**Assignment 3**

**Title:**

Parsing a Family Tree Using a Knowledge Base and Inference Rules

**Aim:**

To parse a family tree using a knowledge base of facts and inference rules, and to query relationships such as parent, sibling, cousin, grandparent, etc.

**Objectives:**

1. To represent a family tree using a formal knowledge base with facts and relationships.
2. To implement parsing and querying over the family tree.
3. To use logical inference rules to deduce relationships such as sibling, cousin, and grandparent.
4. To analyze the efficiency of querying and parsing complex family structures.

**Theory:**

Family Tree Representation:

A family tree is a hierarchical structure that represents relationships between individuals, such as parent-child relationships, and extends to other relationships like siblings, cousins, grandparents, etc.

Knowledge Base:

A knowledge base (KB) is a collection of facts and rules about a particular domain (in this case, the family tree). The facts are assertions about specific relationships (e.g., “John is the parent of Alice”), while rules allow inference of new information based on known facts.

Examples of Facts:

* parent(John, Alice) (John is Alice's parent)
* parent(Mary, Alice) (Mary is Alice's parent)
* male(John) (John is male)
* female(Mary) (Mary is female)

**Examples of Inference Rules:**

* Sibling Rule: Two individuals are siblings if they share at least one parent.

sibling(X, Y) :- parent(P, X), parent(P, Y), X != Y.

* Grandparent Rule: An individual is a grandparent if they are the parent of someone's parent.

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

Procedure:

1. Define the Knowledge Base:

* Define facts about family members using relationships like parent, male, female, etc.
* Example family structure:

John and Mary are parents of Alice and Bob.

Robert and Sarah are parents of John.

* Represent these facts in the KB:

parent(John, Alice).

parent(Mary, Alice).

parent(John, Bob).

parent(Mary, Bob).

parent(Robert, John).

parent(Sarah, John).

* Include gender information

male(John).

female(Mary).

male(Bob).

1. Define Inference Rules:

* Parent Rule: Already defined through facts.
* Sibling Rule: Two individuals are siblings if they have the same parents

sibling(X, Y) :- parent(P, X), parent(P, Y), X != Y.

* Grandparent Rule: A person is a grandparent if they are the parent of a parent

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

* **Cousin Rule:** Two people are cousins if their parents are siblings.

cousin(X, Y) :- parent(A, X), parent(B, Y), sibling(A, B).

1. Parsing the Family Tree:

* Create a function to parse the knowledge base and generate a family tree.
* Start by identifying all individuals and their relationships.
* Use the rules to deduce new relationships, such as siblings, cousins, grandparents, etc.

1. Query the Knowledge Base:

* Implement functions to query specific relationships from the family tree:
* Who are Alice’s siblings?
* sibling\_query(Alice) -> [Bob]
* Who are Bob's grandparents?
* grandparent\_query(Bob) -> [Robert, Sarah]
* Who are Alice's cousins?
* cousin\_query(Alice) -> [] (if no cousins in this simple family tree)

1. Adding More Complexity:

* Expand the family tree by adding more generations and individuals.
* Add more complex relationships such as aunts, uncles, and in-laws.
* Update the knowledge base and rules accordingly.

1. Performance and Efficiency:

* Test the efficiency of querying the knowledge base for complex family trees.
* Measure the time taken to infer relationships with increasing family sizes.
* Explore optimizations, such as caching inferred relationships or indexing the KB.

Expected Output:

1. A structured family tree representation in the form of a knowledge base (facts and rules).
2. Successful queries demonstrating the relationships between family members.
3. For example:
   1. Querying Alice’s siblings should return Bob.
   2. Querying Bob’s grandparents should return Robert and Sarah.
4. Deduction of new relationships (e.g., cousins, grandparents) using logical inference rules.
5. Efficient querying of family relationships as the family tree grows in complexity.

**Procedure for Example Problem (Family Tree):**

1. **Define Facts:**
   * Represent the family tree with facts (e.g., parent, male, female).
2. **Define Inference Rules:**
   * Implement rules to infer relationships like siblings, cousins, and grandparents.
3. **Parse the Knowledge Base:**
   * Traverse the family tree to deduce relationships using inference rules.
4. **Querying:**
   * Execute queries to determine specific relationships such as siblings, grandparents, or cousins.

**Conclusion:**

In this lab, we parsed a family tree using a knowledge base and inference rules. Using facts and rules, we were able to define and query relationships between family members such as siblings, grandparents, and cousins. The use of logical inference allowed us to deduce new relationships from known facts, demonstrating the power of knowledge-based systems in handling hierarchical family structures.