AI LAB - 2

P SAI DEEKSHITH
1BM18CS148

01/01/21

2nd Program.

Given (KB): A => B and C => D

Query : A v C => B v D

Program :-

```
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}']
```

P. SNDEEKSHITH
1BM18CS148

return clause in contradictions or reverse(clause) in contradictions

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

P.SAI DEEKSHITH
1BM18CS148

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

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

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

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

            step[' '] = {f''
            Resolved from step[i]
            and {temp[j]}

            return step.

else if (len(gen) == 1):
    clause+ = [f'{gen[0]}{}']
else:
    if contradiction (query, f'{terms1[0]}{} v
```

3

P. SAI PEEKSHITH
1BM18CS148

```
{terms2[0]}').
temp.append(f'{terms1[0]}v{terms2[0]}')
steps[' '] = f"Resolved {temp1[i]} and
              {temp2[j]} to {temp[-1]}.

return steps.

for clauses in clauses:
    if clause not in clauses and clause != reverse(clause):
        temp.appen(clause)
        steps[clause] = f'Resolved from
                  {temp[i]} and {temp[j]}.'

    j = (j+1) % n.

    i = i+1

return steps.
```

P. SAI PEEKSHITH
1BM18CS148

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

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

if __name__ == "__main__":
    main()
```