19/01/24.

Date __/__/__
Page

LAB-9

FOL TO CNF Conversion

```
def getAttributes (string):
    expr = '\([^)]+\)'
    matches = re.findall (expr, string)
    return [m for m in str(matches) if m.isalpha()]

def getPredicates (string):
    expr = '[a-z~]+\([A-Z a-z, ]+\)'
    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.strip(']')
    s = list(string)
    for i, c in enumerate (string):
        if c == '|':
            s[i] = '&'
        elif c == '&'
            s[i] = '|'
    string = ''.join(s)
    string = string.replace('~~', '')
    return f'[{string}]' if flag else string
```

```
def Skolemization(sentence):
    SKOLEM_CONSTANTS = [f'{chr(c)}' for c in
                        range(ord('A'), ord('Z')+1)]
    statement = ''.join(list(sentence).copy())
    matches = re.findall('[∀∃].', statement)
    for match in matches[::-1]:
        statement = statement.replace(predicate)
    s = list(string)
    for i, c in enumerate(string):
        statements = re.findall('\([^\(]+\)', statement)
    for s in statements:
        statement = statement.replace(s, s[1:-1])
    for predicate in getPredicates(statement):
        attributes = getAttributes(predicate)
        if ''.join(attributes).islower():
            statement = statement.replace(match[1],
                                          SKOLEM_CONSTANTS.pop())
        else:
            aL = [a for a in attributes if a.islower()]
            aU = [a for a in attributes if not
                  a.islower()][0]
            statement = statement.replace(aU, f'
                {SKOLEM_CONSTANTS.pop(0)}({aL[0] if
                len(aL) else match[1]})')
    return statement
```

OUTPUT:

[~animal(y) | loves(x,y)] & [~loves(x,y) | animal(y)]
[animal(G(x)) & ~loves(x,G(x))] | [loves(F(x),x)]
[~american(x) | ~weapon(y) | ~sells(x,y,z) |
~hostile(z)] | criminal(x)]

Algorithm:

1. Create a List of Skolem_constants
2. Find ∀, ∃
   If the attributes are lowercase, replace them with a skolem constant.
   Removed used skolem constant or function from the list.
   If the attributes are both lowercase and uppercase replace the uppercase attribute with a skolem function.
3. Replace ⟺ with '__'
   transform — as $Q \equiv (P \Rightarrow Q) \wedge (Q \Rightarrow P)$
4. Replace ⟹ with '_'
5. Apply demorgan's law
   replace ~[
   as ~P & ~Q if (| was present)
   replace ~[
   as ~p | ~Q if (& was present)
   replace ~~ with ' '

Example:

$\forall x$ king(x) ∧ Greedy(x) ⇒ Evil(x)

King(Richard) ∧ Greedy(Richard) ⇒ Evil(Richard)

~[King(Richard) ∧ Greedy(Richard)] ∨ Evil(Richard)

~King(Richard) ∨ ~Greedy(Richard) ∨ Evil(Richard)

Harshitha R-1BM21CS075

[~animal(y)|loves(x,y)]&[~loves(x,y)|animal(y)]

[animal(G(x))&~loves(x,G(x))]|[loves(F(x),x)]

[~american(x)|~weapon(y)|~sells(x,y,z)|~hostile(z)]|criminal(x)