Report on solution o Family Relations using Pro-log/Python

Zeemal Urooj-> 2018-UET-NML-CS-19
Sajid Hameed-> 2018-UET-NML-CS-13
Mahnoor Awan-> 2018-UET-NML-CS-15
Syed Uzair-> 2018-UET-NML-CS-20
Zeeshan Ahmed-> 2018-UET-NML-CS-18
Noor Nabi-> 2018-UET-NML-CS-02
Shah Rukh-> 2018-UET-NML-CS-29

12-March-2021

NAMAL INSTITUTE

Abstract

Our aim of this project is, we have to integrate two languages, Pro-log and Python. We made two interfaces for that purpose, front-end and back-end interface. We will do back-end work on pro-log and front-end work on python. We did this in order to provide ease to the user so that he can run any of his/her query without any difficulty. Using this mechanism, we solved a problem of family tree of "Khan Family". User can enter the query without any difficulty and get the relationship between family members.

Contents

1 Introduction

Pro-log is a logic programming language. And it has a very important role in Artificial intelligence. Pro-log is a declarative programming language. In this, logic is defined as relations include facts and rules. Pro-log is used for pattern matching over natural language parse trees.
There is a theory which states that if ever anyone discovers exactly what the Universe is for and why it is here, it will instantly disappear and be replaced by something even more bizarre and inexplicable. There is another theory which states, that this has already happened.
Using this mechanism, we will solve the problem of a family tree. User will easily know any of the relation by just putting the required information. User will enter its queries in an interface of python and python will get values from the pro-log and return it the user.

2 Goals

Our goal of this project is to first analyze the given family tree. Then we need to make relations between the nodes of the trees using the three facts given in the question. After that we have to make functions for the rules which a user can ask from the program.
In order to ease the user, we have to make our back-end function in the pro-log and we have to make an interface in the python language so that our user can easily enter his query and get the required result.

3 Basic features of Pro-log

3.1 Facts

Facts define the properties of the object and also define their relation. And the collection of facts forms the data bases and then convert the data base into logical format. We expressed the properties and relation of object such as gender of character e.g male and relation with percentage sign.
In our case, we have only to tackle with only three facts because we can solve any of our required query using only three facts. These three facts are:

- mianbiwi(X,Y).
- parents(X,Y).
- gins(X,Y).

3.2 Rules

Using rules we derive new properties or relation of object from the existing one. Rules consist of a term called head and conjunction of goal called body. The body is true only if the head is true. Variables of rules are shared between head and body. Facts and rules are also called clauses. We use these rules to get result of any of the query. If our query is valid and the program has its answer it will return the value otherwise it will return the false value. We have rules in our program like give below:

- pota(X,Y) :-
- dada(X,Y) :-
- nawasa(X,Y) :- etc

3.3 Term

In Pro-log, data in any form called Term. Numbers, alphabets, variables, atoms all of them are term. Terms which are made up of another term is called **Sub-term**.

3.4 Atom

Atom is a single unit which is parsed by the pro-log reader. It is composed of a sequence of characters. It does not has any inherent meaning. It is a general purpose name.

4 Methodology

4.1 Problem

We have a family tree of a family named as "Khan Family". We have to make a program which will give us relation between any two nodes. Like if we have to find the father of A, we should not need to focus on whole tree. Our program should give us the required output by just writing the query in a simple interface.
We should develop such a program which should provide ease to the user.

A user can easily enter a simple query to find the answer of required relation between two members.

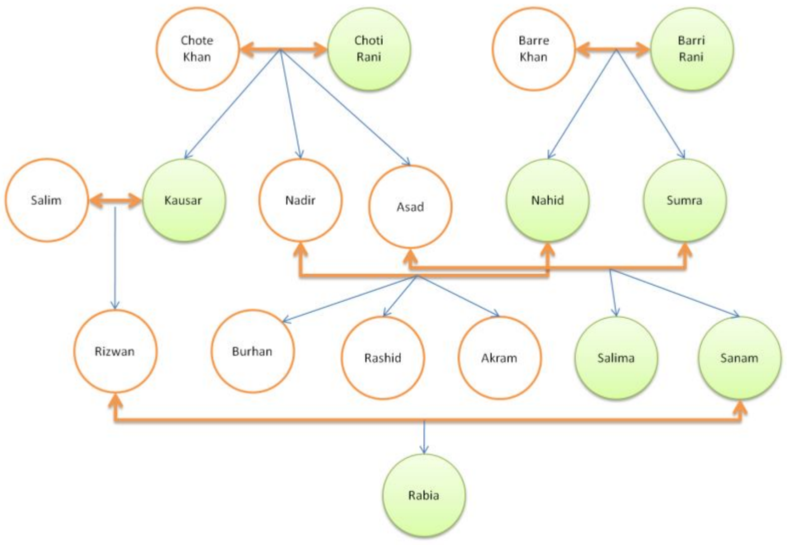


Figure 1: Khan Family Tree

4.2 Task Division

Tasks were divided into different group members.

4.2.1 Project Manager

Mahnoor Awan was managing the project activities throughout the project flow.

4.2.2 Pro-log

Syed Uzair and Muhammad Sajid Hameed Khan were writing the pro-log code for back-end interface.

4.2.3 Python

Noor Nabi and Shah Rukh Khan were writing the python code for front-end interface.

4.2.4 Report Writing

Zeemal Urooj and Muhammad Zeeshan Ahmed were writing the report of project.

4.3 Interfaces

We developed two major interfaces in this project. One is back-end and second is front-end interface. Back-end interface is simply designed to find the solution of given problem in pro-log programming language. And front-end interface is designed in python programming language for the better interaction of user with interfaces.

4.4 Solution

We analyzed the whole tree. Then we made relations between the nodes of the trees by using the given facts. On the basis of these facts, we made our rules which we also called clause or functions. These functions will give us the value of our required query.
To achieve the ease for the user, we make our back-end code on pro-Log which includes all the relations and functions and we made our front end interface in python because it is quite difficult for the user to enter query in the pro-log.

4.5 Operations

A user can perform various operations using our program. Some of them are as follows:

- beta(X,Y) :-
- sala(X,Y) :-
- baapdada(X,Y) :-
- nani(X,Y) :-
- khala(X,Y) :-
- sassur(X,Y) :- etc

A user can get the value of X by giving the value of Y to the program or can also get the value of Y by giving the value of X. Another operation which a user can perform is, he can check whether a relation is valid or invalid by giving the values of both X and Y.

4.6 Tools

We used different programming tools in this project. Details of these tools are given below:

4.6.1 SWI-Pro-log

To designed the back-end interface in pro-log programming language, we used **SWI-Pro-log** to code the solution in pro-log programming language.

4.6.2 Visual Studio Code

To design the front-end interface in python programming language, we used **Visual Studio Code** to code front-end interface.

4.7 Coding in Pro-log

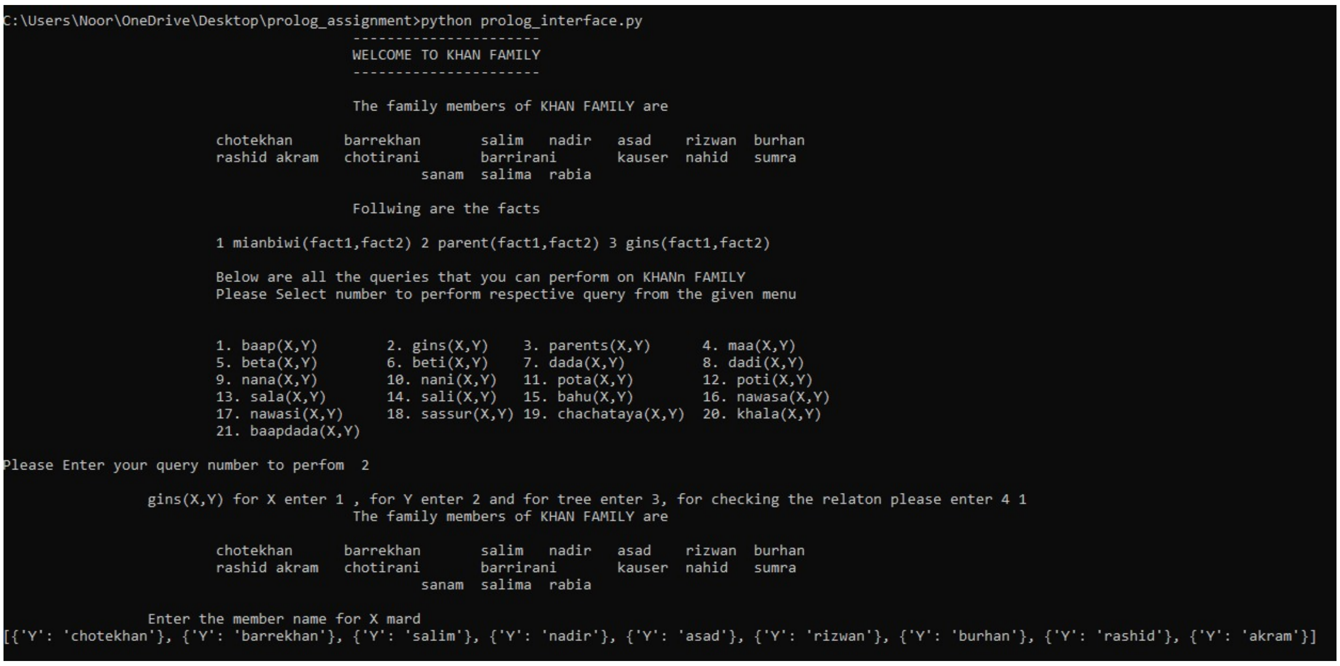
First of all, we find the solution of the given problem of family tree using facts, rules and goals in pro-log language. We used fact, rules, relations and goals in this language to design the solution. After making the solution, we tested the program on different goals of problem. And all results were true.

4.8 Coding in Python

After coding the solution in Pro-log, we had to code front-end interface in python. Python code was for better interaction of user. We will take the query from user in the front-end interface and find the solution using pro-log at the back-end interface. We used the **Pyswip** library to bridge the pro-log code and python code.
We used function **prolog.consult("FileName")** to connect python with the pro-log. An existing knowledge base stored in a Prolog file can also be consulted, and queried. Assuming the filename "prolog.pl" and the Python is being run in the same working directory. Function **prolog.query(query)** runs the query received by the user.

4.9 Integration

After designing both back-end and front-end interfaces, we had to integrate the both interfaces so that user can easily use the front-end interface and get the solution from back-end interface. In front-end interface, user will just enter the query and this query will run in python code and will be passed to back-end interface. And back-end interface will find the solution of query and return it to the user.
After integration of both interfaces, we were getting final interface which is given below:



```
C:\Users\Noor\OneDrive\Desktop\prolog_assignment>python prolog_interface.py
                                         ----------------------
                                         WELCOME TO KHAN FAMILY
                                         ----------------------

                                         The family members of KHAN FAMILY are

                         chotekhan      barrekhan       salim   nadir   asad    rizwan  burhan
                         rashid akram   chotirani       barrirani       kauser  nahid   sumra
                                                sanam   salima  rabia

                                         Follwing are the facts

                         1 mianbiwi(fact1,fact2) 2 parent(fact1,fact2) 3 gins(fact1,fact2)

                         Below are all the queries that you can perform on KHANn FAMILY
                         Please Select number to perform respective query from the given menu


                         1. baap(X,Y)        2. gins(X,Y)    3. parents(X,Y)      4. maa(X,Y)
                         5. beta(X,Y)        6. beti(X,Y)    7. dada(X,Y)         8. dadi(X,Y)
                         9. nana(X,Y)        10. nani(X,Y)   11. pota(X,Y)        12. poti(X,Y)
                         13. sala(X,Y)       14. sali(X,Y)   15. bahu(X,Y)        16. nawasa(X,Y)
                         17. nawasi(X,Y)     18. sassur(X,Y) 19. chachataya(X,Y)  20. khala(X,Y)
                         21. baapdada(X,Y)

Please Enter your query number to perfom  2

                 gins(X,Y) for X enter 1 , for Y enter 2 and for tree enter 3, for checking the relaton please enter 4 1
                                         The family members of KHAN FAMILY are

                         chotekhan      barrekhan       salim   nadir   asad    rizwan  burhan
                         rashid akram   chotirani       barrirani       kauser  nahid   sumra
                                                sanam   salima  rabia

                 Enter the member name for X mard
[{'Y': 'chotekhan'}, {'Y': 'barrekhan'}, {'Y': 'salim'}, {'Y': 'nadir'}, {'Y': 'asad'}, {'Y': 'rizwan'}, {'Y': 'burhan'}, {'Y': 'rashid'}, {'Y': 'akram'}]
```

Figure 2: Interface after integration

4.10 Limitations

During the design of solution of problem, we had some limitations.

- We have to use only three facts described in section **3.1 Facts**.

5 Conclusion

We made the front-end and back-end interface respectively in python and pro-log language. Then our finals task was to integrate the both interfaces so users can use it easily without entering the pro-log commands. After integrating the program of pro-log programming language and python programming language, we build an interactive user interface with hundred percent accuracy. User were able to enter the query of finding relationships of family members. After entering the query, all the possible relations from back-end interface (pro-log program) are returned. If a user wants to check a specific relation between two family members, it will simply return true or false showing that they have above relation or not. And it can also return the all possible relations between members. This interface was very easy to use for all type of users.

6 References

- PySwip. Retrieved March 12, 2021, from https://pypi.org/project/pyswip/
- Pro-log Video Tutorial. Retrieved March 12, 2021, from https://www.youtube.com/watch?v=W0l5WcUSwm4
- Pro-log pdf1 tutorial. Retrieved March 12, 2021, from http://cs.union.edu/~striegnk/courses/esslli04prolog/slides/0.day1.pdf
- Pro-log pdf2 tutorial. Retrieved March 12, 2021, from http://cs.union.edu/~striegnk/courses/esslli04prolog/slides/0.day2.pdf