|  |  |
| --- | --- |
| **EXPT NO:3** | **First Order Logic and Inference (Chaining and Resolution)** |
| **DATE: 24.09.2025** |

**PRE-LAB QUESTIONS (PROVIDE BRIEF ANSWERS TO THE FOLLOWING QUESTIONS)**

1. **What is the difference between propositional logic and first-order logic?**

Propositional logic represents facts as indivisible propositions, while first-order logic introduces quantifiers and predicates, allowing representation of relationships and properties among objects, making it more expressive.

1. **Define a predicate and give an example in the context of family relationships.**

A predicate expresses properties or relations of objects. Example: Parent(x, y) means "x is a parent of y," useful for modeling family relationships like Parent(John, Mary).

1. **What is unification in First-Order Logic and why is it important in inference?**

**U**nification is the process of finding substitutions that make different logical expressions identical. It is essential in inference for matching predicates, enabling automated reasoning and rule application.

1. **Differentiate between Forward Chaining and Resolution in First-Order Logic.**

Forward chaining applies inference rules repeatedly from known facts to derive conclusions. Resolution refutes by contradiction, combining clauses systematically until inconsistency or the desired conclusion emerges.

1. **Why is First-Order Logic more expressive than propositional logic? Illustrate with a simple example.**

First-order logic is more expressive because it quantifies over individuals. Example: Propositional: "All humans are mortal" is many separate facts; FOL: ∀x (Human(x) → Mortal(x)).

**IN-LAB EXERCISE**

**OBJECTIVE:**

To implement a Knowledge-Based Agent using First-Order Logic for reasoning in a family relationships domain and demonstrate inference using Forward Chaining and Resolution.

**PROCEDURE:**

**1. Scenario:**

* **Problem Statement:**Given family relationships, infer new relationships using FOL-based inference.
* **Facts:**

Parent(John, Mary)

Parent(Mary, Alice)

Parent(John, Mark)

Parent(Mark, Sam)

Male(John), Male(Mark), Male(Sam)

Female(Mary), Female(Alice)

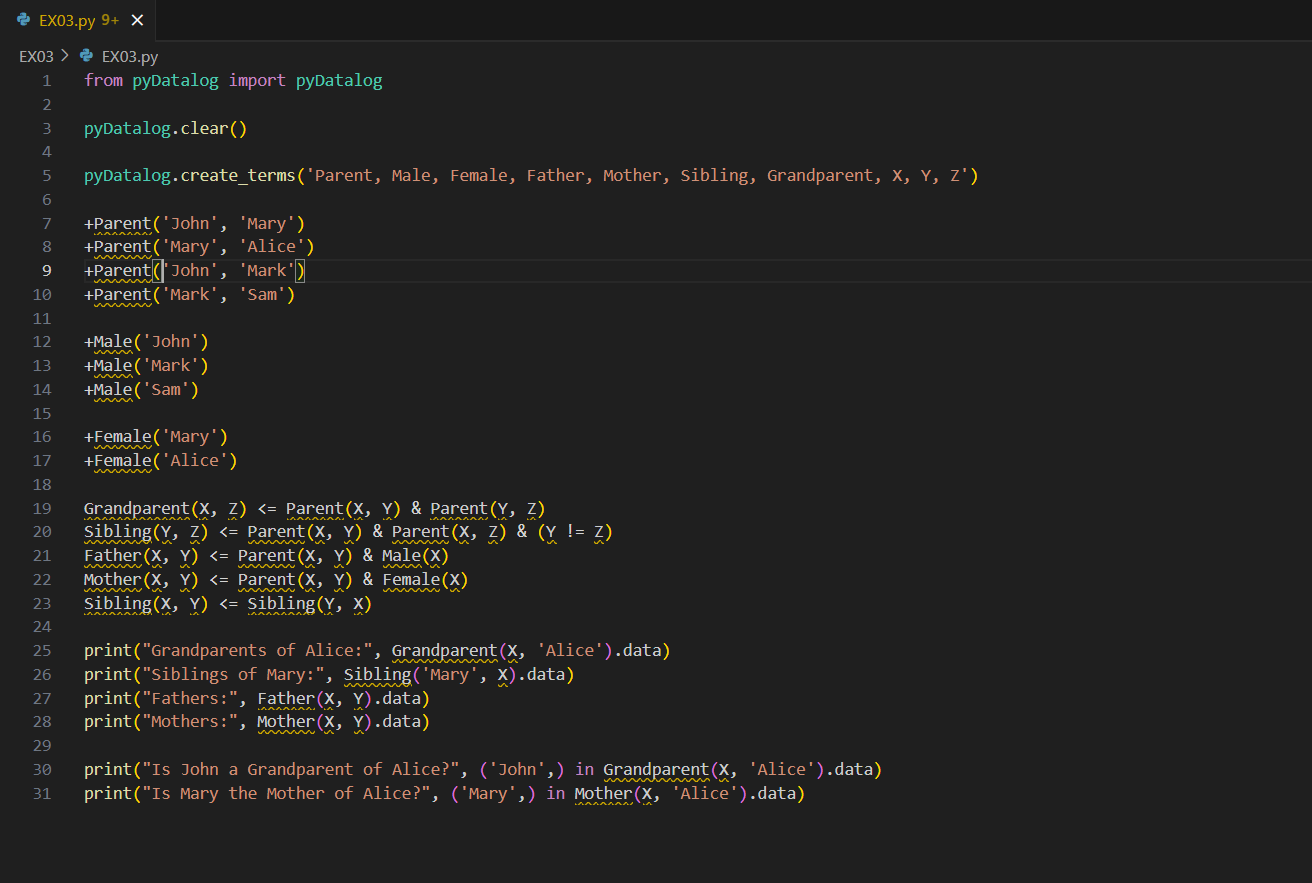
**Rules:**

1. Grandparent Rule: Parent(x, y) ∧ Parent(y, z) → Grandparent(x, z)
2. Sibling Rule: Parent(x, y) ∧ Parent(x, z) ∧ y ≠ z → Sibling(y, z)
3. Father Rule: Parent(x, y) ∧ Male(x) → Father(x, y)
4. Mother Rule: Parent(x, y) ∧ Female(x) → Mother(x, y)
5. Symmetric Sibling: Sibling(x, y) → Sibling(y, x)

**2. Implementation Steps:**

* Define predicates using **pyDatalog.**
* Encode rules using FOL.
* Perform inference by querying the knowledge base.

**SCREENSHOT OF CODE**

****

**SCREENSHOT OF OUTPUT**

**A computer screen with white text

AI-generated content may be incorrect.**

**POST-LAB QUESTIONS (PROVIDE BRIEF ANSWERS TO THE FOLLOWING QUESTIONS)**

1. **How would the system infer that “John is the grandfather of Alice”? What steps are involved in this derivation?**

The system applies rules: Parent(John, Bob) ∧ Parent(Bob, Alice) → Grandfather(John, Alice). It matches facts, unifies variables, and derives Grandfather(John, Alice) through inference chaining.

1. **If a new fact Female(Mark) is added by mistake, how will it affect the inference of Father and Mother?**

Adding Female(Mark) wrongly may confuse inference: rules like Mother(x,y) ← Parent(x,y) ∧ Female(x) could incorrectly infer Mother(Mark,y) for some child, producing false conclusions.

1. **What is the limitation of Forward Chaining in terms of computational efficiency?**

Forward chaining can be computationally inefficient because it explores all applicable rules exhaustively, generating many irrelevant intermediate facts, leading to combinatorial explosion and unnecessary computations before reaching goals.

1. **How does Resolution differ from Chaining in First-Order Logic? When is Resolution preferred?**

Resolution works by contradiction, combining clauses until deriving false, proving the query. Chaining is goal- or data-driven. Resolution is preferred for completeness in theorem proving and refutations.

1. **Explain how unification helps in resolving queries in a First-Order Logic knowledge base.**

Unification matches query variables with knowledge base facts, enabling substitution. This lets the system connect abstract rules with specific instances, resolving queries efficiently and supporting generalized logical inference.

**RESULT:**

Hence, the First-Order Logic (FOL) based Knowledge-Based Agent was successfully implemented. The agent was able to infer new facts such as grandparent and sibling relationships using forward chaining and resolution techniques. This experiment demonstrates the expressive power of FOL and its effectiveness in automated reasoning for domains involving complex relationships.

**ASSESSMENT**

|  |  |  |
| --- | --- | --- |
| **Description** | **Max Marks** | **Marks Awarded** |
| Pre Lab Exercise | **5** |  |
| In Lab Exercise | **10** |  |
| Post Lab Exercise | **5** |  |
| Viva | **10** |  |
| **Total** | **30** |  |
| **Faculty Signature** | |  |