AI LAB TEST - 2

S.R. PUNEETH
1BM18CS087
1/1/21

Q. Design forward reasoning system to prove query "Someone who are intelligent cannot read", using forward chaining

```
class ~~Implication~~ Fact
    def __init__(self, expression):
        self.expression = expression
        ~~l = expression.split('=>')~~
        ~~self.lhs = [Fact~~
        ~~self.rhs~~
        predicate, params = self.splitExpression(~~e~~expression)
        self.predicate = predicate
        self.params = params
        self.result = any(self.getConstants())
    def splitExpression(self, expression):
        predicate = getPredicates(expression)[0]
        params = getAttributes(expression)[0].strip('()').split(',')
        return [predicate, 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(~~std~~ self, constants):
        c = constants.copy()
        f = f"{self.predicate}({','.join([(constants.pop(0) if
                isVariable(p) else p for p in self.params])})"
        return fact(f)
```


S.R. Puneeth

Scanned with CamScanner

S·R·PUNEETH
1BM18CS087

```
class Implication
    def __init__(self, expression):
        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 = {}
        new_lhs = []
        for fact in facts:
            for val in self.lhs:
                if val.predicate == fact.predicate:
                    if v:
                        constants[v] = fact.getConstants()[i]
                    new_lhs.append(fact)
        predicate, attributes = getPredicates(self.rhs.expression)[0],
        str(getAttributes(self.rhs.expression)[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
```


S.R.Puneeth

Scanned with CamScanner