Writeup (AI lab)

Amber Nitin
1BM18CS013
AI lab - 2

Program - 5

```
import re

def isVariable(x):
    return len(x)==1 and x.islower() and x.isalpha()

def getAttributes(s):
    expr = "\([^)]+\)"
    m = re.findall(expr, s)
    return m

class Fact:

    def __init__(self,expression)
        self.expression = expression
        predicate, params = self.splitExpression(expression)
        self.predicate = predicate
        self.params = params
        self.result = any(self.getConstants())
```

Scanned with CamScanner

Ambee Mistri
1BM18CS0..
AI lab

```
def splitExpressions(self, e):
    f = getPredicate(e)[0]
    params = getAttribute(e)[0].strip('()').split(',')
    return [f, params]

def getResult(self):
    return self.result

def getConstants(self):
    return [None if isVariable(c) else c for c in
            self.params]

def getVariables(self):
    return [v if isVariable(v) else None for v in
            self.params]

def substitute(self, constants):
    c = constants.copy()
    f = f"{self.predicate}({','.join([constants.
        pop(0) if isVariable(p) else p for
```

Scanned with CamScanner

Ambeer Mishra
IBM 18 CS 013
AI lab

```
        f in self.params]})"
    return Fact(f)

class Implication:
    def __init__(self, s):
        self.s = s
        l = s.split('=>')
        self.lhs = [Fact(f) for f in l[0].split('&')]
        self.rhs = Fact(l[1])

    def evaluate(self, facts):
        constants = {}
        new_lhs = []
        for fact in facts:
            for val in self.lhs:
                if val.predicate == fact.predicate:
                    for i, v in enumerate(val.getVariables()):
                        if v:
                            constants[v] = fact.getConstants()[i]
                    new_lhs.append(fact)
```

Scanned with CamScanner

Ambee Mishra
1BM18CS013
AI lab

```
predicate, attributes = getPredicates(self.rhs.expression)[0],
str(getAttributes(self.rhs.e)[0])

for key in constants:

    if constants[key]:
        attributes = attributes.replace(key, constants[key])

    expr = f'{predicate}{attributes}'
    return Fact(expr) if len(new_lhs) and all
([f.getResult() for f in new_lhs]) else None.

class Kb:
    def __init(self):

        self.facts = set()
        self.implications = set()

    def tell(self, e):

        if "=>" in e:
            self.implications.add(Implication(e))
        else:
            self.facts.add(Fact(e))
```

Scanned with CamScanner

Ambar Mishra
1BM18CS013
AI lab

```
        for i in self.inflications:
            res = i.evaluate(self.facts)
            if res:
                self.facts.add(res)

    def query(self, e):
        facts = set([f.expression for f in self.facts])
        i = 1
        print(f"Querying {e}:")
        for f in facts:
            if fact(f).predicate == fact(e).predicate:
                print(f"\t{i}. {f}")
                i += 1

    def display(self):
        print("All facts:")
        for i, f in enumerate(set([f.expressions for f in
                                   self.facts])):

def main():
    kb = Kb()
```

Amber Mishra
1BM18CS013
AI lab.

```
print("Enter KB:")
print("Enter number of expressions")
n = int(input())
for i in range(n):
    fact = input()
    kb.tell(fact)
print("Enter query")
q = input()
kb.ask(query)
kb.display()
main()
```

Scanned with CamScanner