CS-470: Artificial Intelligence: Project 4

Taylor Martin
University of Idaho
Moscow, Idaho, USA
mart8517@vandals.uidaho.edu

1 ABSTRACT

This project employed techniques in first-order logic and Prolog from the field of Artificial Intelligence, specifically from the content presented in CS-470. The primary objective focused on developing a comprehensive Knowledge Base in Prolog that 'understands' various family relationships and allows the formulation of queries to obtain relevant information. To achieve this objective, a set of facts and rules were carefully defined within the genealogy Knowledge Base. The employed methodology yielded highly positive outcomes, reinforcing the confidence in the accuracy and reliability of the generated results.

2 DESCRIPTION OF THE PROLOG KNOWLEDGE BASE (KB) UTILIZED

2.1 KB Facts

The Knowledge Base in the "King Of The Hill" Genealogy utilizes facts to define parent relationships between individuals. For instance:

```
parent(hank, bobby)
parent(peggy, bobby)
```

In this example, the facts indicate that Peggy and Hank are the parents of Bobby. By declaring such facts, the Knowledge Base establishes the parent-child connections necessary to represent the family relationships accurately.

2.2 KB Rules

The rules defined in the Genealogy Knowledge Base (KB) consistently adhere to the following syntax:

```
mother(X, Y) :-
parent(X, Y),
female(X).
```

This rule signifies that X is considered a mother if X is a parent and also female. All of the rules within the Knowledge Base are formulated as implications, allowing for the representation of various familial relationships and their associated properties.

3 RESULTS

Below are several different queries made to test the accuracy of the Prolog Knowledge Base.

3.1 Successful Queries

```
(15 ms) yes
| ?- father(X,bobby).

X = hank ? |
```

Figure 1: Bobby Hill's Father

```
| ?- mother(X,bobby).

X = peggy ? |
```

Figure 2: Bobby Hill's Mother

```
| ?- cousin(X,bobby).

X = luanne ? |
```

Figure 3: Bobby Hill's Cousin

```
| ?- grandfather(X,bobby).

X = cotton ? |
```

Figure 4: Bobby Hill's Grandfather

```
| ?- son(X,Y).

X = hank
Y = cotton ? a

X = hank
Y = tilly

X = gh
Y = cotton

X = gh
Y = didi

X = hoyt
Y = doc

X = hoyt
Y = maddy

X = bobby
Y = hank

X = bobby
Y = peggy
```

Figure 5: All Sons In KB

```
father(X,Y).

X = cotton
Y = hank ? a

X = doc
Y = peggy

X = cotton
Y = gh

X = doc
Y = hoyt

X = john
Y = leanne

X = hank
Y = bobby

X = hoyt
Y = luanne
```

Figure 6: All Fathers In KB

```
| ?- mother(X,Y).

X = maddy
Y = peggy ? a

X = didi
Y = gh

X = maddy
Y = hoyt

X = jane
Y = leanne

X = peggy
Y = bobby

X = leanne
Y = luanne

(78 ms) yes
```

Figure 7: All Mothers In KB

```
| ?- daughter(X,Y).

X = peggy
Y = doc ? a

X = peggy
Y = maddy

X = leanne
Y = jane

X = leanne
Y = john

X = luanne
Y = hoyt

X = luanne
Y = leanne
```

Figure 8: All Daughters In KB

```
| ?- stepmother(X,Y).

X = didi
Y = hank ? a

(16 ms) no
```

Figure 9: All Stepmothers In KB

3.2 Failed Queries

There was only one specific use case in which I encountered a failure in the query. When attempting to retrieve the uncle or aunt

through marriage of a person, the knowledge base rule failed to provide the expected result. In the given scenario, the query should have correctly identified that Hank is the uncle of Luanne, but it did not produce the desired output.

```
| ?- uncle(X,Y).

X = gh
Y = bobby ? a

X = hoyt
Y = bobby
```

Figure 10: Bobby Hill's Father

4 DISCUSSION

The developed Prolog Genealogy Knowledge Base exhibits several strengths, but also presents certain limitations. By analyzing its performance and usability, we can gain insights into its capabilities and areas for improvement.

One of the notable strengths of the Knowledge Base is its ability to effectively represent and comprehend a wide range of family relationships. Through the carefully defined set of facts and rules, the Knowledge Base provides a solid foundation for capturing intricate connections between individuals. This enables it to accurately answer a variety of questions related to family ties, such as determining parent-child relationships, siblings, grandparents, and more. The utilization of first-order logic and Prolog techniques enhances the expressiveness and flexibility of the Knowledge Base, allowing for complex relationships to be represented and queried.

However, despite its strengths, there are certain limitations that should be acknowledged. One potential weakness is the Knowledge Base's inability to handle certain types of queries or answer specific questions. While it successfully addresses many common family relationship inquiries, there may be instances where it falls short. For example, when attempting to query more complex relationships like uncles through marriage, the Knowledge Base may encounter difficulties and fail to provide the desired answer. Identifying and addressing such gaps in knowledge representation would be beneficial for enhancing the overall effectiveness of the system.

In terms of extensibility, the Knowledge Base demonstrates relative ease in adding new rules. The modular nature of Prolog allows for straightforward integration of additional facts and rules, enabling the expansion of the Knowledge Base to encompass new relationships or scenarios. This flexibility ensures that the system can be continuously updated and refined as new information becomes available or as requirements evolve.

Formulating general queries within the Knowledge Base is also relatively straightforward. The use of Prolog's query mechanism allows users to express their queries in a logical and intuitive manner, making it accessible for users with varying levels of familiarity with the system. The ability to construct queries using logical operators and variables provides a means to obtain specific information from the Knowledge Base, enhancing its usability.

5 CONCLUSION

In conclusion, this project successfully applied techniques in first-order logic and Prolog from the domain of Artificial Intelligence, building upon the knowledge acquired in the CS-470 course. The main objective centered around the creation of a Genealogy Knowledge Base in Prolog, capable of comprehending and representing family relationships. The developed Knowledge Base enabled the formulation of queries to extract information from the system. Through the careful definition of facts and rules within the genealogy Knowledge Base, the project achieved highly favorable outcomes. The results obtained instill a strong sense of confidence in the accuracy and reliability of the system, validating the effectiveness of the employed methodology.

6 CODE APPENDIX

```
 /*
# --------------------
# @file    genealogy.pl
# @author  Taylor Martin
# @date    June 2023
# @class   CS 470 Artificial Intelligence
# @project Project 4 - Genealogy
# @brief   This program uses Prolog to create a family tree
     Knowledge Base (KB) and answer queries about the
     family tree.
# --------------------
*/

/*********** King Of The Hill Genealogy KB Facts
     *************/

% parents of Hank
parent(cotton, hank).
parent(tilly, hank).

% parents of Peggy
parent(doc, peggy).
parent(maddy, peggy).

% parents of gh
parent(cotton, gh).
parent(didi, gh).

% parents of Hoyt
parent(doc, hoyt).
parent(maddy, hoyt).

% parents of Leanne
parent(jane, leanne).
parent(john, leanne).

% parents of Bobby
parent(hank, bobby).
parent(peggy, bobby).

% parents of Luanne
parent(hoyt, luanne).
parent(leanne, luanne).
```

```
% males in genealogy
male(bobby).
male(hank).
male(cotton).
male(hoyt).
male(doc).
male(john).
male(gh).

% females in genealogy
female(peggy).
female(luanne).
female(didi).
female(leanne).
female(maddy).
female(jane).


% *************King Of The Hill Genealogy KB
     Rules*******************

mother(X, Y) :-
    parent(X, Y),
    female(X).

father(X, Y) :-
    parent(X, Y),
    male(X).

child(X, Y) :-
    parent(Y, X).

partner(X, Y) :-
    child(Z, X),
    child(Z, Y),
    X \= Y.

grandparent(X, Y) :-
    parent(X, Z),
    parent(Z, Y).

grandchild(X, Y) :-
    grandparent(Y, X).

grandfather(X, Y) :-
    grandparent(X, Y),
    male(X).

grandmother(X, Y) :-
    grandparent(X, Y),
    female(X).

paternalgrandfather(X, Y) :-
    father(X, Z),
    father(Z, Y).

maternalgrandfather(X, Y) :-
    father(X, Z),
    mother(Z, Y).

paternalgrandmother(X, Y) :-
    mother(X, Z),
    father(Z, Y).

maternalgrandmother(X, Y) :-
    mother(X, Z),
    mother(Z, Y).

greatgrandparent(X, Y) :-
    parent(P, Y),
    grandparent(X, P).

greatgrandchild(X, Y) :-
    greatgrandparent(Y, X).

son(X, Y) :-
    child(X, Y),
    male(X).

daughter(X, Y) :-
    child(X, Y),
    female(X).

granddaughter(X, Y) :-
    grandchild(X, Y),
    female(X).

grandson(X, Y) :-
    grandchild(X, Y),
    male(X).

ancestor(X, Y) :-
    parent(X, Y).

ancestor(X, Y) :-
    parent(Z, Y),
    ancestor(X, Z).

descendant(X, Y) :-
    ancestor(Y, X).

relative(X, Y) :-
    ancestor(Z, X),
    ancestor(Z, Y).

sibling(X, Y) :-
    parent(Z, X),
    parent(Z, Y),
    X \= Y.

sister(X, Y) :-
    sibling(X, Y),
    female(X),
    X \= Y.

brother(X, Y) :-
    sibling(X, Y),
    male(X),
    X \= Y.

uncle(X, Y) :-
    brother(X, Z),
    child(Y, Z).
```

```
aunt(X, Y) :-
    sister(X, Z),
    child(Y, Z).

cousin(X, Y) :-
    grandparent(Z, X),
    grandparent(Z, Y),
    \+sibling(X, Y),
    X \= Y.

nephew(X, Y) :-
    aunt(Y, X),
    male(X);
    uncle(Y, X),
    male(X).

niece(X, Y) :-
    aunt(Y, X),
    female(X);
    uncle(Y, X),
    female(X).

stepchild(X, Y) :-
    parent(Z, X),
    partner(Z, Y),
    \+parent(Y, X).

stepfather(X, Y) :-
    stepchild(Y, X),
    male(X).

stepmother(X, Y) :-
    stepchild(Y, X),
    female(X).
```