

## Artificial and Computational Intelligence Assignment 2

### Problem statement 1 : Gaming

Two-player Solution-Based Crossword Puzzle: This crossword puzzle is a two-player game where the points are awarded based on the character count in the word (Cat gets 3 points). The player must choose one of the words given and correctly place it in the grid (For example, 6D is Cabbage, 3A is Cucumber) . If the player places it in the wrong grid, then one point is deducted and the other player has to place it in the grid till the right grid is found. The game continues till all the words are placed correctly. Do not use Random seed and select two best solutions after 10 iterations. Print the final variable-value details.

#	#	#	#	#	C	E	L	E	R	Y	O
#	#	#	#	#	A	#	#	#	#	#	N
C	U	C	U	M	B	E	R	#	#	#	I
A	#	A	#	#	B	#	#	#	#	#	O
U	#	P	#	C	A	R	R	O	T	#	N
L	#	S	#	#	G	#	#	#	#	P	#
I	#	I	#	B	E	E	T	R	O	O	T
F	#	C	#	E	#	#	O	#	#	T	#
L	#	U	#	A	#	#	M	#	#	A	#
O	#	M	#	N	#	#	A	#	#	T	#
W	#	#	#	S	#	#	T	#	#	O	#
E	#	#	#	#	#	#	O	#	#	#	#
R	A	D	D	I	S	H	#	#	#	#	#

### Problem statement 2: Logic

The Pragyan Rover from Chandrayan 3 has to move forward, left, right, and backward (c0,c1,c2,c3) and data comes from 14 parameters (Dataset attached). Use the below decision tree and create Prolog rules to predict which direction to move for the given condition. Take some of the attribute values from the user by giving suitable user prompts and predict the class

a3 = false

| a10 = false: c0 (11.0/1.0)

| a10 = true

| | a2 = false

| | | a13 = false: c3 (12.0)

| | | a13 = true

| | | | a0 = false: c0 (2.0)

| | | | a0 = true: c3 (2.0)

| | a2 = true: c0 (7.0/2.0)

a3 = true

| a0 = false

| | a8 = false: c1 (12.0)

| | a8 = true

| | | a10 = false: c0 (5.0)

```

| | | a10 = true: c1 (4.0)

| a0 = true

| | a10 = false

| | | a8 = false

| | | | a4 = false: c2 (6.0/1.0)

| | | | a4 = true: c3 (2.0)

| | | a8 = true: c0 (7.0)

| | a10 = true

| | | a5 = false: c1 (5.0)

| | | a5 = true

| | | | a6 = false: c0 (7.0/2.0)

| | | | a6 = true: c3 (3.0)

```

Dataset : <https://drive.google.com/file/d/1YaN7E8Pb2Bua5MQCXAYiEpPc0EWA0-RW/view?usp=sharing>

**Evaluations will be based on the following.**

1. Use Min-Max algorithm and implement the game in PYTHON (35% marks)
2. Derive the rules from the given decision tree and code as Prolog rules. (35% marks)
3. Interactive implementation. Dynamic inputs-based run of the game with step wise board display and error free game ending. (15% marks)
4. Interactive implementation. Dynamic inputs-based run of the logic expert system with step wise options display and error free recommendation & ending. (15% marks)

**Important Note:**

- You are provided with the python notebook template which stipulates the structure of code and documentation. Use well intended python code.
- Use a separate MS word document for explaining the theory part. Do not include the theory part in the Python notebook except Python comments.
- The implementation code must be completely original and executable.
- Please keep your work (code, documentation) confidential. If your code is found to be plagiarized, you will be penalized severely. Parties involved in the copy will be considered equal partners and will be penalized severely.