### ***Course: CS420 - Artificial Intelligence***

### ***Class 20CTT – Term I/2022-2023***

ONLINE EXERCISE

**Question 01.** (Adapted from Prof. Ziv-Bar Joseph, Carnegie Mellon University, Course 10-701 Machine Learning materials.) NASA wants to discriminate Martians (M) from Humans (H) based on these features (attributes): Green ∈ {𝑁, 𝑌}, Legs ∈ {2, 3}, Height ∈ {𝑆, 𝑇}, Smelly ∈ {𝑁, 𝑌}. Your available training data is as follows (N = No, Y = Yes, S = Small, T = Tall). Note that it is

| # | Height | Green | Legs | Smelly | Target: Species |
| --- | --- | --- | --- | --- | --- |
| 1 | S | Y | 3 | Y | M |
| 2 | T | Y | 3 | N | M |
| 3 | S | Y | 3 | N | M |
| 4 | T | Y | 3 | N | M |
| 5 | T | N | 2 | Y | M |
| 6 | T | Y | 2 | Y | H |
| 7 | S | N | 2 | N | H |
| 8 | T | N | 3 | N | H |
| 9 | S | N | 3 | N | H |
| 10 | T | N | 3 | N | H |

Build an ID3 decision tree classifier from the above training dataset. Attributes are evaluated using Information Gain. Ties are broken such that an attribute with earlier alphabetical order is preferred.

1. Present the calculations required to choose the attribute for the root node

|  | Target | Height | | Green | | Legs | | Smelly | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| S | T | Y | N | 2 | 3 | Y | N |
| H | 1 | 1 | 1 | 0.722 | 0.722 | 0.918 | 0.985 | 0.918 | 0.985 |
| AE |  | 1 | | 0.722 | | 0.965 | | 0.965 | |
| IG | 0 | | 0.278 | | 0.035 | | 0.035 | |

The root is **\_\_\_\_\_\_\_\_\_\_\_\_\_\_Green\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

1. Present the calculations required to choose the attribute for the root’s left branch

| Green YES | Target | Height | | Legs | | Smelly | |
| --- | --- | --- | --- | --- | --- | --- | --- |
| S | T | 2 | 3 | Y | N |
| H | 0.722 | 0 | 0.918 | 0 | 0 | 1 | 0 |
| AE |  | 0.551 | | 0 | | 0.4 | |
| IG |  | 0.171 | | 0.722 | | 0.322 | |

The left child of the root is **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_Legs\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

1. Present the calculations required to choose the attribute for the root’s right branch

| Green NO | Target | Height | | Legs | | Smelly | |
| --- | --- | --- | --- | --- | --- | --- | --- |
| S | T | 2 | 3 | Y | N |
| H | 0.722 | 0 | 0.918 | 1 | 0 | 0 | 0 |
| AE |  | 0.551 | | 0.4 | | 0 | |
| IG |  | 0.171 | | 0.322 | | 0.722 | |

The right child of the root is **\_\_\_\_\_\_\_\_\_\_\_\_Smelly\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

1. Draw the complete ID3 decision tree

**Question 02.** From the sentence "Heads I win, tails you lose," prove that "I win" by using propositional logic refutation resolution.

1. Represent the given sentence in propositional logic using only the following prepositions: Head , Tail (the coin’s tail), IWin (I win), and YouLose (you lose)
2. Add some general knowledge axioms about coins, winning, and losing
3. Convert the propositional logic sentences to their CNF equivalents
4. Apply propositional refutation resolution to prove the above conclusion.

a) Head => IWin

Tail => YouLose

b)

Head ∨ Tail

IWin => YouLose

IWin => ¬Tail

YouLose => ¬Head

YouLose => IWin

c) 1. ¬Head ∨ IWin

2. ¬Tail ∨ YouLose

3. Head ∨ Tail

4. ¬IWin ∨ YouLose

5. ¬IWin ∨ ¬Tail

6. ¬YouLose ∨ ¬Head

7. ¬YouLose ∨ IWin

8. ¬IWin

d)

9. ¬YouLose from 7, 8

10. ¬Tail from 9, 2

11. Head from 10, 3

12. IWin from 1, 11

13. • from 8, 12

**Question 03.** Consider the following KB.

| 1. Buffalo(x) ∧ Pig(y) → Faster(x,y) 4. Buffalo(Bob)  2. Pig(y) ∧ Slug(z) → Faster (y,z) 5. Pig(Pat)  3. Faster(x,y) ∧ Faster (y, z) → Faster(x,z) 6. Slug(Steve) |
| --- |

Use first-order logic forward chaining to prove **Faster(Bob, Steve)**. If several rules apply, use the one with the smallest number. Show the chaining process step by step, using the numbering of the sentences to identify how you are using the rules and facts in the KB. For either presentation method, you will need to **indicate the unifications**.

| 7. Faster(Bob, Pat) from 1, 4, 5. θ = {x/Bob, y/Pat}  8. Faster(Pat, Steve) from 2, 5, 6. θ= {y/Pat, z/Steve}  9. Faster(Bob, Steve) from 7, 8, 3. Θ = {x/Bob, y/Pat, z/Steve}  Thus KB entails Faster(Bob, Steve) |
| --- |

**Question 04.** Consider the following English sentences

1. Every child loves Santa.

2. Everyone who loves Santa loves any reindeer.

3. Rudolph is a reindeer, and Rudolph has a red nose.

4. Anything which has a red nose is weird or is a clown.

5. No reindeer is a clown.

6. Scrooge does not love anything which is weird.

1. Build a FOL knowledge base from the above sentences, using only given predicates.

CHILD(x): “x is a child” LOVES(x, y): “x loves y”

REINDEER(x): “x is a reindeer” REDNOSE(x): “x has red nose”

WEIRD(x): “x is weird” CLOWN(x): “x is a clown”

. 1. ∀x CHILD(x) ⇒ LOVES(x, Santa)

2. ∀x LOVES(x, Santa) ⇒ [∀y REINDEER(y) ⇒ LOVES(x, y)]

3. REINDEER(Rudolph) ∧ REDNOSE(Rudolph)

4. ∀x REDNOSE(x) ⇒ [WEIRD(x) ∨ CLOWN(x)]

5. ∀x REINDEER(x) ⇒ ¬CLOWN(x)

6. ∀x WEIRD(x) ⇒ ¬LOVES(Scrooge, x)

1. Convert the propositional logic sentences to their CNF equivalents

1. ¬CHILD(x) ∨ LOVES(x, Santa)

2. ¬LOVES(x, Santa) ∨ ¬REINDEER(y) ∨ LOVES(x, y)

3. REINDEER(Rudolph) ∧ REDNOSE(Rudolph)

4. ¬REDNOSE(x) ∨ WEIRD(x) ∨ CLOWN(x)

5. ¬REINDEER(x) ∨ ¬CLOWN(x)

6. ¬WEIRD(x) ∨ ¬LOVES(Scrooge, x)

1. Apply propositional refutation resolution to prove that “Scrooge is not a child.”

7. CHILD(Scrooge): negation of conclusion

—

8. WEIRD(Rudolph) ∨ CLOWN(Rudolph) from 3b and 4. θ = {x/Rudolph}

9. WEIRD(Rudolph) ∨ ¬REINDEER(Rudolph) from 8 and 5. θ = {x/Rudolph}

10. WEIRD(Rudolph) from 9, 3a

11. ¬LOVES(Scrooge, Rudolph) from 10, 6. θ = {x/Rudolph}

12. ¬LOVES(Scrooge, Santa) ∨ ¬REINDEER(Rudolph) from 11, 2. θ = {x/Scrooge, y/Rudolph}

13. ¬CHILD(Scrooge) ∨ ¬REINDEER(Rudolph) from 12, 1. θ = {x/Scrooge}

14. ¬REINDEER(Rudolph) from 13, 7

15. • from 14, 3a

Thus, KB entails “Scrooge is not a child.”

**Question 05.** Rewrite the following sentence in first-order logic

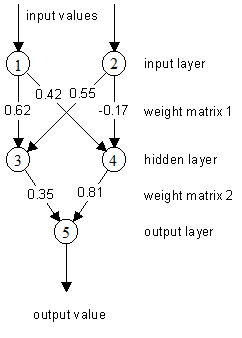
Politicians can fool some people all of the time, and they can fool all people some of the time, but they can’t fool all of the people all of the time.

using only the predicates given below.

Politician(x): x is a politician Person(x): x is a person

Time(x): t is a time period Fool(x, y, t): x fools y at time t

∀x Politician(x) ⇒ [[∃yPerson(y) ∧ [∀z Time(z) ⇒ Fool(x, y, t)]] ∧ [∀p Person(p) ⇒ [∃k Time(k) ∧ Fool(x, p, k)]] ∧ ¬[∀zPerson(z) ⇒ [∀m Time(m) ⇒ Fool(x, z, m)]]]

**Problem 06.** Consider the multi-layer neural network shown below. 

The network parameters setting is as follows.

* The set of weights are

w1 = 0.62 w2 = 0.42 w3 = 0.55 w4 = -0.17 w5 = 0.35 w6 = 0.81

* There is no bias. There is no threshold.
* Learning rate is set to 0.25.
* Sigmoid activation function

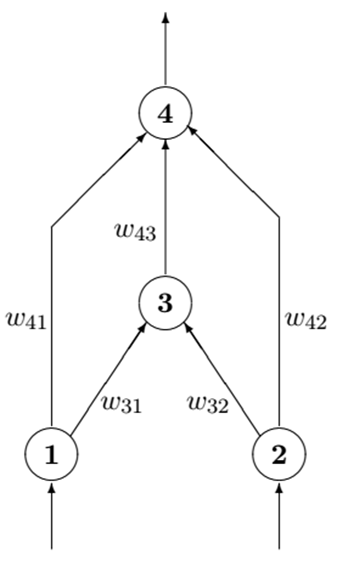
1. Compute the output values for all the hidden and output neurons when input signals come to neuron 1 and 2 are **both 1s** and output signal is 1.

| Neuron | 3 | 4 | 5 |
| --- | --- | --- | --- |
| Output | 0.763 | 0.562 | 0.673 |

1. Adjust the weights according to the computed values above.

| Weights | w1 | w2 | w3 | w4 | w5 | w6 |
| --- | --- | --- | --- | --- | --- | --- |
| Values | 0.621 | 0.424 | 0.551 | -0.167 | 0.364 | 0.820 |

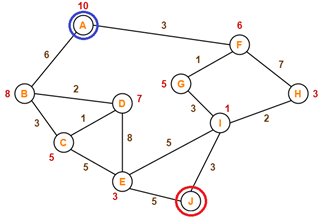
**Question 07.** In the network shown below, all the units have binary inputs (0 or 1), unipolar step functions and binary outputs (0 or 1). The weights for this network are w31 = 1, w32 = 1, w41 = 1, w42 = 1 and w43 = −2. The threshold of the hidden unit (3) is 1.5 and the threshold of the output unit (4) is 0.5. The threshold of both input units (1 and 2) is 0.5, so the output of these units is the same as the input.



Which Boolean functions can be computed by this network? Justify your answer by showing detailed calculations.

| 1 | 2 | 3 | 4 |
| --- | --- | --- | --- |
| 0 | 0 | 0 | 0 |
| 0 | 1 | 0 | 1 |
| 1 | 0 | 0 | 1 |
| 1 | 1 | 1 | 0 |

**Question 08.** Consider the following graph. The initial state is marked with a BLUE circle, and the goal state is marked with a RED circle. Ties are broken in alphabetical order.

****

For each of the following search strategies, state the order in which states are expanded and the path returned. Vertices should be presented in their exact order, and they are separated by a single space, (e.g., S A B C)

Note that the path returned will not be accepted if the list of expanded states is wrong.

Breadth-first search

List of expanded states: A B F C D G H E Path returned: A B C E J

Uniform cost search:

List of expanded states: A F G B I D C H J Path returned: A F G I J

Depth-first search:

List of expanded states: A B C D E Path returned: A B C D E J

Greedy best-first search:

List of expanded states: A F H I Path returned: A F H I J

Graph-search A\*:

List of expanded states: A F G I J Path returned: A F G I J

| **Question 09.** Consider the 4-bishops problem. Every state of the problem has 4 bishops on the board, each of which is in a separate column.  Answer the following questions: |  |
| --- | --- |

The total number of states in the state space is:

| 4^4 = 256 |
| --- |

Each step of the search moves a bishop within its own column. How many successors can a state generate?

| 3 x 4 = 12 |
| --- |

Each state of the problem can be represented in the genetic algorithm as 4 digits, each indicating the position of a bishop in that column. For example, S = 4213.

Let **nb** be the number of attacking pairs of bishops of state n.

Define the fitness function for a state n:

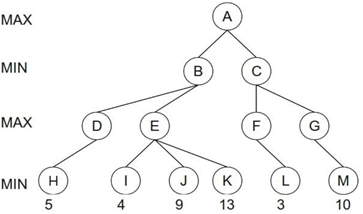
| **f(n) = 4C2 - nb** |
| --- |

The current generation includes 4 states: S1 = 2341; S2 = 2132; S3 = 1232; S4 = 4321.

Calculate the value of Fit(n) for each of the 4 states and the probability that each of them will be chosen in the “selection” step.

| State n | S1 | S2 | S3 | S4 |
| --- | --- | --- | --- | --- |
| Fit(n) | **6-4** | **6-2** | **6-4** | **6-6** |
| Prob(n) | **2/8 = 0.25** | 0.5 | 0.25 | 0 |

**Question 10.** Consider the following game tree. Assume that the root node corresponds to the MAX player and the search always visits children left-to-right.



Compute the final backed-up computed by minimax algorithm. (No alpha-beta pruning at this step)

| A | B | C | D | E | F | G |
| --- | --- | --- | --- | --- | --- | --- |
| 5 | 5 | 3 | 5 | 13 | 3 | 10 |

Compute the final backed-up computed by alpha-beta pruning. If a node is pruned, mark X.

| A | B | C | D | E | F | G |
| --- | --- | --- | --- | --- | --- | --- |
| 5 | 5 | 3 | 5 | 9 | 3 | X |

Using the minimax calculations from part a), without performing any alpha-beta calculation, rotate the children of each node in the above tree at every level to ensure maximum alpha-beta pruning. Fill in the nodes with the letter of the corresponding node. Draw the new edges

|  |
| --- |

**Question 11.** *The 8-puzzle problem*. Apply the hill-climbing algorithm with Manhattan distance heuristic to find a solution for the following pair of initial and goal states.

| Initial state  2 8 3  1 6 4  7 - 5 | Goal state  1 2 3  8 - 4  7 6 5 |
| --- | --- |

Your work should address the following requirements

- Draw the search tree including all possible successors of expanded states (except the goal)

- Calculate the heuristic value for every node

- Mark the optimal strategy found

| **Question 12.** Consider the problem of coloring the six regions (numbered 1...6) in the following map using three colors: R, G and B, so that no adjacent regions have the same color. Two regions are adjacent if they share part of an edge (note: they are NOT adjacent if they only share a corner). |  |
| --- | --- |

If initially every variable has all three possible values except region 1 has known value R (1 = R) and region 2 has known value G (2 = G). What is the result of the Forward Checking algorithm for this step?

| Variables | 1 | 2 | 3 | 4 | 5 | 6 |
| --- | --- | --- | --- | --- | --- | --- |
| Final domains | **R** | **G** | G B | B | R B | B |

Assume the initial domains of the regions in the map above are given as 1 = {R, G, B}, 2 = {R, G}, 3 = {R, G, B}, 4 = {R}, 5 = {R, G, B}, and 6 = {R}. What is the result of applying the Arc Consistency algorithm, AC-3, starting at Region 1?

| Variables | 1 | 2 | 3 | 4 | 5 | 6 |
| --- | --- | --- | --- | --- | --- | --- |
| Final domains | G B | G | R G B | R | R G B |  |

Assume no variables have been assigned yet, solve the CSP using backtracking with forward checking. Ties (after considering all necessary heuristics) are resolved by numeric ordering (e.g., if both region 1 and region 2 are possible, choose region 1).

For every step, present the *MRV values for all regions that are not colored yet*. If there are many *regions that have the same minimum MRV*, present the DH values for these regions.

Step 1

| Variables | 1 | 2 | 3 | 4 | 5 | 6 |
| --- | --- | --- | --- | --- | --- | --- |
| MRV | R G B | R G B | R G B | R G B | R G B | R G B |
| DH | 3 | 3 | 2 | 3 | 2 | 3 |

Color the region \_\_\_\_\_\_\_1\_\_\_\_\_\_\_\_\_\_\_\_\_ with the color \_\_\_\_\_\_\_R\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Step 2

| Variables | 1 | 2 | 3 | 4 | 5 | 6 |
| --- | --- | --- | --- | --- | --- | --- |
| MRV |  | R G B | G B | G B | R G B | G B |
| DH |  |  | 1 | 2 |  | 2 |

Color the region \_\_\_\_\_\_4\_\_\_\_\_\_\_\_\_\_\_\_\_\_ with the color \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_G\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Step 3

| Variables | 1 | 2 | 3 | 4 | 5 | 6 |
| --- | --- | --- | --- | --- | --- | --- |
| MRV |  | R B | G B |  | R G B | B |
| DH |  |  |  |  |  |  |

Color the region \_\_\_\_\_\_\_6\_\_\_\_\_\_\_\_\_\_\_\_\_ with the color \_\_\_\_\_\_\_\_B\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Step 4

| Variables | 1 | 2 | 3 | 4 | 5 | 6 |
| --- | --- | --- | --- | --- | --- | --- |
| MRV |  | R | G B |  | R G B |  |
| DH |  |  |  |  |  |  |

Color the region \_\_\_\_\_\_\_2\_\_\_\_\_\_\_\_\_\_\_\_\_ with the color \_\_\_\_\_\_\_\_\_R\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Step 5

| Variables | 1 | 2 | 3 | 4 | 5 | 6 |
| --- | --- | --- | --- | --- | --- | --- |
| MRV |  |  | G B |  | G B |  |
| DH |  |  | 1 |  | 1 |  |

Color the region \_\_\_\_\_\_3\_\_\_\_\_\_\_\_\_\_\_\_\_\_ with the color \_\_\_\_\_\_\_\_\_G\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Step 6

Color the region \_\_\_\_\_5\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ with the color \_\_\_\_\_\_\_\_\_\_\_\_B\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_