## Indian Institute of Information Technology Surat

****

# Lab Report on

# Artificial Intelligence (CS 701) Practical

**Submitted by**

### [RAHUL KUMAR SINGH] (UI21CS44)

**Course Faculty**

### Dr. Ritesh Kumar

### Mrs. Archana Balmik

## Department of Computer Science and Engineering

## Indian Institute of Information Technology Surat

## Gujarat-394190, India

**Aug-2024**

## Lab No: 4

## Aim:

To model complex family relationships, compute population densities, and solve the Towers of Hanoi puzzle in Prolog.

## Description:

Q1. N. Wirth's textbook Algorithms + data structures = programs (1976) contains the following story.

“I married a widow (call her W) who has a grown-up daughter (D). My father (F), who visited us quite often, fell in love with my step-daughter and married her. Hence my father became my son-in-law and my step-daughter became my mother. Some months later, my wife gave birth to a son (S1), who became the brother-in-law of my father, as well as my uncle. The wife of my father-that is, my step-daughter also had a son (S2).”

(i) Using Prolog, create a list of facts that represents the situation in the above story.

(ii) Add rules defining the family relationships (such as father-in-law) described in the story.

(iii) Show how a Prolog system would use your program to prove the goal “I am my own grandfather".

Q2. Load the following facts into density.pl, consult the prolog file and calculate the population density of each country.

pop(usa,327). /\* 327 million \*/

pop(india,1300). /\* 1.3 billion \*/

pop(australia, 25).

pop(canada,36).

area(usa,4). /\* millions of square miles \*/

area(india,1).

area(australia,3).

area(canada,4).

Q3. Towers of Hanoi puzzleAim: To move N disks from the left peg to the right peg using the center peg as an

auxiliary holding peg.

Rule: At no time can a larger disk be placed upon a smaller disk.

Using prolog, develop a set of rules which can be applicable for any number of disks (for

ex: N=3 or 5 or 7 etc.) with peg size = 3.

## Code:

**Q1**

% Facts

married(john, widow).

married(father, step\_daughter).

parent(widow, step\_daughter).

% parent(john, step\_daughter).

parent(step\_daughter, john).

parent(john, son1).

parent(step\_daughter, son2).

% Definition

step\_daughter\_of(X, Y) :-

married(Y, Z),

parent(Z, X),

\+ parent(Y, X).

grandfather(X, Y) :-

parent(X, Z),

parent(Z, Y).

father\_in\_law(X, Y) :-

married(Y, Z),

parent(X, Z).

sibling(X, Y) :-

parent(P, X),

parent(P, Y),

X \= Y.

uncle\_of(X, Y) :-

parent(Z, Y),

sibling(X, Z).

brother\_in\_law\_of(X, Y) :-

married(X, Z),

sibling(Y, Z).

brother\_in\_law\_of(X, Y) :-

married(Y, Z),

sibling(X, Z).

% Method

is\_my\_own\_grandfather(Person) :-

grandfather(Person, Person).

**Q2**

pop(usa,327).

pop(india,1300).

pop(australia,25).

pop(canada,36).

area(usa,4).

area(india,1).

area(australia,3).

area(canada,4).

density(Country, Density) :-

pop(Country, Population),

area(Country, Area),

Density is Population / Area.

**Q3**

hanoi(0, \_, \_, \_).

hanoi(N, A, B, C) :-

N > 0,

M is N - 1,

hanoi(M, A, C, B),

write('Move disk '), write(N), write(' from '), write(A), write(' to '), write(C), nl,

hanoi(M, B, A, C).

## Output:

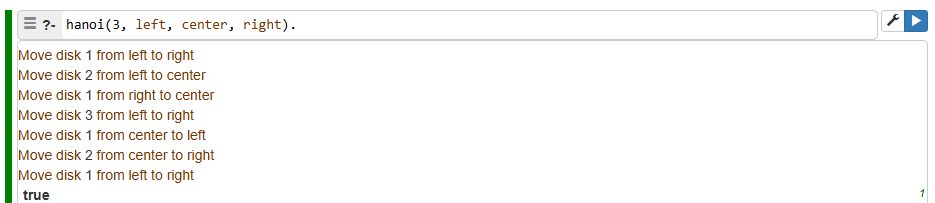
**Q1**

****

**Q2**

****

**Q3**

****

## Conclusion:

* Modeled complex family relationships in Prolog.
* Computed population densities using Prolog facts.
* Developed rules to solve the Towers of Hanoi puzzle for any number of disks.