

AI LAB TEST-2

Jurraj Singh
18M18CS126

Pgm-3

```
import re
def isVariable(m):
    return len(m) == 1 and ...
def getAttributes(s):
    expr = '\((.*)\)'
    matches = re.findall(expr, s)
    return matches
def getPredicates(s):
    expr = ...
```

class Fact:

```
def __init__(self, expr):
    self.expression = expression
    predicate, params = self.split_expr()
    self.predicate = predicate
    self.params = params
    self.result = any(self.get_constants())
```

```
def split_expression(self, expr):
    predicate = ...
    params = getAttributes(expr[0].strip())
    return (predicate, params)
```

```
def getResult
def getConstants
```

(2)


```

def getvariables(self):
def substitute(self, constants):
    c = constant-copy()
    f = f + self.predicate(x', 'join(constants,
    pop(0) if variable(p) else p for p in self.params
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))
        for i in self.implications:
            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, l
    print(f'querying {e}:')
    for f in facts:

```

②


```

    fact(f).predicate = fact(e).predicate
    print(f' {f} + {13} = {f} ')
    i += 1

def display(self):
    print("All facts: ")
    for i, f in enumerate([set([f.expr for f in
                                self.facts]))):
        print(f' {f} + {13} = {f} ')

```

class implication:

```

def __init__(self, expr):
    self.expression = expression
    l = expression.split(' = ')
    self.lhs = [fact(f) for f in l[0].split('&')]
    self.rhs = fact(l[1])

```

def evaluate(self, facts):

```

    constants = {3}
    new_lhs = []

```

for fact in facts

for val in self.lhs:

if val.predicate == fact.predicate
 if i, v in enumerate(val.get_variables())

③

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

expr = f' {predicates} {attribute}'
return fact(expr) if len(new-lys) and
all([f.getResult for f in new-lys])
else None.

