

Name: Rudramuni G.m

USN: 13m18CS085

Bafna Gold

Sub: AI LAB-2

Date:

Page:

- (4) Convert the following English sentence into FOL.
Then convert FOL into CNF & use it as Input
English: All gardeners Like the sun

FOL: $\forall x \text{ Gardener}(x) \Rightarrow \text{Likes}(x, \text{sun})$

converting FOL into CNF code:

import re

def GetAttributes(String):

exp = '\([^\)]+\)',

matches = re.findall(exp, String)

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

def GetPredicates(String):

exp = '[a-zA-Z]+\([^\)]+\)',

return re.findall(exp, String)

def Demorgan(Sentence):

String = ' '.join(list(Sentence).copy())

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

flag = 'E' 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):

01

Rudra

Name: Rudramuni Gm

USN: 18M18CS085

```
if C == 'V':
    S[i] = 'v'
elif C == 'N':
    S[i] = 'V'

String = ".join(S)
String = String.replace('vv', 'v')
return F'[{String}]' if flag else String

def Skolemization(sentence):
    SKOLEM_CONSTANTS = [f' {u}_{i}' for i, u in enumerate
                        (ord('A'), ord('Z') + 1)]
    Statement = ".join(list(sentence).copy())
    matches = ".join(list(sentence).copy())
    matches = re.findall('[+?]', Statement)
    for match in matches[::-1]:
        Statement = Statement.replace(s, s[:-1])
    for Predicate in getPredicates(Statement):
        attributes = getAttributes(Predicate)
        if ".join(attributes).islower():
            Statement = Statement.replace(match[1],
                                           SKOLEM_CONSTANTS.pop(0))
    else:
        AL = [a for a in attributes if a.islower()]
        AU = [a for a in attributes if not a.islower()]
        Statement = Statement.replace(AU, f' {SKOLEM_CONSTANTS.pop(0)} ({AL[0] if len(AL) else
        match[1]})')
    return Statement

def tol-to-ent(tol):
    Statement = tol.replace("=", "-")
    while '-' in Statement:
```


Name : Rudransh Chibram

USN : IBM18C8088

Bafna Gold

Date: Page:

Lab Test - 02

```
i = Statement.index(' ')
new-Statement = '[' + Statement[i:] + ']'
Statement[i+1:] = Statement[i+1:] + '[' + Statement[i+1:]
Statement = new-Statement
enqs = '[' + Statement[i:] + ']'
statements = de.tindall(enqs, Statement)
for i, s in enumerate(statements):
    if '[' in s and ']' not in s:
        statements[i] += ']'
top is in statements:
statement = statement.replace(s, top + s + s)
```

while '[' in statement:

```
i = Statement.index(' ')
no = Statement.index('[')
Statement[i+1:] = Statement[i+1:] + '[' + Statement[i+1:]
Statement[i+1:] = Statement[i+1:] + ']'
```

$a \Rightarrow b \equiv a \Rightarrow b \wedge b \Rightarrow a$

statement = new-Statement

Statement = statement.replace(' > ', ' < ')

replace > using :- $a \Rightarrow b \equiv a \vee \neg b$

while ' < ' in Statement:

replace < with \neg

while ' < ' in Statement:

replace \neg with <

next Skolemization

next step drop <

next step apply distribution \vee over \wedge

03

Rudra

Lab-Test -02

Name: Rudramuni G.m
USN: 1Bm18CS085

Steps involved:

(1) Eliminate Implication

(2) move \neg inwards

(3) standardize variable

(4) Skolemization

(5) Drop \forall

(6) Distribute \vee over \wedge .

(7) Full