Lab program 10

Create a knowledge base consisting of first order logic statements and prove the given query using forward reasoning

Algorithm

1. Initialize knowledge Base (KB)
   Start with an empty knowledge base

2. Add FOL statements to KB
   Add relevant FOL statements to the knowledge base. These statements represent facts and rules about the domain

3. Define forward reasoning rules
   Specify the rules for forward reasoning

4. Initialize working memory
   Create a working memory to store the current state of the knowledge base during the reasoning process.

5. Ask Query
   Formulate the query you want to prove

6. Forward reasoning loop

Repeat the following steps until the query

a. Iterate through each rule.

b. Apply rules whose conditions match the current state

c. Add the conclusions of the applied

4. Check query in working memory
Verify if query is now proved or can't be inferred

5. output result

If the query is proven output "Query is true" otherwise output Query is false.

Code

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

    def add_rule(self, conditions, conclusion):
        self.rules.append({"conditions":
            set(conditions), "conclusion":
            conclusion})
```

```
def forward_reasoning(self, query):
    working_memory = set()
    unchanged = False

    while not unchanged:
        unchanged = True
        for rule in self.rules:
            if rule['conditions'].issubset(working_memory) and rule['conclusion'] not in working_memory:
                working_memory.add(rule['conclusion'])
                unchanged = False
    return query in working_memory

kb = KnowledgeBase()
kb.add_rule(["P", "Q"], "R")
kb.add_rule(["R", "S"], "T")
kb.add_rule(["T", "U"], "V")

query = "V"

result = kb.forward_reasoning(query)

if result:
    print("Query is true")
else:
    print("Query is false")
```

Output

Enter the query.

(["P", "Q"], "R")
(["R", "S"], "T")
(["T", "U"], "V")

Query is false.

Enter the query

(["P", "Q"], "R")
(["Q", "R",], "P")
(["R", "P",], "Q")

Query is true.