

replit.com/samirkumar111/RosyEnormousSpools

Run

UpgradeShare

Files

main.py

```
main.py
1
2
3 def negate(term):
4     return f'({term})' if term[0] != '(' else term[1]
5
6 def reverse(clause):
7     if len(clause) > 2:
8         t = split_terms(clause)
9         return f'({t[1]})v({t[0]})'
10    return ''
11
12 def split_terms(rule):
13     exp = '({*[ABCD])'
14     terms = re.findall(exp, rule)
15     return terms
16
17
18 def contradiction(goal, clause):
19     contradictions = [ f'({goal})v({negate(goal)})', f'({negate(goal)})v({goal})' ]
20     return clause in contradictions or reverse(clause) in contradictions
21
22
23 def resolve(rules, goal):
24     temp = rules.copy()
25     temp += [negate(goal)]
26     steps = dict()
27     for rule in temp:
28         steps[rule] = 'Given.'
29     steps[negate(goal)] = 'Negated conclusion.'
30     i = 0
31     while i < len(temp):
```

Console

Shell

```
Enter the kb:
~AvB ~CvD
Enter the query:
~A^~CvBvD

Step | Clause | Derivation
-----|-----|-----
1. | ~AvB | Given.
2. | ~CvD | Given.
3. | ~A^~CvBvD | Negated conclusion.
```

replit.com/samirkumar111/RosyEnormousSpools

Run

UpgradeShare

Files

main.py

```
main.py
45
46
47     clauses += [f'({gen[0]})v({gen[1]})']
48     else:
49         if contradiction(goal, f'({gen[0]})v({gen[1]})'):
50             temp.append(f'({gen[0]})v({gen[1]})')
51             steps[''] = f'Resolved {temp[i]} and {temp[j]} to {temp[-1]}, which is in turn null. \nA contradiction is found when {negate(goal)} is assumed as true. Hence, {goal} is true.'
52             return steps
53         elif len(gen) == 1:
54             clauses += [f'({gen[0]})']
55         else:
56             if contradiction(goal, f'({terms1[0]})v({terms2[0]})'):
57                 temp.append(f'({terms1[0]})v({terms2[0]})')
58                 steps[''] = f'Resolved {temp[i]} and {temp[j]} to {temp[-1]}, which is in turn null. \nA contradiction is found when {negate(goal)} is assumed as true. Hence, {goal} is true.'
59                 return steps
60             for clause in clauses:
61                 if clause not in temp and clause != reverse(clause) and reverse(clause) not in temp:
62                     temp.append(clause)
63                     steps[clause] = f'Resolved from {temp[i]} and {temp[j]}.'
64             j = (j + 1) % n
65             i += 1
66             return steps
67
68 def mainn():
```

Console

Shell

```
Enter the kb:
~AvB ~CvD
Enter the query:
~A^~CvBvD

Step | Clause | Derivation
-----|-----|-----
1. | ~AvB | Given.
2. | ~CvD | Given.
3. | ~A^~CvBvD | Negated conclusion.
```

replit.com interface showing a Python script and its execution output.

Files: main.py

main.py:

```
1 def negate(term):
2     return f'
3     {term}' if
4     term[0] != '~'
5     else term[1]
6
7 def reverse
8     (clause):
9     if len(clause)
10        > 2:
11        t =
12        split_term
13        s(clause)
14        return f'
15        {t[1]}v{t
16        [0]}'
17        return ''
18
19 def split_terms
20     (rule):
21     exp = '(~*
22     [ABCD])'
23     terms =
24     re.findall
25     (exp, rule)
26     return terms
27
28 def contradiction
```

Console:

Enter the kb:
~AvB ~CvD
Enter the query:
~A~CvBvD

Step	Clause	Derivation
1.	~AvB	Given.
2.	~CvD	Given.
3.	~A~CvBvD	Negated conclusion.

15:12
01/01/2021