

Assignment 8: Building a Family Tree in Prolog

Nguyen The Duy Khanh

005019181

MSCS-632 - Advanced Programming Languages

Vanessa Cooper

April 10, 2025

Table of Contents

Implementation Process	3
Sample Queries	4
Reference	4

Implementation Process

The implementation began by defining basic facts to represent gender (male, female) and parent-child relationships. These facts served as the foundation for building more complex relationships through logical rules. Derived relationships were implemented using Prolog's rule syntax:

- **grandparent** used a two-level parent inference.

```
1  % Grandparent
2  grandparent(X, Y) :-
3      parent(X, Z),
4      parent(Z, Y).
5
```

Figure 1: Grandparent implementation with Prolog

- **sibling** checked if two individuals shared at least one common parent.

```
6  % Sibling
7  sibling(X, Y) :-
8      parent(Z, X),
9      parent(Z, Y),
10     X \= Y.
11
```

Figure 2: Sibling implementation with Prolog

- **cousin** was built by checking if the parents of two individuals were siblings. To support flexible querying, additional predicates such as `children` and `siblings` used `findall` to gather lists of individuals based on logical conditions.

```
% Cousin
cousin(X, Y) :-
    parent(P1, X),
    parent(P2, Y),
    sibling(P1, P2),
    X \= Y.
```

Figure 3: Cousin implementation with Prolog

Recursive logic was introduced in the descendant rule to handle direct and indirect descendant queries. This required careful base and recursive case construction to avoid infinite loops and ensure termination.

Sample Queries

- **Children of a person:** `children(john, Children) => Children = [Paul, Lisa]`
- **Siblings of a person:** `siblings(paul, Siblings) => Children = [Lisa]`
- **Check if one is children of another:** `child(mike,paul) => true`

References

Khanh Nguyen. https://github.com/khanhntd/MSCS632_Assignment8