Prolog Lab Assignments Set-1

Course: Artificial Intelligence (IAIN532C)

Instructor: Prof. Anupam Date: 24.09.19

Date of Evaluation 27.09.19, (Firyday) Time: **In the Lab Duration**

Instruction for Submission and Evaluation: Each Group have to upload soft copy of the all solution strategy (word or pdf), programs (.pl files), and screenshot of outputs (zip) by 27.09.19, 3pm or at the time of evaluation. Submission link will be uploaded soon in your AI Part-B Course page in my website (http://gcjana.in/). Evaluation will be taken place in the Lab Duration on Friday (27.04.19). *Evaluation strategy:* Viva/program explanation/ running code etc.

Instruction for preparing Solutions: Before programming, first brief the solution strategy in terms of initial state and different types of permissible moves, final states and draw the derivation diagram showing execution trace for the following problems. After defining the Solution Strategy write the prolog program for the following problems.

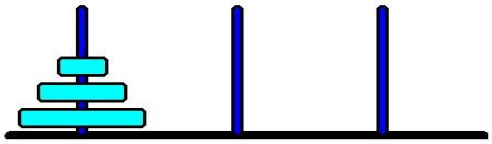
◆

1) Write a prolog program to define two predicates evenlength (list) and oddlength(list) so that they are true if their argument is a list of even or odd length.

2) Write a recursive Prolog program Towers of Hanoi puzzle

Description about problem:

This object of this famous puzzle is to move N disks from the left peg to the right peg using the center peg as an auxiliary holding peg. At no time can a larger disk be placed upon a smaller disk. The following diagram depicts the starting setup for N=3 disks.



3) Write a prolog program of water jug problem.

Description about problem:

A Water Jug Problem: You are given two jugs, a 4-gallon one and a 3-gallon one, a pump which has unlimited water which you can use to fill the jug, and the ground on which water may be poured. Neither jug has any measuring markings on it. How can you get exactly 2 gallons of water in the 4-gallon jug?

State Representation and Initial State – we will represent a state of the problem as a tuple (x, y) where x represents the amount of water in the 4-gallon jug and y represents the amount of water in the 3-gallon jug. Note $0 \le x \le 4$, and $0 \le y \le 3$. Our initial state: (0,0)

Goal Predicate – state = (2,y) where $0 \le y \le 3$.

4) Write a prolog program for missionaries and cannibals

Description About problem:

In this problem, three missionaries and three cannibals must cross a river using a boat which can carry at most two people, under the constraint that, for both banks, that the missionaries present on the bank cannot be outnumbered by cannibals. The boat cannot cross the river by itself with no people on board.

Hint:

First let us consider that both the missionaries (M) and cannibals(C) are on the same side of the river.

	Left	Right
Initially the positions are:	**0M, 0C and**	**3M, 3C (B)**
Now let's send 2 Cannibals to left of bank:	**0M, 2C (B) and**	**3M, 1C**
Send one cannibal from left to right:	**0M, 1C and**	**3M, 2C (B)**
Now send the 2 remaining Cannibals to left:	**0M, 3C (B) and**	**3M, 0C**
Send 1 cannibal to the right:	**0M, 2C and**	**3M, 1C (B)**
Now send 2 missionaries to the left:	**2M, 2C (B) and**	**1M, 1C**
Send 1 missionary and 1 cannibal to right:	**1M, 1C and**	**2M, 2C (B)**
Send 2 missionaries to left:	**3M, 1C (B) and**	**0M, 2C**
Send 1 cannibal to right:	**3M, 0C and**	**0M, 3C (B)**
Send 2 cannibals to left:	**3M, 2C (B) and**	**0M, 1C**
Send 1 cannibal to right:	**3M, 1C and**	**0M, 2C (B)**
Send 2 cannibals to left:	**3M, 3C (B) and**	**0M, 0C**

• Here (B) shows the position of the boat after the action is performed. Therefore, all the missionaries and cannibals have crossed the river safely.

5) i. The following relation classifies numbers into three classes: positive, zero and negative:

```
Class (Number, positive) :- Number > 0.
Class (0, zero).
Class (Number, negative) :- Number < 0.
```

Define this procedure in a more efficient way using cuts.

ii. Define the procedure

split(Numbers, Positives, Negatives)

which splits a list of numbers into two lists: positive ones (including zero) and negative ones. For example,

split([3,-1,0,5,-2], [3,0,5], [-1,-2])

Propose two versions: one with a cut and one without.