

5. IMPLEMENTATION OF VARIOUS KNOWLEDGE REPRESENTATION TECHNIQUES.

Contents

Introduction:	2
Propositional Logic:.....	2
2. First-Order Logic:.....	2
3. Frames:.....	3
4. Semantic Networks:	4
5. Ontologies:	4
6. Rule-Based Systems:	5

Introduction:

Knowledge representation in artificial intelligence involves capturing and organizing information in a way that can be used for reasoning and problem-solving. Here are implementations of various knowledge representation techniques:

Propositional Logic:

```
```python
class PropositionalLogic:
 def __init__(self):
 self.knowledge_base = set()

 def add_sentence(self, sentence):
 self.knowledge_base.add(sentence)

 def ask_query(self, query):
 return query in self.knowledge_base

Example usage:
pl = PropositionalLogic()
pl.add_sentence("P -> Q")
pl.add_sentence("~Q")
print(pl.ask_query("~P"))
```
```

2. First-Order Logic:

```
```python
from sympy import symbols, Implies, Not, ask, satisfiable

class FirstOrderLogic:
 def __init__(self):
```

```

self.knowledge_base = set()

def add_sentence(self, sentence):
 self.knowledge_base.add(sentence)

def ask_query(self, query):
 return ask(query, set(self.knowledge_base))

Example usage:
x, y = symbols('x y')
fol = FirstOrderLogic()
fol.add_sentence(Implies(x, y))
fol.add_sentence(Not(y))
print(fol.ask_query(Not(x)))
...

```

### 3. Frames:

```

``python
class AnimalFrame:
 def __init__(self, name, habitat, sound):
 self.name = name
 self.habitat = habitat
 self.sound = sound

Example usage:
lion = AnimalFrame("Lion", "Grasslands", "Roar")
print("Name:", lion.name)
print("Habitat:", lion.habitat)
print("Sound:", lion.sound)
...

```

#### 4. Semantic Networks:

```
```python
class SemanticNetwork:
    def __init__(self):
        self.nodes = {}
        self.edges = set()

    def add_node(self, node):
        self.nodes[node] = set()

    def add_relation(self, node1, relation, node2):
        self.edges.add((node1, relation, node2))
        self.nodes[node1].add((relation, node2))
        self.nodes[node2].add((relation, node1))

# Example usage:
sn = SemanticNetwork()
sn.add_node("Cat")
sn.add_node("Mammal")
sn.add_relation("Cat", "is-a", "Mammal")
print(sn.nodes)
print(sn.edges)
```
```

#### 5. Ontologies:

```
```python
from owlready2 import Thing, ObjectProperty, onto
```

```

# Define ontology
onto = onto.Ontology("http://example.org/ontology#")

# Define classes and properties
class Animal(Thing):
    pass

class Mammal(Animal):
    pass

class hasLegs(ObjectProperty):
    domain = [Animal]
    range = [int]

# Example usage:
lion = Animal("lion")
lion.hasLegs = 4
print(lion.hasLegs)
...

```

6. Rule-Based Systems:

```

``python
from pyknow import KnowledgeEngine, Fact

class AnimalRuleEngine(KnowledgeEngine):
    @Rule(Fact(kind='mammal'))
    def mammal_rule(self):
        print("It's a mammal.")

    @Rule(Fact(legs=4))

```

```
def four_legs_rule(self):  
    print("It has four legs.")
```

Example usage:

```
engine = AnimalRuleEngine()  
engine.reset()  
engine.declare(Fact(kind='mammal', legs=4))  
engine.run()
```