

Implementation of Family Tree

AMRIT KANDEL (078BEI004), PRASISH TIMALSINA (078BEI026)

Submitted to: Department of Electronics, Communications and Information

ABSTRACT In this lab, we implement a Family Tree using Python sets and lists. It allows users to add family relationships in the form of either parent or child and then automatically infers and answers questions about other relations like siblings, grandparents, grandchildren, uncles/aunts, and nephews/nieces. It stores facts as parent/child relationships and uses logical rules to derive other relations. Users can query relationships or list all related members for a person. The system helps in understanding family relationships and how logical inference can be used in simple expert systems. This project demonstrates the use of Python sets, conditionals, and functions to build an interactive, rule-based system.

INDEX TERMS Family Tree, Inference, Python

I. INTRODUCTION

A. FAMILY TREE

A family tree is a structured representation of family relationships arranged in a hierarchical format. It typically begins with an individual and expands through generations to include their descendants or ancestors. The structure visually shows how family members such as parents, children, grandparents, and siblings are connected.

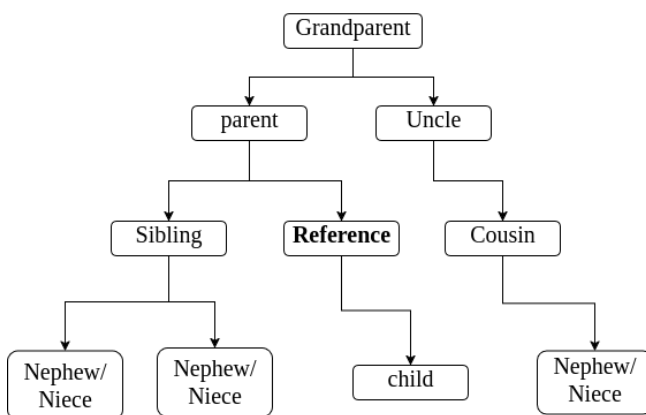


FIGURE 1. Family Tree

II. EXPERIMENTAL DETAILS

1. OBJECTIVE

The objective of this lab was to develop a Python-based system that stores and infers family relationships like parent, child, sibling, grandparent, uncle/aunt, and nephew/niece.

2. TOOLS USED

- **Programming Language:** Python
- **IDE:** VS Code

3. INPUT

Commands from the user to add facts or query relationships.

Examples: add parent A B

query grandparent B A

list sibling C

4. PROCEDURE

- The program starts and waits for user input.
- The user adds relationships using add parent or add child.
- The system stores each fact as a (parent, child) pair.
- When a query is made, the program uses logic to infer the relationship.

III. RESULTS

The results obtained is shown below:

```

Family Tree Inference System
Type 'help' for available commands.
Added fact: Ronaldo is a child of Messi.
Added fact: Mbappe is a child of Messi.
Added fact: Palmer is a child of Haaland.
Added fact: Haaland is a child of Messi.
Added fact: Pogba is a child of Ronaldo_wife
.
Added fact: Pogba is a child of Ronaldo.
Added fact: Muller is a child of Ronaldo.
Added fact: Casemero is a child of Mbappe.
Added fact: Romero is a child of Mbappe.
  
```

```

Known Facts:
  Casemero is child of Mbappe
  Haaland is child of Messi
  Mbappe is child of Messi
  Muller is child of Ronaldo
  Palmer is child of Haaland
  Pogba is child of Ronaldo
  Pogba is child of Ronaldo_wife
  Romero is child of Mbappe
  Ronaldo is child of Messi
Grandchildren(s) of Messi: Casemero, Muller,
  Palmer, Pogba, Romero
Unknown command. Type 'help' for options.

Available commands:
  add parent [A] [B]      : A is parent of B
  add child [A] [B]       : A is child of B
  query [rel] [A] [B]     : Is A [rel] of B?
  list [rel] [A]          : List all [rel]s
                           of A
  show                    : Show all known
                           facts
  exit                    : Quit the program

Valid relations:
  parent, child, sibling, grandparent,
  grandchild, uncle_or_aunt,
  nephew_or_niece

Query: Pogba is a sibling of Muller: Yes
Query: Casemero is a sibling of Palmer: No
Query: Haaland is a uncle_or_aunt of
  Casemero: Yes
...Exiting.

```

IV. DISCUSSION AND CONCLUSION

In this experiment, we developed a Family Tree Inference System using Python that allows users to add and query family relationships. The system stores basic facts like "A is the parent of B" and uses logical rules to infer extended relationships such as siblings, grandparents, grandchildren, uncles/aunts, and nephews/nieces. For example, after adding that A is the parent of B and B is the parent of C, the system correctly infers that A is the grandparent of C. Similarly, if A and B have the same parent, they are identified as siblings. The program uses simple data structures like sets and applies condition-based logic to reason about relationships. The results showed that the system responds accurately to various user queries and provides meaningful relationship outputs. Overall, this experiment demonstrated how rule-based logic and programming can be used to model and infer complex family relationships in a simple and effective way.