29-12-2020

AI lab test-2

B Praneeth
1BM18CS023

3. Resolution

```
import re

def resolve (kb, query)
temp = kb.copy()
tempt = [negation(query)]
steps = dict()
for rule in temp:
        steps[rule] = 'Given'
steps[negation(query)] = 'Negated query'
i=0
while i < len(temp):
        n= len(temp)
        j= (i+1)%n
        clauses = [ ]
        while j>i
                t1 = split_terms(temp[i])
                t2= split_terms(temp[j])
```


B Pran

```
for c in t1:
    if negation(c) in t2:
        t1 = [t for t in ~~terms~~1 t1 if t!=c]
        t2= [ x for x in t2 if x != negation(c)]
        gen = t1 + t2
        if len(gen) == 2:
            if gen[0] != negation (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]}, which is null

                return steps

        elif len(gen) == 1:
            clauses += [f'{gen[0]}'
```

```
        else:
            if contradiction(query, f'{terms1[0]} v {t2[0]}'):
                temp.append(f'{terms1[0]} v {t2[0]}')
                steps[' '] = f"Resolved {temp[i]} and {temp[j]}
    for clauses in clauses:
        if clause not in temp and clause != reverse(clause) and
        reverse(clause) not in temp:
            temp.append(clause)
            steps[clause] = f'Resolved from {temp[i]} and
                                {temp[j]}.'
        j = (j+1) %n
    i += 1
    return steps

def negation(term):
    return f'[~term]' if term[0] != '~' else term[1]
```

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

    return clause in contradictions or reverse(clause) in
    contradictions

def resolution(kb, query):
    kb = kb.split('  ')
    steps = resolve(kb, query)
    print('Step    Clause    Derivation')
    i = 1
    for step in steps:
        print(f'{i}    {step}    {steps[step]}')
        i += 1

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


BPraneeth