5-B

=> flogram 02%

import re

def main (rules, goal):
rules = rules. ext. + ( · · )

step = resolve (rula, goal)

Print C' In Step (+) Chaup (+1 Devisation (+')

Print ('-' + 30)

for sley in alegs:

Print ( f. 2:3:141 2 alep3/41 2 alep8[160] 3/4)

del negate (term): letur f'Eto { else form [i]}

of reverce (dans):

if lun (claul) > 2: t = 41.t - Herry (claul)

rotur j. { f [ i] } v { 1 [ o] } retur"

old aplitatems (rul):
exp: (~ [ABCD])

tens - 1e. findal (esp. vulo) lating lens

Aplit - kens ('NAVE')

satralitain (god, clad): def contradicts - Cf. Egod's v Enegate (god) 5. f' Enegate (god) 3 v Egod's) celu claus in contradicts or course (claus) in contradicts verde (rule, goal): temp = rules - copy () tup + = (negate (god)) sty = dict 0 for rule in lap: Rleps [rule] = 'Given.' steps [nog ate (god)) = 'vegated conclusion: who ; < h ( kg); n = le (fepp) j = (i+i) v.n Claude = () while j 1 = i; tens 1 = split - tons (leap (i)) Harry 2 = spl: 1 - lay (lamp [j]) for C in low 1: if negate (c) in low 2: t1 = [t for t in least if t!=c] dz = Ct fort in leasz if t! = negat (c))

gen = t + tz.

if la (gen) = = 2:

i b gen [o]! = ngate(gan[i]): => clave += [f' {gen [0]} v {gen [i]}'] else:

Cotract dan (goal, f' Egen [0])

Vigen [1]; ):

Coll V [gon [1]; ) remp apped (f' squisol} of gar[i]) steps[i] = & " Revoluted Themp [i] & { kemp [j] } to { hemp[-1] }, elif lan (gen) == 1: Claux += [ ] [gen[ 0] 2'] ele. Contradiction (goal of Elens 180) - v { less 2 (0] }'): lap apped (j' [las 160] } v[ks 2[0]]): step 6["] = f " Rubal { leng [i]} ad Etemp [j] }, A. j = (j+) y.N 3

P rule = 'AVN C ~ AVB ~ CUB' ~

P => B: ~ AVB, B=> C, ~ BVC

goal = 'c'

main (vula, god)

H