Page :
Date: / /

ABHINAV VERMA
IBM18CS003

Abhinav
29-12-2020

AI Lab test-2

3.

code

```
import re

def negate(term):
    return f'~{term}' if term[0] != '~' else
        term[1]

def reverse(clause):
    if len(clause) > 2:
        t = split_terms(clause)
        return f'{t[1]} v {t[0]}'
    return ''

def split_terms(rule):
    exp = '(~*[PQRS])'
    terms = re.findall(exp, rule)
    return terms

def contradiction(query, clause):
    contradictions = [f'{query} v {negate(query)}',
                      f'{negate(query)} v {query}']

    return clause in contradictions or
```

Page :
Date: / /

reverse(clause) in contradiction

```
def resolve(kb, query):
    temp = kb.copy()
    temp += [negate(query)]
    steps = dict()

    for rule in temp:
        step[rule] = 'Given'
    steps[negate(query)] = 'Negated conclusion'

    i = 0
    while i < len(temp):
        n = len(temp)
        j = (i+1) % n
        clauses = []
        while j != i:
            terms1 = split_terms(temp[i])
            terms2 = split_terms(temp[j])
            for c in terms1:
                if negate(c) in terms2:
                    t1 = [t for t in terms1
                          if t != c]

                    t2 = [t for t in terms2
                          if t != negate(c)]
```

Abhinav

Page :
Date: / /

```
gen = t1 + t2
if len(gen) == 2:
    if gen[0] != negate(gen[1]):
        clauses += [f'{gen[0]} v {gen[1]}']

else:
    if contradiction(query, f'{gen[0]} v {gen[1]}'):

        temp.append(f'{gen[0]} v {gen[1]}')

        steps[''] = f"Resolved {temp[i]} and {temp[j]} to {temp[-1]}"

        return steps
elif len(gen) == 1:
    clauses += [f'{gen[0]}']
else:
    if contradiction(query, f'{terms1[0]} v {terms2[0]}'):

        temp.append(clause)
        steps[clause] = f'Resolved from {temp[i]} and {temp[j]}'

        j = (j+1) % n
    i += 1
return steps
```

Ashwini

Page :
Date: / /

```
def resolution(kb, query):
    kb = kb.split(';')
    steps = resolve(kb, query)
    print('\nStep\t|Clause\t|Derivation\t')
    print('-' * 30)
    i = 1
    for step in steps:
        print(f'{i}.\t|{step}\t|{steps[step]}\t')
        i += 1

def main():
    print("Enter the kb:")
    kb = input()
    print("Enter the query:")
    query = input()
    resolution(kb, query)
```

Abhinav