1 12/1/2024

FOL to CNF conversion

Step1: remove_implications

return "Or (Not ({part[0]}), {part[1]})"

Step2: apply-demorgans-law

formula.replace ('Not (And', 'Or(Not)').replace('Not(Or', 'And (Not')

Step3: distribute_quantifiers

formula.replace ('Forall (', 'And (').replace ('Exits (', 'Or(')

Step1: Create a list of SKOLEM_CONSTANTS

Step2: Find ∀, ∃

If the attributes are lower case, replace them with a skolem constant

remove used skolem constant or function from the list

If the attributes are both lowercase and uppercase replace the uppercase attribute with a Skolem function

Step3: replace <=> with '-'

transform - as Q≡ (P=>Q) ^ (Q=>P)

Step4: replace => with '-'

Step5: Apply demorgan's law

replace ~[

as ~P&~Q if (| was present)

replace ~[

as ~P|~Q if (& was present)

replace ~~ with ''

FOL to CNF

```
print( Skolemization( fol_to_cnf ("animal (y) <=> loves (x,y)")))
print ( Skolemization( fol_to_cnf ("∀x [∀y [animal(y) => loves (x,y)]]
                    => [∃z [loves (z,x)]]")))
print ( fol_to_cnf ("[american (x) & weapon (y) & sells (x,y,z) &
                    hostile(z)] => criminal (x)"))
```

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)

```
print(Skolemization(fol_to_cnf("animal(y)<=>loves(x,y)")))
print(Skolemization(fol_to_cnf("∀x[∀y[animal(y)=>loves(x,y)]]=>[∃z[loves(z,x)]]")))
print(fol_to_cnf("[american(x)&weapon(y)&sells(x,y,z)&hostile(z)]=>criminal(x)"))
```

```
[~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)
```