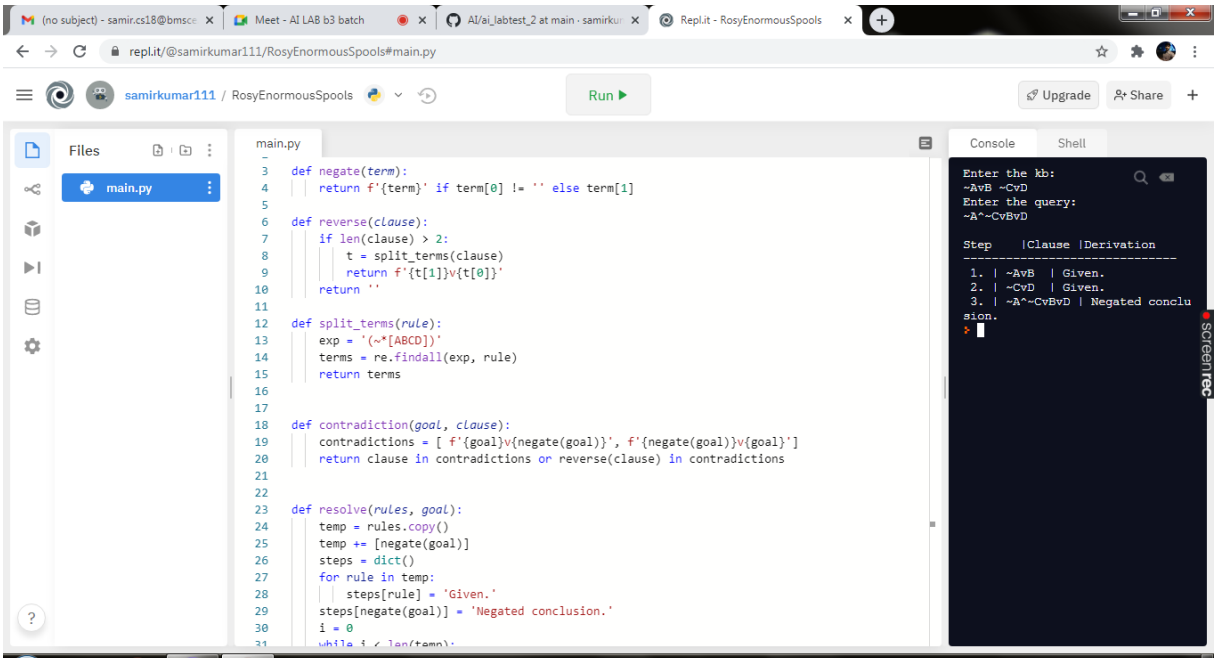
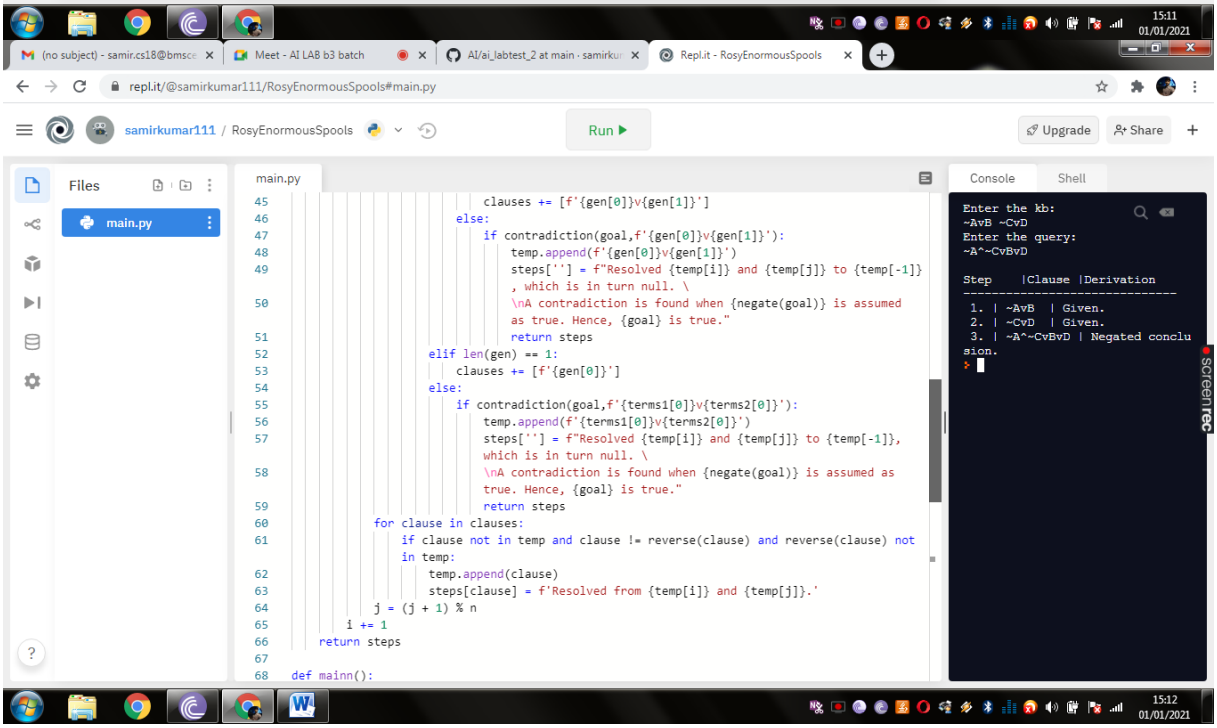
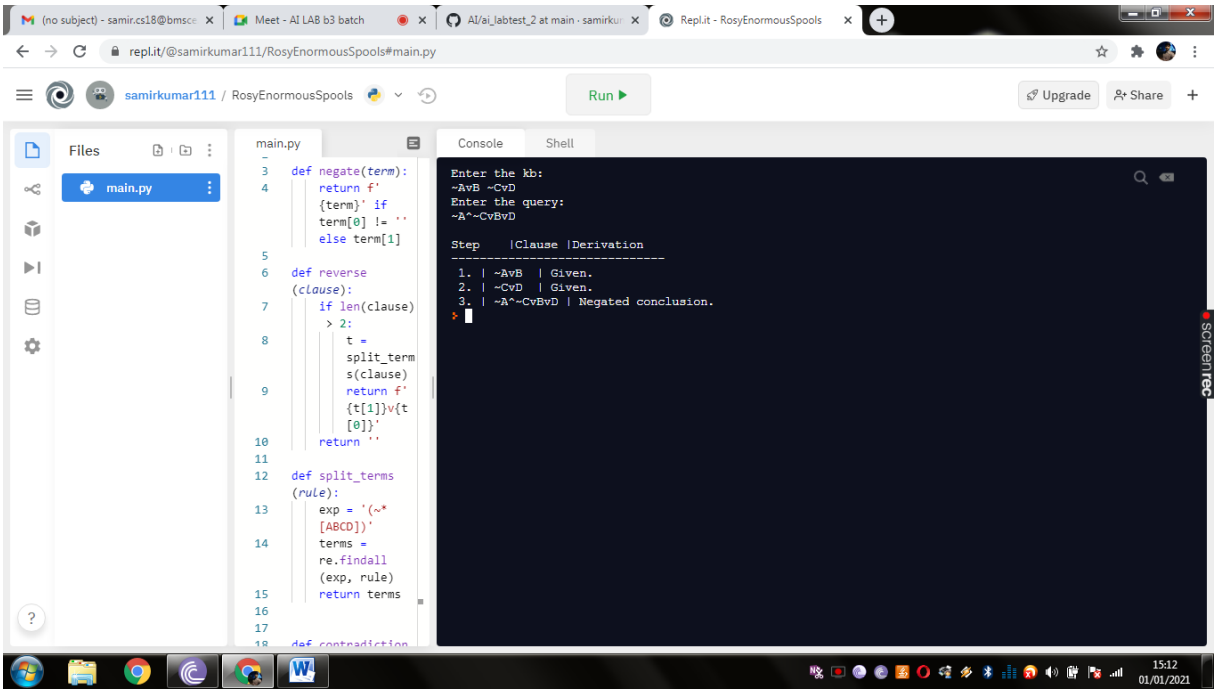

(no subject) - samir.cs18@bmsce
Meet - AI LAB b3 batch
AI/ai_labtest_2 at main · samirkumar
Repl.it - RosyEnormousSpools
repl.it/@samirkumar111/RosyEnormousSpools#main.py
samirkumar111 / RosyEnormousSpools
Run
Upgrade
Share
Files
main.py
def negate(term):
return f'{term}' if term[0] != '' else term[1]
def reverse(clause):
if len(clause) > 2:
t = split_terms(clause)
return f'{t[1]}v{t[0]}'
return ''
def split_terms(rule):
exp = '(~*[ABCD])'
terms = re.findall(exp, rule)
return terms
def contradiction(goal, clause):
contradictions = [f'{goal}v{negate(goal)}', f'{negate(goal)}v{goal}']
return clause in contradictions or reverse(clause) in contradictions
def resolve(rules, goal):
temp = rules.copy()
temp += [negate(goal)]
steps = dict()
for rule in temp:
steps[rule] = 'Given.'
steps[negate(goal)] = 'Negated conclusion.'
i = 0
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.
15:11
01/01/2021


(no subject) - samir.cs18@bmsce
Meet - AI LAB b3 batch
AI/ai_labtest_2 at main · samirkumar
Repl.it - RosyEnormousSpools
repl.it/@samirkumar111/RosyEnormousSpools#main.py
samirkumar111 / RosyEnormousSpools
Run
Upgrade
Share
Files
main.py
clauses += [f'{gen[0]}v{gen[1]}']
else:
if contradiction(goal,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 in turn null. \
\nA contradiction is found when {negate(goal)} is assumed
as true. Hence, {goal} is true."
return steps
elif len(gen) == 1:
clauses += [f'{gen[0]}']
else:
if contradiction(goal,f'{terms1[0]}v{terms2[0]}'):
temp.append(f'{terms1[0]}v{terms2[0]}')
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."
return steps
for clause 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 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.
15:12
01/01/2021


(no subject) - samir.cs18@bmsce
Meet - AI LAB b3 batch
AI/ai_labtest_2 at main · samirkur
Repl.it - RosyEnormousSpools
repl.it/@samirkumar111/RosyEnormousSpools#main.py
samirkumar111 / RosyEnormousSpools
Run
Upgrade
Share
Files
main.py
def negate(term):
return f'
{term}' if
term[0] != ''
else term[1]
def reverse
(clause):
if len(clause)
> 2:
t =
split_term
s(clause)
return f'
{t[1]}v{t
[0]}'
return ''
def split_terms
(rule):
exp = '(~*
[ABCD])'
terms =
re.findall
(exp, rule)
return terms
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.
15:12
01/01/2021