Experiment no. 8

OMKAR SHIVARKAR	121A9057
B3	AIML

Aim: Write a program to implement first order Logic in python

Program 1 and Output:

In [2]:

```
class Predicate:
    def __init__(self, name, arity):
        self.name = name
        self.arity = arity

    def __str__(self):
        return f"{self.name}/{self.arity}"

class Atom:
    def __init__(self, predicate, arguments):
        self.predicate = predicate
        self.arguments = arguments

    def __str__(self):
        return f"{self.predicate.name}({', '.join(map(str, self.arguments))})"

class KnowledgeBase:
    def __init__(self):
        self.facts = []

    def add_fact(self, fact):
        self.facts.append(fact)

    def query(self, query):
        for fact in self.facts:
            if fact.predicate == query.predicate and fact.arguments == query.arguments:
                return True
        return False

if __name__ == "__main__":
    # Define predicates
    is_human = Predicate("is_human", 1)
    is_mortal = Predicate("is_mortal", 1)

    # Create atoms (facts)
    facts = [
        Atom(is_human, ["John"]),
        Atom(is_mortal, ["John"]),
    ]

    # Create a knowledge base
    kb = KnowledgeBase()
    for fact in facts:
        kb.add_fact(fact)

    # Query the knowledge base
    query1 = Atom(is_human, ["John"])
    query2 = Atom(is_mortal, ["John"])

    if kb.query(query1):
        print(f"{query1} is true.")
    else:
        print(f"{query1} is false.")

    if kb.query(query2):
        print(f"{query2} is true.")
    else:
        print(f"{query2} is false.")
```

```
is_human(John) is true.
is_mortal(John) is true.
```

Program 2 and Output:

```
class Person:
    def __init__(self, name, gender):
        self.name = name
        self.gender = gender

# Define some individuals
alice = Person("Alice", "female")
bob = Person("Bob", "male")
carol = Person("Carol", "female")
david = Person("David", "male")

# Define a predicate for the parent relationship
def is_parent(parent, child):
    # Implement your specific parent-child relationship logic here
    pass

# Define the facts about the parent-child relationships
parent_child_facts = [
    is_parent(alice, bob),
    is_parent(alice, carol),
    is_parent(bob, david)
]

# Define a predicate for the gender of a person
def has_gender(person, gender):
    return person.gender == gender

# Define some facts about gender
gender_facts = [
    has_gender(alice, "female"),
    has_gender(bob, "male"),
    has_gender(carol, "female"),
    has_gender(david, "male")
]

# Example query: Is Alice a parent?
print("Is Alice a parent?", is_parent(alice, bob))

# Example query: Is Bob a parent?
print("Is Bob a parent?", is_parent(bob, carol))
```

```
Is Alice a parent? None
Is Bob a parent? None
```

Conclusion: Hence we successfully implemented first order Logic in python