

Date : 29/12/2020

Akshay S Bhaskar

Sign : AKS

18N18CS011

AI Lab Test -2

Question number 4 : FOL to CNF

Writeup :

import re

def FOL to CNF(~~fol~~):

statement = fol.replace(" \Leftrightarrow ", "-")

while '-' in statement:

idx = statement.index('-')

st1 = '[' + statement[:idx] + '=>' + statement[idx+1:]
+ '=>' + statement[idx:] + 'I'

statement = st1

statement = statement.replace("=>", "-")

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

statements = re.findall(exp, statement)

for i, s in enumerate(statements):

if '[' in s and 'I' not in s:

statements[i] += 'I'

for s in statements:

statement = statement.replace(s, FOL to CNF(s))

while '-' in statement:

idx = statement.index('-')

b = statement.index('[') if '[' in statement else 0

st1 = '~' + statement[b:idx] + 'I' +
statement[idx+1:]

statement = statement[:b] + st1 if b > 0 else st1

[[[

Date: 29/12/2020

Akshay S Bhargadewaj

Sign: 

IBN18CS011

AI Lab Test - 2

while '~A' in statement:

```
idx = statement.index('~A')
statement = list(statement)
statement[idx], statement[idx+1], statement[idx+2]
= '(', '[', statement[idx+2], '~'
statement = ''.join(statement)
```

while '~]' in statement:

```
idx = statement.index('~]')
s = list(statement)
s[idx], s[idx+1], s[idx+2] = '(', '[', '~'
statement = ''.join(s)
statement = statement.replace('~[A~', '~A~')
statement = statement.replace('~[E~', '~E~')
exp = '~([A[E].)'
statements = re.findall(exp, statement)
for s in statements:
    statement = statement.replace(s, FOLtoCNF(s))
exp = '~\[[^]]+\]'
statements = re.findall(exp, statement)
for s in statements:
    statement = statement.replace(s, DeMorgan(s))
return statement,
```

def attributes(str1):

```
exp = '\([^\)]+\)'
match = re.findall(exp, str1)
return [m for m in str(match) if m.isalpha()]
```

Date: 29/12/2020

Akshay S Bharadwaj

Sign: 

IBH18CS011

A] Lab Test - 2

```
def predicates(st1):
```

```
    exp = '[a-z~]+\([A-Za-z,]+\)
```

```
    return re.findall(exp, st1).
```

```
def deMorgan(sentence):
```

```
    st1 = ''.join(list(sentence).copy())
```

```
    st1 = st1.replace('~', '')
```

```
    flag = '[' in st1
```

```
    st1 = st1.replace('~[', '')
```

```
    st1 = st1.strip(']')
```

```
    for p in predicates(st1):
```

```
        st1 = st1.replace(p, f'~{p}')
```

```
    s = list(st1)
```

```
    for i i, c in enumerate(st1):
```

```
        if c == '|':
```

```
            s[i] = '&'
```

```
        elif c == '&':
```

```
            s[i] = '|'
```

```
    st1 = st1 = st1 ''.join(s)
```

```
    st1 = st1.replace('~', '')
```

```
    return f'[{st1}]' if flag else st1
```

```
def skolemize(sent):
```

```
    constants = [f'c{chr(c)}' for c in range(ord('A'),  
                                                ord('Z') + 1)]
```

```
    statement = ''.join(list(sent).copy())
```

```
    matches = re.findall('[vz].', statement)
```

Date: 29/12/2020

Ashay S. Sharda

Sign: AS

IBMBCSO11

AI Lab Test - 2

```
for m in match [::-1]:
    statement = statement.replace(m, '')
    statements = re.findall('\[[^\]]+\]', statement)

    for st in statements:
        statement = statement.replace(st, st[1:-1])

    for p in predicates(statement):
        att = attributes(p)
        if ' ', join(att).islower():
            statement = statement.replace(m[1],
                                           constants.pop(0))
        else:
            aU = [a for a in att if not a.islower()]
            statement = statement.replace(aU,
                                           f'{{ constants.pop(0) }}({match[1]})')

    return statement
```