

29/12/20

Date

AI LAB TEST - II

Abhishek Kumar Soni
18M18CS004
5A



Program 4

(FOL to CNF)

import re

~~def remove_brackets~~

def getAttribute(string):

expr = '\([^\)]+\)'

matches = re.findall(expr, string)

return [m for m in matches if m.isalpha()]

def getPredicates(string):

expr = '[a-z~]+\([A-Za-z]+\)'

return re.findall(expr, string)

def DeMorgan(sentence):

string = ' '.join(list(sentence).copy())

string = string.replace('~', '')

flag = '[' in string

string = string.replace('~', '')

string = string.strip('[')

for predicate in getPredicates(string):

string = string.replace(predicate, '{predicate}')

s = list(string)

for i, c in enumerate(string):

if c == '1':

s[i] = '&'

elif c == '&':

s[i] = '1'

string = ' '.join(s)

string = string.replace('~', '')

return '{[string]}' if flag else string



def Skolemization(sentence):

SKOLEM_CONSTANTS = [j'chr(c)' for c in range(ord('A'), ord('Z') + 1)]

statement = ' '.join(list(sentence)) (copy())

matches = re.findall('[\u0394\u0395]', statement)

for match in matches [::-1]:

statement = statement.replace(match, '')

statements = re.findall('([\u0394\u0395]+)', statement)

for s in statements:

statement = statement.replace(s, s[1:-1])

for predicate in getPredicates(statement):

attributes = getAttributes(predicate)

if ' '.join(attribute.islower()):

statement = statement.replace(match[1], SKOLEM_CONSTANTS.pop(0))

else:

all = [a for a in attributes if not a.islower()][0]

statement = statement.replace(all, j'{' SKOLEM_CONSTANTS.pop(0) }'(match[1]))

return statement

def fol_to_cnf(fol):

statement = fol.replace("<=>", "_")

while '_' in statement:

i = statement.index('_')

new_statement = '[' + statement[:i] + '=>' + statement[i+1:] + ']' & '[' + statement[i+1:] + '=>' + statement[:i] + ']'

statement = new_statement

~~statement = statement.replace(match[1], SKOLEM_CONSTANTS.pop(0))~~



```

Statement = statement.replace("=>", "-")
expr = '\[([^\]]+)\]'
statements = re.findall(expr, statement)
for i, s in enumerate(statements):
    if '[' in s and ']' not in s:
        statements[i] += ']'
for s in statements:
    statement = statement.replace(s, join_ho_cg(s))
while '-' in statement:
    i = statement.index('-')
    br = statement.index '[' if '[' in statement else 0
    new_statement = '~' + statement[br:i] + '|' + statement[i+1:]
    statement = statement[:br] + new_statement if br > 0
    else new_statement
while '~V' in statement:
    i = statement.index('~V')
    statement = list(statement)
    statement[i], statement[i+1], statement[i+2] =
        ']', statement[i+2], '~'
    statement = ''.join(statement)
while '~]' in statement:
    i = statement.index('~]')
    s = list(statement)
    s[i], s[i+1], s[i+2] = 'V', s[i+2], '~'
    statement = ''.join(s)
statement = statement.replace('~[V', '[~V')
statement = statement.replace('~]', '[~]')
expr = '(\[([^\]]+)\])'
statements = re.findall(expr, statement)
for s in statements:
    statement = statement.replace(s, join_ho_cg(s))
expr = '\[([^\]]+)\]'

```

statements = re.findall(expr, statement)

for s in statements:

statement = statement.replace(s, DeMorgan(s))

return statement

statement = input("Enter formula statement:")

print(f"formula converted to cnf : {Skolemization(formula to cnf (statement))}")