Namal Institute Mianwali

Khan Family Problem

Artificial Intelligence

Department of Computer Science

Year 3, Semester 5

Group Members:

Muhammad Asad	**BSCS201901**
Muhammad Tariq	**BSCS201912**
Muhammad Asif	**BSCS201904**
Areeba Rashid	**BSCS201918**
Hurmat Ilyas	**BSCS201902**
Fouzia Nawaz	**BSCS201922**

Submitted To:

Dr. Junaid Akhtar

Submitted Date:

2nd January, 2022

Abstract

Our program explores the inter-family relationships of the Khan Family. It canfind a person's baap, beti, bahu, beta, pota, dada, nawasa, nana, sussar, dadi,chachataya, nani, khala, sala, or baapDada if that relationship exists. The front end of theprogram is written in Python and the backend in Prolog. Python is a general-purpose andhigh-level programming language. It is an interpreted language. On the other hand, Prologis a logic programming language. It is primarily intended as a declarative programminglanguage. It has applications in artificial intelligence and computational linguistics.

Table of contents.

Abstract **2**

What is prolog? 4

Key Features: **4**

1. **Unification** **4**
2. **Backtracking** **4**
3. **Recursion** **4**
4. **Facts** **4**
5. **Terms** **5**
6. **Queries** **5**
7. **Variables** **5**
8. **Rules** **5**

Example **5**

Problem Statement 6

Khan Family Tree 7

Rules of this problem 7

Prolog:

Prolog is a logic programming language. it's an important role in AI. Unlike several different programming languages, logic programming is meant primarily as a declarative programming language. In prolog, logic is expressed as relations (called Facts and Rules). The core heart of logic programming lies within the logic being applied. Formulation or Computation is administered by running a question over these relations.

In prolog, we tend to declare some facts. These facts represent the cognitive content of the system. We will question the cognitive content. we tend to get output as affirmative if our question is already within the cognitive content or it's inexplicit by cognitive content, otherwise, we tend to get output as negative. So, cognitive content is thought-about like info, against that we will question. programming language facts square measure expressed in a definite pattern. Facts contain entities and their relation. Entities square measure written inside the parentheses separated by comma (,). Their relation is expressed at the beginning and outdoors the parenthesis. each fact/rule ends with a dot (.).

Key Features:

1. **Unification**

   The basic idea is can the given terms be made to represent the same structure.

2. **Backtracking**

   When a task fails, the prolog traces backwards and tries to satisfy the previous task.

3. **Recursion**

   Recursion is the basis for any search in the program.

4. **Facts**

   Facts are statements that describe properties of objects or relationships betweenobjects.

   Example: *mianBiwi (‘ChoteKhan’, ‘ChotiRani’).*

In the above example "**mianBiwi**" is a relation between two objects"**ChoteKhan**"and "**ChotiRani**".Collections of many facts and rules form a database. It represents the knowledge insome logical form.

5. **Terms**

   In prolog, all kind of data is called terms. e.g., parent (abc, def), that whole is considered as a complex term consisting of simple terms i.e., abc, def.

6. **Queries**

   A query is basically a request to retrieve information from the database. e.g.? - parent (X, kauser). Ans: chotekhan, chotirani

7. **Variables**

   They are kind of prolog terms that starts with capital letters or with underscore "_".? - mianbiwi (X, chotirani). "X" is a variable.

8. **Rules**

   Rules help us to derive new property from old defined facts from the knowledge base. beti (X, Y): - parent (Y, X), gins (female, X).

Example

Facts:

```
food(burger).       // burger is a food

food(sandwich).     // sandwich is a food

food(pizza).        // pizza is a food

lunch(sandwich).    // sandwich is a lunch

dinner(pizza).      // pizza is a dinner
```

Rule:

meal(X): - food(X). // Every food is a meal OR
Anything is a meal if it is a food

Goals:

? - food(pizza). // Is pizza a food?

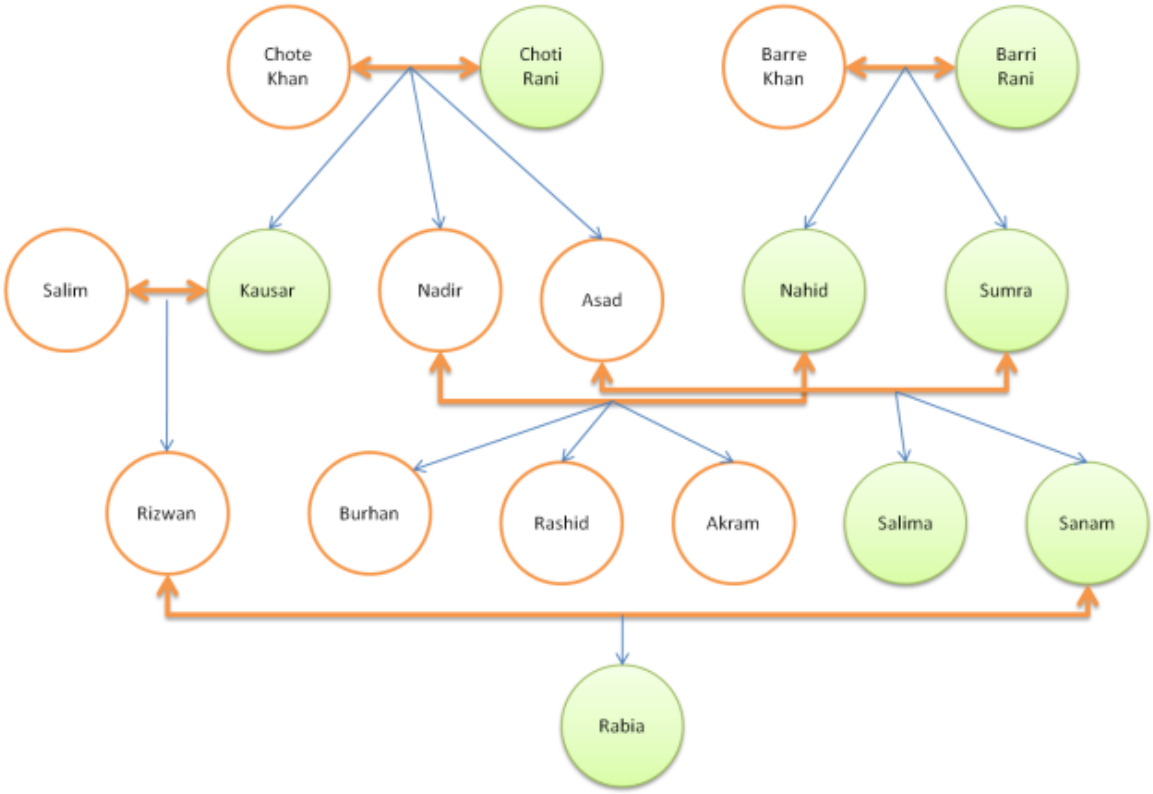
? - meal(X), lunch(X). // Which food is a meal and lunch?

? - dinner(sandwich). // Is a sandwich a dinner?

Problem Statement

In this project, we have to code khan's family tree by using provided facts, and then we must implement rules given in the requirements document. The application should be able to ask few questions like, "Who is the father of X, who is beta of X",etc.

Khan Family Tree



Rules of this problem

- baap (Variable1, Variable2): -parent (Variable1, Variable2), gins ('Male', Variable1).
- beti (Variable1, Variable2).
- beta (Variable1, Variable2).
- dada (Variable1, Variable2).
- nana (Variable1, Variable2).
- dadi (Variable1, Variable2).
- nani (Variable1, Variable2).
- sala (Variable1, Variable2).
- bahu (Variable1, Variable2).
- pota (Variable1, Variable2).

- nawasa (Variable1, Variable2).
- sussar (Variable1, Variable2).
- chachataya (Variable1, Variable2).
- khala (Variable1, Variable2).
- baapDada (Variable1, Variable2).

Back-End:

We implemented the given rules.

- mianBiwi (mian, biwi)
- parent (parent, child)
- gins (gender, person)

Rules:

- **beti (X, Y):** -
  parent (Y, X), gins ('Female', X). This rule finds the daughter if we provide parent to this rule such as X is the daughter of Y and we can find this by using the fact parent (Y, X) and gins ('Female', X).
- **beta (X, Y):** -
  parent (Y, X), gins ('Male', X). This rule finds the son if we provide parents to this rule. Such as X is the son of Y and we can find this by using the fact parent (Y, X) and gins ('Male', X).
- **dada (X, Y):** -
  parent (X, Z), parent (Z, Y), gins ('Male', X), gins ('Male', Z). This rule finds the dada if we provide pota/poti to this rule. Such as X is dada of Y and we can find this by getting parent of the parent of Y giving the condition that all parents are male.
- **nana (X, Y): -**

parent (X, Z), parent (Z, Y), gins ('Male', X), gins ('Female', Z). This rule finds the nana if we provide nawasaa/nawasii and vice versa. We can find the female parent (Z) of child Y and then find the male parent of mama (Z).

- **dadi (X, Y):** -
  parent (X, Z), parent (Z, Y), gins ('Female', X), gins ('Male', Z). This rule finds the dadi if we provide pota/poti and vice versa. We can find male parent (Z) of child Y and then find the female parent of papa (Z).
- **nani (X, Y): -**
  parent (X, Z), parent (Z, Y), gins ('Female', X), gins ('Female', Z). This rule finds the nani if we provide nawasaa/nawasii and vice versa. We can find the female parent (Z) of child Y and then find the female parent of mama (Z).
- **sala (X, Y): -**
  mianBiwi (Y, Z), parent (A, Z), gins ('Female', Z), parent (A, X), gins ('Male', A), gins ('Male', X). This rule finds the sala such as brother of the wife. Find the wife of the husband (Y) and then find the parents of the wife, after this gets all the male child of wife parent.
- **bahu (X, Y):** -
  parent (Y, Z), gins ('Female', X), gins ('Male', Z), mianBiwi (Z, X). Bahu is the wife of the son. To find bahu, we first find son of Y and then find the wife of the son.
- **pota (X, Y):** -
  parent (Y, Z), parent (Z, X), gins ('Male', X), gins ('Male', Z). We find the son of the son.
- **nawasa (X, Y):** -
  parent (Y, Z), parent (Z, X), gins ('Male', X), gins ('Female', Z). We find daughter's son.
- **sussar (X, Y):** -
  mianBiwi (Y, Z), parent (X, Z), gins ('Male', X), gins ('Female', Z), gins ('Male', Y).

- **sussar (X, Y): -**
  mianbiwi (Z, Y), parent (X, Z), gins ('Male', X), gins ('Male', Z), gins ('Female', Y). This finds the husband's father (or) wife's father.
- **baapDada (X, Y): -**
  parent (X, Y), gins ('Male', X).
- **baapDada (X, Y):** -
  parent (X, Z), baapDada (Z, Y), gins ('Male', Z), gins ('Male', X). This finds ancestors of a given child.
- **khala (X, Y): -**
  parent (Z, Y), gins ('Female', Z), parent (V, Y), gins ('Male', V), parent (A, Z), gins (male, A), parent (A, X), gins (female, X), not (mianBiwi (V, X)). This rule finds khala sister(s) of mother of a child. The main logic in this rule is that find female children of nana and then exclude son's / daughter's mother by using 'not (mianBiwi (papa, khala))' such as all female children of nana who are not the wife of child's (Y) father.
- **chachataya (X, Y): -**
  parent (Z, Y), gins (male, Z), parent (M, Y), gins (female, M), parent (A, Z), gins (male, A), parent (A, X), gins (male, X), not (mianBiwi (X, M)).
  This rule finds the chacha/Taya brother of the father of a child. The main logic in this rule is that find male children of dada and then exclude son's / daughter's father by using "not (mianBiwi (chacha, mama))" i.e., all male children of dada who are not husbands of child's (Y) mother.

Front End:

In front end we used prolog logic using the interface. We integrated prolog and python for querying using the interface.

PySwip:

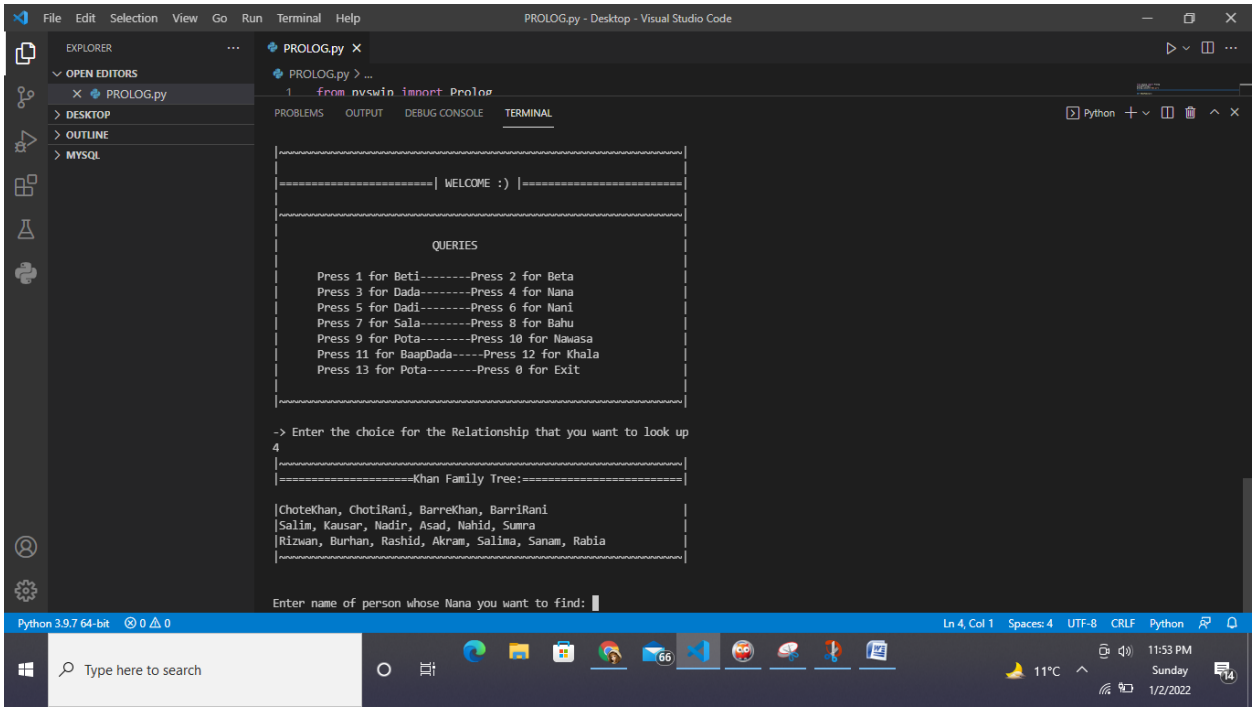
PySwip is a Python library. It provides us a utility that makes it easy to query with the backend Prolog using a python interface.

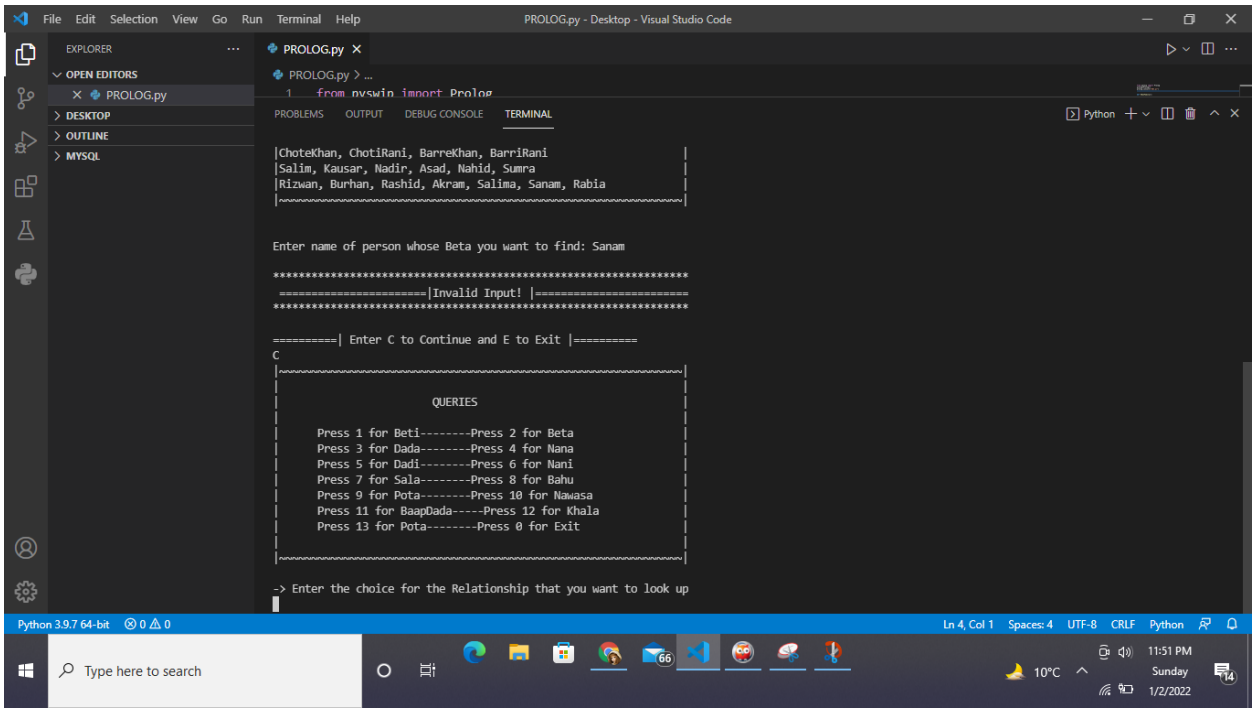
There are two functions used first one is:

Prolog.consult(filename.pl): This function links the backend prolog file with the python program. Backend prolog file contains facts and rules from which we can query for certain results derived from rules.

Prolog. query (query): Prolog. query(query) gives input to the backend prolog file as query and returns the result of that query. We give input 'Y' to that function and store returned result in the 'value' array.

Front End:




File Edit Selection View Go Run Terminal Help
PROLOG.py - Desktop - Visual Studio Code
EXPLORER
OPEN EDITORS
PROLOG.py
DESKTOP
OUTLINE
MYSQL
PROLOG.py
from pyswip import Prolog
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
Python
|ChoteKhan, ChotiRani, BarreKhan, BarriRani |
|Salim, Kausar, Nadir, Asad, Nahid, Sumra |
|Rizwan, Burhan, Rashid, Akram, Salima, Sanam, Rabia |
Enter name of person whose Beta you want to find: Sanam
|Invalid Input! |
| Enter C to Continue and E to Exit |
C
QUERIES
Press 1 for Beti--------Press 2 for Beta
Press 3 for Dada--------Press 4 for Nana
Press 5 for Dadi--------Press 6 for Nani
Press 7 for Sala--------Press 8 for Bahu
Press 9 for Pota--------Press 10 for Nawasa
Press 11 for BaapDada-----Press 12 for Khala
Press 13 for Pota--------Press 0 for Exit
-> Enter the choice for the Relationship that you want to look up
Python 3.9.7 64-bit
Ln 4, Col 1 Spaces: 4 UTF-8 CRLF Python
Type here to search
10°C
11:51 PM
Sunday
1/2/2022