1, G1), (3, G2), (4, G3), (1, G1G2), (1, G1G3), (3, G1G2)$ (3,G2G3),(4,G1G3),(4,G2G3),(2,G1G2),(2,G1G3),(2,G2G3)

Obviously, the students may have a slightly different representation, e.g. having 3 boolean values that indicate with a 1 which goal has been visited.

Many students missed even the logic of what a state should contain as information for the search to be able to proceed from node/state to node/state.

b) Draw a search tree out to a depth of 3 moves, including repeated states. Circle the repeated states. **2 points**

Answer:



The tree must make use of nodes that contain information about the visited goals so as to make the continuation of the search possible. I have given 0.5 points out of 1.5 if the students just showed the tree (basically, they all got the overall shape of the tree) but missed the information about the visited goals in the state (the large majority in what I have marked so far have missed this).
Obviously, you can judge case by case how to assess the quality of the answer and whether the student approaches the correct logic of search.

c) For A* search, one possible heuristic, h_1, is the Manhattan distance from the agent to the nearest goal that has not yet been reached. Prove that h_1 is consistent. **(1 point)**

Answer:
From any node *m*, Manhattan distance to the nearest goal is always either $h_1[m] = 1$ or $h_1[m] = 2$. If $h_1[m] = 2$, then any step we take will move us to a node *n* such that $h_1[n] = 1$. If $h_1[m] = 1$, then it is possible to move away from the goal (to position *n* such that $h_1[n] = 2$), or it is possible to move toward the goal (in which case we reach the goal, and so the definition of "nearest goal that has not been reached" changes to one of the other goals, and again we have $h_1[n] = 2$). So the change in heuristic is always $h_1[m] - h_1[n] \in \{-1, 1\}$.
The distance travelled from any node *m* to its neighbour *n* is always one step. If *n* is closer to completing the maze than *m*, then the distance from *m* to the goal, d[*m*], minus the distance from *n* to the goal, $d[n]$, is one: $d[m] - d[n] = 1$, whereas $h_1[m] - h_1[n] \in \{-1, 1\}$,
so $d[m] - d[n] \geq h_1[m] - h_1[n]$. Since, by definition, $d[m] - d[n]$ is the cost of moving from *m* to *n*, then $h_1[m] \leq h_1[n] + c(m,n)$. So h_1 is consistent.

d) Another possible heuristic is based on the Manhattan distance M[N,G] between two positions, and is given by $h_2[n] = M[G_1,G_2] + M[G_2,G_3] + M[G_3,G_1]$, that is, h_2 is the sum of the Manhattan distances from goal 1 to goal 2, then to goal 3, then back to goal 1. Prove that h2 is not admissible. **(0.5 point)**

Answer:
Notice that $h_2[n] = 6$ for every node *n*, so we only have to find a counter-example for which the total cost of the best path is $d[n] < 6$. But that is easy: the starting node S has a cost of $d[S] = 5$ which is less than $h_2[S]$, so h_2 is not admissible.

e) Prove that $h_2[n]$ is dominant to $h_1[n]$. **(0.5 point)**
Answer:
$h_1[n] \in \{1, 2\}$, whereas $h_2[n] = 6$ for any *n*, so $h_2[n] \geq h_1[n]$. So h_2 dominates h_1.