27/10/25

Lab - 9

First Order Logic

Create a knowledge base consisting of FOL statements and prove the given query using Resolution

Proof by Resolution

Given KB or Premises

-> John likes all kind of food
-> Apple and vegetables are food
-> Anything anyone eats and not killed is food
-> Anil eats peanuts and still alive.
-> Harry eats everything that Anil eats
-> Anyone who is alive implies not killed
-> Anyone who is not killed implies alive

Prove by Resolution that:
John likes peanut

Representation in FOL

a. $\forall x : food(x) \rightarrow likes(John, x)$
b. $food(Apple) \wedge food(vegetables)$
c. $\forall x \forall y : eats(x, y) \wedge \neg killed(x) \rightarrow food(y)$
d. $eats(Anil, Peanuts) \wedge alive(Anil)$
e. $\forall x : eats(Anil, x) \rightarrow eats(Harry, x)$
f. $\forall x : \neg killed(x) \rightarrow alive(x)$
g. $\forall x : alive(x) \rightarrow \neg killed(x)$
h. $likes(John, Peanuts)$

lements and

. Eliminate implication

a. $\forall x \neg food(x) \vee likes(John, x)$
b. $food(Apple) \wedge food(vegetables)$
c. $\forall x \forall y \neg [eats(x,y) \wedge \neg killed(x)] \vee food(y)$
d. $eats(Anil, Peanuts) \wedge alive(Anil)$
e. $\forall x \neg eats(Anil, x) \vee eats(Harry, x)$
f. $\forall x \neg [\neg killed(x)] \vee alive(x)$
g. $\forall x \neg alive(x) \vee \neg killed(x)$
h. $likes(John, Peanuts)$

. Move negation ($\neg$) inwards and rewrite

a. $\forall x \neg food(x) \vee likes(John, x)$
b. $food(Apple) \wedge food(vegetables)$
c. $\forall x \forall y \neg eats(x,y) \vee killed(x) \vee food(y)$
d. $eats(Anil, Peanuts) \wedge alive(Anil)$
e. $\forall x \neg eats(Anil, x) \vee eats(Harry, x)$
f. $\forall x killed(x)] \vee alive(x)$
g. $\forall x \neg alive(x) \vee \neg killed(x)$
h. $likes(John, Peanuts)$

. Rename variable or standardize variables

a. $\forall x \neg food(x) \vee likes(John, x)$
b. $food(Apple) \wedge food(vegetables)$
c. $\forall y \forall z \neg eats(y, z) \vee killed(y) \vee food(z)$
d. $eats(Anil, Peanuts) \wedge alive(Anil)$
e. $\forall w \neg eats(Anil, w) \vee eats(Harry, w)$
f. $\forall g killed(g)] \vee alive(g)$
g. $\forall k \neg alive(k) \vee \neg killed(k)$
h. $likes(John, Peanuts)$

• Drop universe

a. $\neg food(x) \vee likes(John, x)$

b. $food(Apple)$

c. $food(vegetables)$

d. $\neg eats(y, z) \vee killed(y) \vee food(z)$

e. $eats(Anil, Peanuts)$

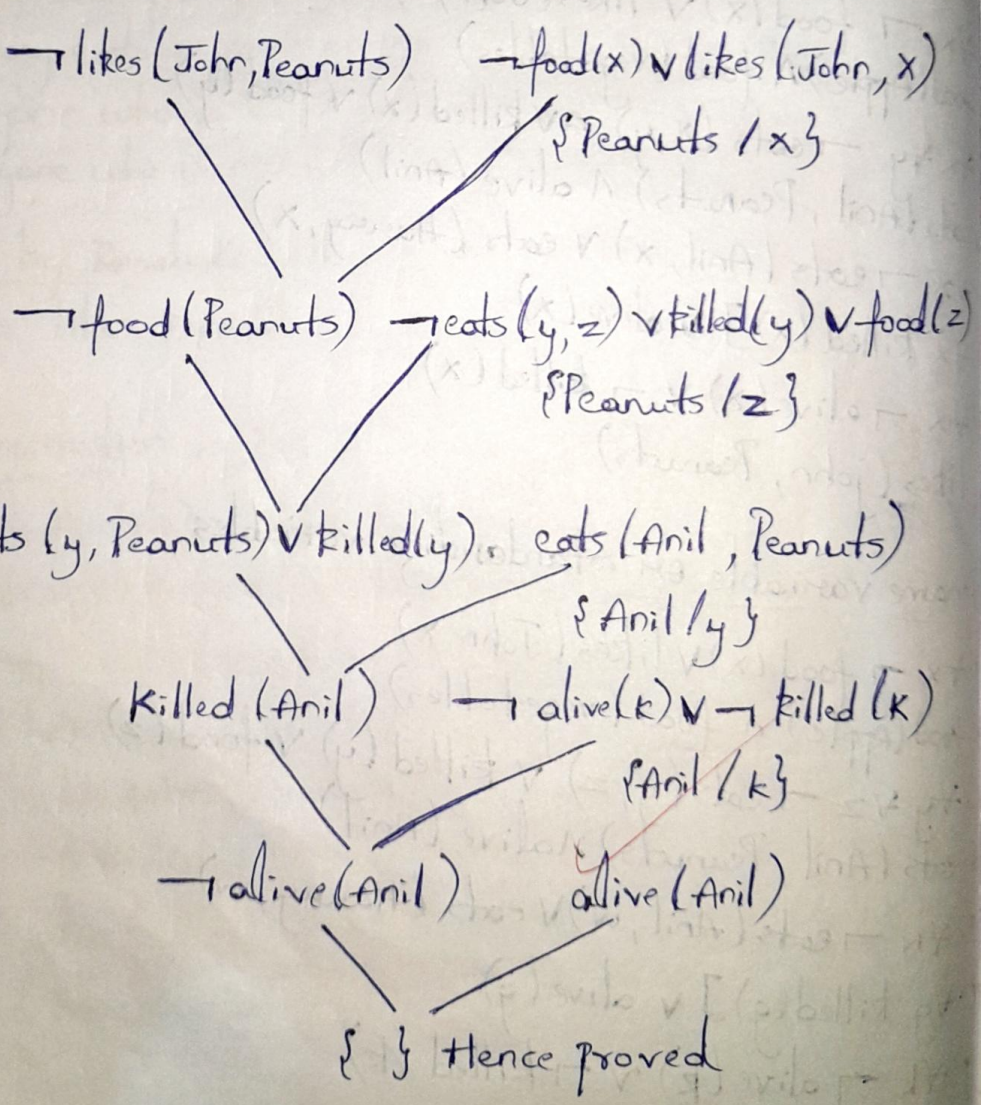
f. $alive(Anil)$

g. $\neg eats(Anil, w) \vee eats(Harry, w)$

h. $killed(g) \vee alive(g)$

i. $\neg alive(k) \vee \neg killed(k)$

j. $likes(John, Peanuts)$



Algorithm

1. Input:
   - Knowledge Base (KB)
   - Query (Q)
2. Convert KB and $\neg Q$ to clausal Form:
   - Eliminate implications
   - Move negations inward
   - standardize variables
   - skolemize (remove $\exists$ quantifiers)
   - Drop universal quantifiers
   - Convert to CNF.
3. Apply Resolution:
   - Repeatedly resolve pairs of clauses that contain complementary literals
   - Add new clauses to the KB
   - Stop if:
     - Empty clause ($\perp$) is derived $\rightarrow Q$ is true
     - No new clauses can be added $\rightarrow Q$ is False
4. output True/False for Query Q