



$\Rightarrow$  def contradiction(goal, clause):  
 contradicts =  $\{ \{ \text{goal} \} \vee \{ \text{negate}(\text{goal}) \} : \{ \{ \text{negate}(\text{goal}) \} \vee \{ \text{goal} \} \}$   
 return clause in contradicts or clause in contradicts

def resolve(rule, goal):  
 temp = rule.copy()  
 temp += [negate(goal)]  
 step = dict()  
 for rule in temp:  
 steps[rule] = 'Given.'  
 steps[negate(goal)] = 'negated conclusion.'

i = 0  
 while i < len(temp):  
 n = len(temp)  
 j = (i+1) % n  
 clauses = []

while j != i;

temp1 = split - temp[temp[i]]

temp2 = split - temp[temp[j]]

for c in temp1: if negate(c) in temp2:

t1 = [t for t in temp1 if t != c]

t2 = [t for t in temp2 if t != negate(c)]

gen = t1 + t2.

if len(gen) == 2:

if gen[0] != negate(gen[1]):



$\Rightarrow$  clause  $+= \{f' \{gen[0]\} \vee \{gen[1]\}'\}$

else:

if

contradiction goal,  $f' \{gen[0]\}$

$\vee \{gen[1]\}'$ :

temp.append( $f' \{gen[0]\} \vee \{gen[1]\}'$ ) steps[""] =

$f'$  Reduced  $\{temp[i]\} \in \{temp[j]\}$  to  $\{temp[-1]\}$ ,

return steps

elif len(gen) == 1:

clause +=

$[f' \{gen[0]\}']$

else:

if contradiction goal,  $f' \{kns[0]\}$

$\vee \{kns[2[0]'\}$ :

temp.append( $f' \{kns[0]\} \vee \{kns[2[0]'\}$ :

steps[""] =  $f'$  Reduced  $\{temp[i]\}$  and

$\{temp[j]\}$ , then

$j = (j+1) \times n$

$i += 1$

return steps

rules = ~~from B to A~~

$\Rightarrow$  rules = ' $A \vee \neg C \sim A \vee B \sim C \vee B$ '

$P \Rightarrow B : \neg A \vee B, B \Rightarrow C, \neg B \vee C$

goal = 'c'

main (rules, goal)