CS570 – Artificial Intelligence Project 4

Chihsiang Wang

Genealogy with Prolog

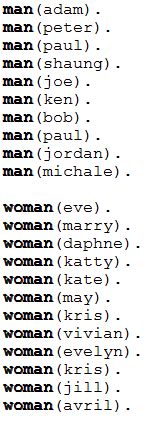
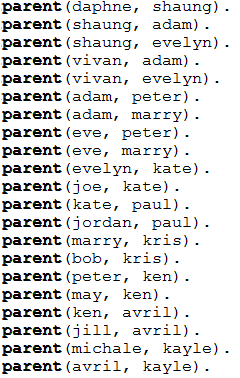
By Chihsiang Wang

**Abstract**

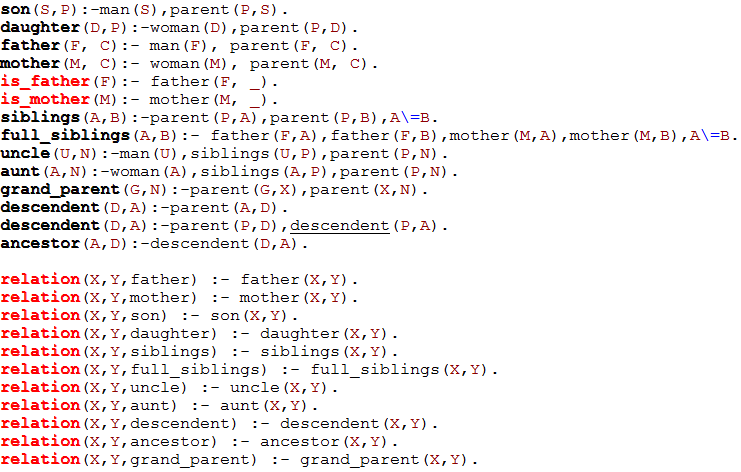
In this project I am going to use Prolog language to implement a genealogy. Prolog is a general-purpose [logic programming](http://en.wikipedia.org/wiki/Logic_programming) language, and it’s also one of first logic programming language. Prolog program describe relations between two or more objects, it has two clauses: facts and rule. In this project I gave two kinds of facts to define the genders for each persons in this genealogy, for example, man (adam) means Adam is a man, woman (eve) means Eve is a woman; and I also define the fact that who is whose parent, for example, parent (daphne, shaung) shows that Daphne is Shaung’s parent. Using these facts I can build several rules to define the relations between each persons, in this project I defined these relations: son, daughter, father, mother, siblings (same mother or father, but different mother or father), full\_siblings(must be same mother and father), uncle, aunt, grandparent, descendent, ancestor. After build the knowledge base, I can ask the question and get the answer (true or false), or I can ask to list all names, which I want to get. The result of program always shows me the satisfactory answers – nothing can be answered incorrectly.

**Knowledge Base**

There have three kinds of facts in the program: man, woman, and who are whose parents, these three kinds of facts already enough to build a complete genealogy. The genealogy graph is attached on the appendix page.

 **Figure 1. Facts.**

Applies by these facts, I can build other relation rules; it’s pretty logic things. In this project I build 25 rules, it shows as figure.

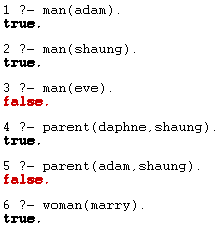


**Figure 2. Rules.**

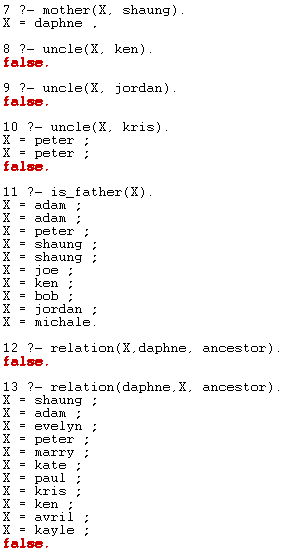
After build a knowledge base with these facts and base, the prolog language will do logic reasoning to connect each objects, and now I can ask the program some questions.

**Result**

First of all, I can ask some simple questions to get the answer true or false, the sample output lists by number 1 – 6.



1. Is Adam a man? Answer: true.
2. Is Shaung a man? Answer: true.
3. Is Eve a man? Answer: false.
4. Is Daphne a parent of Shaung? Answer: true.
5. Is Adam a parent of Shaung? Answer: false.
6. Is Marry a woman? Answer: true.



7. Who is mother of Shaung? Answer: Daphne.

8. Who is uncle of Ken? Answer: false, Ken has no uncle.

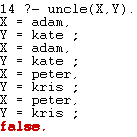
9. Who is uncle of Jordan? Answer: false.

10. Who is uncle of Kris? Answer: Peter.

11. Who is a father? Answer: Adam, Peter, Shaung, Joe, Ken, Bob,Jordan.

12. Who is Daphne’s ancestor? Answer: Shaung, Adam, Evelyn, Peter, Marry, Kate, Paul, Kris, Ken, Avril, Kayle.

Or I can use a variable to show what can make this rule true, lists number 7 – 13, When the answer is more than one, I will need to use “;” symbol to show each answers, if there has no more answers, the compile will return false. I also can use two variables to get more complex answers, for example, I use uncle(X,Y) to show that who are whose uncle in a whole genealogy.

****

14. Adam is Kate’s uncle, Peter is Kris’s uncle

**Discussion**

Prolog language is really different as other programming languages like C, Java, Python, it has no if/else, for/while statements. Prolog’s programming style is more like human’s nature thinking and logic seasoning. A knowledge base determines how good a system works; so it means right facts and rules is most important setting. I have to be really careful while cdoing a prolog program, because even just one mistake or typo can destroy whole logic. Facts build rules, and facts are also rules, how to use facts to create rules, and use these rules to create more rules is the trickiest part while coding. When I first to do the program, I wrote too much facts, like to define the relation of “father” and “mother”, and that is a lot of hard typing works. But after I found some reference about Prolog, I just realize that how to use facts to build rules, so finally I modified my knowledge base only has three kinds of facts, and it’s totally enough for this project. Prolog is a very logical programming language, but a code designer will need a good and clear logic think as well. After build a good knowledge base, the answer from it can always be satisfied, in this project, it has 100% correct answers.

**Appendix**

The figure 1 and 2 shows all of my knowledge base.

The genealogy graph list below:

