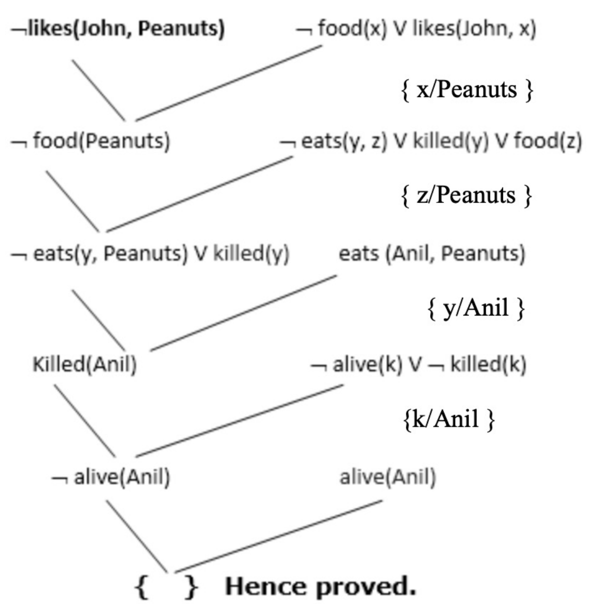
CS 4100: Introduction to AI

Wayne Snyder
Northeastern University

Lecture 5: Resolution Theorem Proving in First-Order Logic



First-Order Logic: Semantics

INTERPRETATION: $\mathcal{U}$

P, Q, f, g, c

$\mathbb{I} \models F$

From last time.....

Definition 3.4

- An atomic formula $p(t_1, \ldots, t_n)$ is *true* (or valid) under the interpretation $\mathbb{I}$ if, after interpretation and evaluation of all terms $t_1, \ldots, t_n$ and interpretation of the predicate p through the n-place relation r, it holds that

$$(\mathbb{I}(t_1), \ldots, \mathbb{I}(t_n)) \in r.$$

- The truth of quantifierless formulas follows from the truth of atomic formulas—as in propositional calculus—through the semantics of the logical operators defined in Table 2.1 on page 25.
- A formula $\forall x\ F$ is true under the interpretation $\mathbb{I}$ exactly when it is true given an arbitrary change of the interpretation for the variable x (and only for x)
- A formula $\exists x\ F$ is true under the interpretation $\mathbb{I}$ exactly when there is an interpretation for x which makes the formula true.

The definitions of semantic equivalence of formulas, for the concepts satisfiable, true, unsatisfiable, and model, along with semantic entailment (Definitions 2.4, 2.5, 2.6) carry over unchanged from propositional calculus to predicate logic.

$f(g(x), a)$

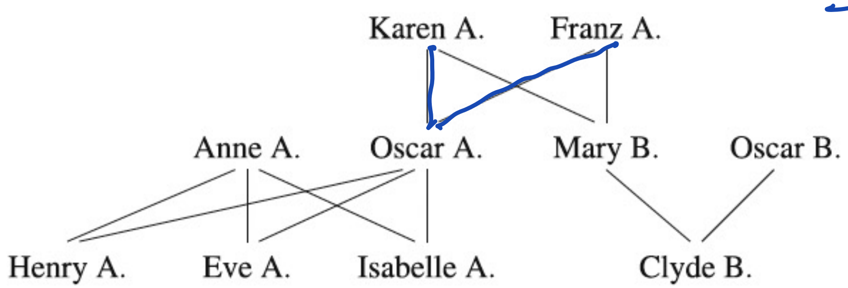
$\forall x\ F$

$x \in \mathcal{U}$

$\exists x\ F$

$x \in \mathcal{U}$

First-Order Logic: Semantics



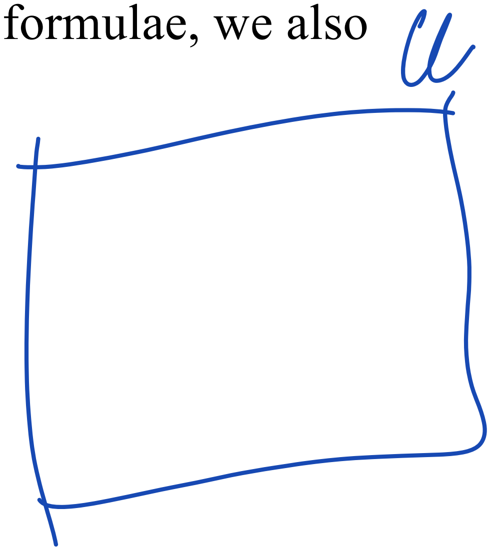
KNOWLEDGE GRAPH

$$
\begin{aligned}
KB \equiv\ & female(karen) \wedge female(anne) \wedge female(mary) \\
& \wedge female(eve) \wedge female(isabelle) \\
& \wedge child(oscar, karen, franz) \wedge child(mary, karen, franz) \\
& \wedge child(eve, anne, oscar) \wedge child(henry, anne, oscar) \\
& \wedge child(isabelle, anne, oscar) \wedge child(clyde, mary, oscarb) \\
& \wedge (\forall x\, \forall y\, \forall z\, child(x, y, z) \Rightarrow child(x, z, y)) \\
& \wedge (\forall x\, \forall y\, descendant(x, y) \Leftrightarrow \exists z\, child(x, y, z) \\
& \vee (\exists u\, \exists v\, child(x, u, v) \wedge descendant(u, y))).
\end{aligned}
$$

First-Order Logic: Equivalent Formulae

From last time: Equivalence of formulae in FOL:

In addition to all the rules for equivalence of propositional formulae, we also have the following rules involving quantifiers:



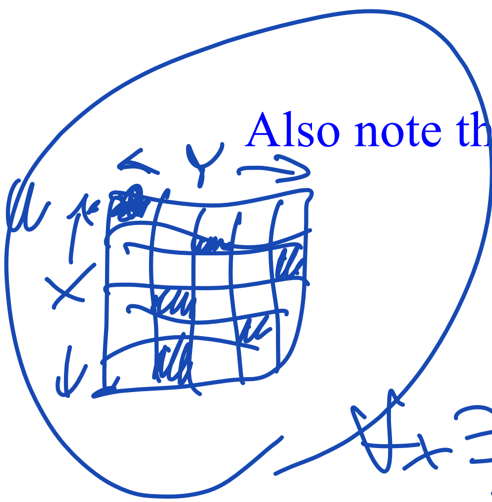
$$\forall x\ F \quad \equiv \quad \neg\boxed{\exists x\ \neg F}$$

$$\exists x\ F \quad \equiv \quad \neg\boxed{\forall x\ \neg F}$$

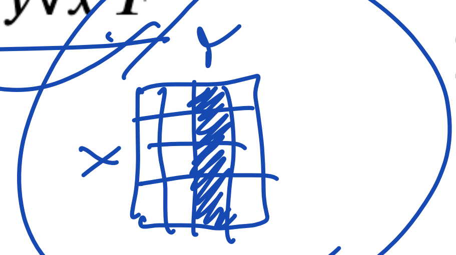
$$\forall x \forall y\ F \quad \equiv \quad \forall y \forall x\ F$$

$$\exists x \exists y\ F \quad \equiv \quad \exists y \exists x\ F$$

Also note that you can not exchange the order of different quantifiers:



$$\forall x \exists y\ F \quad \not\equiv \quad \exists y \forall x\ F$$



$$\exists x\ F[\ldots x \ldots x \ldots] \quad \equiv$$

$\forall x \exists y.\ y < x$

$\mathbb{N}$ $\{0, 1, 2, \ldots$

First-Order Logic: Semantics

$\exists y \forall x.\ y \leq x$

First-Order Logic: Semantics (DIGRESSION)

Equality is a special case of a relation which is always given the natural interpretation, using the axioms:

$$
\begin{array}{lll}
\forall x & x = x & \text{(reflexivity)} \\
\forall x\ \forall y & x = y \Rightarrow y = x & \text{(symmetry)} \\
\forall x\ \forall y\ \forall z & x = y \wedge y = z \Rightarrow x = z & \text{(transitivity).}
\end{array}
\quad (3.1)
$$

$P(x\ldots) = \text{TRUE}$

$=(x, y)$

$$\forall x\ \forall y\ x = y \Rightarrow p(x) \Leftrightarrow p(y) \quad \text{(substitution axiom).}$$

$$\forall x\ \forall y\ x = y \Rightarrow f(x) = f(y) \quad \text{(substitution axiom)}$$

$x = Y$ ~~then~~, $P(\ldots X \ldots) \Rightarrow P(\ldots Y \ldots)$

(END DIGRESSION)

First-Order Logic: Semantics

First-Order Logic: Equivalent Formulae

From last time: Equivalence of formulae in FOL:

In addition to all the rules for equivalence of propositional formulae, we also have the following rules involving quantifiers:

$$\forall x\, F \equiv \neg \exists x\, \neg F$$

$$\exists x\, F \equiv \neg \forall x\, \neg F$$

$$\forall x \forall y\, F \equiv \forall y \forall x\, F$$

$$\cancel{\exists x \exists y\, F \equiv \exists y \exists x\, F}$$

$\Rightarrow$

$\Leftrightarrow$

DRIVE $\neg$ INSIDE

GET RID OF $\exists$

DRIVE $\vee$ BELOW $\wedge$

CNF FOR FOL

Also note that you can not exchange the order of different quantifiers:

$$\forall x \exists y\, F \not\equiv \exists y \forall x\, F$$

First-Order Logic: Equivalent Formulae

As we shall see, Resolution theorem proving in FOL requires that the formula does not contain existential quantifiers. Fortunately, a technique called Skolemization can be used to remove existentials without changing the satisfiability of the formula.

Skolemization involves substituting a new constant for the existentially quantified variable in the formula:

$\mathcal{U}$

WITNESS FOR $\exists$

$$\exists x\, F[\ldots x \ldots x \ldots] \quad \equiv \quad F[\ldots c \ldots c \ldots]F$$

$\bullet c$

The reason we can do this without changing the fundamental character of the problem (i.e., whether it is satisfiable or unsatisfiable) is:

$$\mathbb{I} \models \exists x\, F[\ldots x \ldots x \ldots]$$

$\exists x \forall y.\ x \leq y$

if and only if

$\emptyset$ $\quad$ $\mathcal{U} = \mathbb{N}$

$$\mathbb{J} \models F[\ldots c \ldots c \ldots]$$

where $\mathbb{J}$ is the same as $\mathbb{I}$ except that it provides an interpretation for the new constant. This is always possible because of the definition of $\exists$.

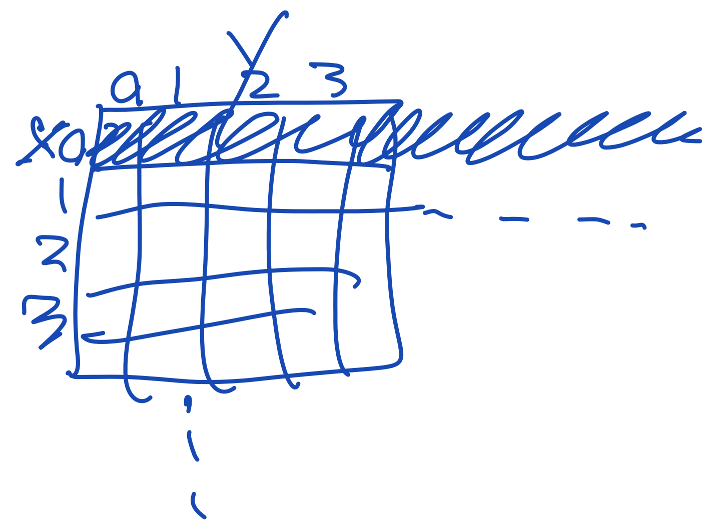
First-Order Logic: Equivalent Formulae

$\exists F \ldots$

Example:

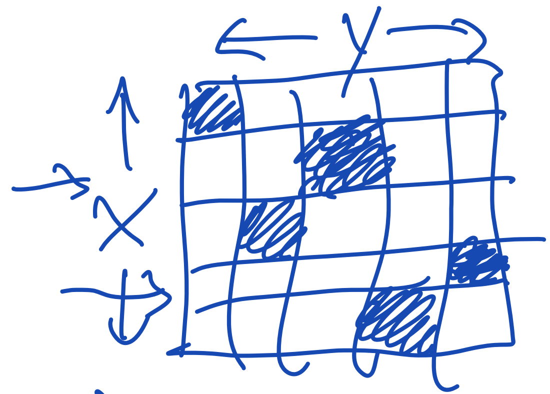
$\mathbb{N} \quad \{a, 1, 2, \ldots$

$\exists x \forall y.\ x \leq y$



First-Order Logic: Equivalent Formulae

Skolemization can also be done when an existentially-quantified variable occurs inside the scope of a universal quantifier:



$$\mathbb{I} \models \forall x \exists y\, F[\ldots y \ldots y \ldots]$$

if and only if

$$\mathbb{J} \models \forall x\, F[\ldots f(\cancel{y}\,x) \ldots f(\cancel{y}\,x) \ldots]$$

$f(x) = y$

where $\mathbb{J}$ is the same as $\mathbb{I}$ except that it provides an interpretation for the new function symbol f . This function "chooses" an value for y for each of the values of x. Since the formula says that you can always do this, the function f can always be given an interpretation in $\mathbb{J}$.

$\forall x \exists y.\ y < x$

$\mathbb{Z}$

First-Order Logic: Equivalent Formulae

Example:

First-Order Logic: Equivalent Formulae

To summarize: Any set of formulae S can be transformed into a set S' where there are no existential quantifiers, where S is satisfiable if and only if S' is satisfiable.

Why is this important ?

Resolution theorem proving can be performed in FOL for sets of universally-quantified formulae (no ∃'s) in conjunctive normal form:

$x_1 \dots x_n$

$$\Big(\forall x_1, \dots, x_k (L_1 \vee L_2 \vee \cdots \vee L_{m_1}) \wedge \forall \dots (L_1 \vee L_2 \vee \cdots \vee L_{m_2}) \wedge \cdots \wedge \forall \dots (L_1 \vee L_2 \vee \cdots \vee L_{m_n}) \Big)$$

where each L_i is an atomic formula (predicate or negation of a predicate). As usual, we represent a CNF using sets of literals, where now all variables are assumed universally quantified:

$P(t_1 \dots t_n$ $\neg P(t_1 \dots t_n$

$$\Big\{ \overset{\forall}{\{L_1, L_2, \cdots, L_{m_1}\}}, \dots \overset{\forall}{\{L_1, L_2, , L_{m_2}\}}, \cdots, \overset{\forall}{\{L_1, L_2, \cdots, L_{m_n}\}} \Big\}$$

Full disclosure: When using resolution in FOL, we hardly ever use Skolemization, instead writing our set of formulae without using ∃. Whew !

First-Order Logic: Resolution

The ONLY thing that changes in resolution when we go to FOL is that we have to deal with a new form of the resolution rule. Briefly, we have to account for when precisely there is a conflict between two literals of opposite sign.

Some examples will clarify the problem.

DEF RESOLVE(C_1, C_2)

⋮

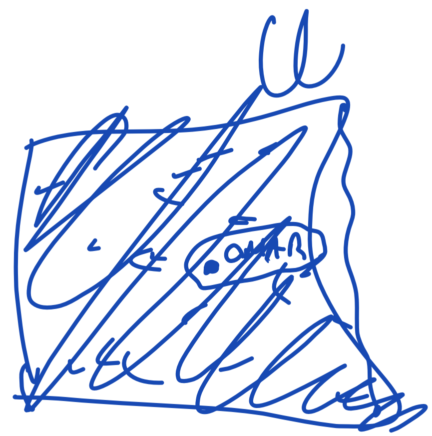
Example 1

Everyone loves logic.
Prove that omar loves logic.

$\forall x.\ \text{LovesLogic}(x)$
LovesLogic(omar)

$\forall x(\text{LovesLogic}(x)) \land \neg\ \text{LovesLogic}(\text{omar})$

$A \quad \neg A$



$\{\ \ \{\text{LovesLogic}(x)\ \},\ \ \{\ \neg\ \text{LovesLogic}(\text{omar})\ \}\ \ \}$

Why is this unsatisfiable?

$\{\}$

$x \mid$ omar

LovesLogic(omar)

$$\frac{\forall x.\ LL(x)}{LL(c)}$$

INSTANTIATION
FOR ANY CONSTANT C

SUBSTITUTION: $x \mapsto$ OMAR

x/OMAR

First-Order Logic: Resolution

Example 2

Everyone in CS4100 loves logic.	$\forall x.\ \mathrm{InCS4100}(x) \rightarrow \mathrm{LovesLogic}(x)$
Omar is in CS4100	InCS4100(omar)
Prove that omar loves logic.	LovesLogic(omar)

$$\forall x(\neg\mathrm{InCS4100}(x)\ \cancel{\rightarrow}\ \vee\ \mathrm{LovesLogic}(x)) \wedge \mathrm{InCS4100}(\mathrm{omar}) \wedge \neg \mathrm{LovesLogic}(\mathrm{omar})$$

$$\{\ \{\neg\mathrm{InCS4100}(x), \mathrm{LovesLogic}(x)\},\ \{\mathrm{InCS4100}(\mathrm{omar})\},\ \{\neg\mathrm{LovesLogic}(\mathrm{omar})\}\ \}$$

σ

{INCS4100(OMAR), LOVESLOGIC(OMAR)}

UNIFIER

$\sigma = \{x/\text{OMAR}\}$

{LOVESLOGIC(OMAR)}

{}

✓

First-Order Logic: Resolution

Example 3

All of Omar's friends love logic. $\forall x.\ \text{Friend}(\text{omar}, \text{x}) \rightarrow \text{LovesLogic}(x)$

Ali is friends with everyone. $\forall y.\ \text{Friend}(\text{y}, \text{ali})$

Prove that ~~omar~~ ALI loves logic. $\text{LovesLogic}(\cancel{\text{omar}})$ ALI

$\forall x(\text{Friend}(\text{omar}, \text{x}) \rightarrow \text{LovesLogic}(x)) \land \text{Friend}(\text{y}, \text{ali}) \land \neg \text{LovesLogic}(\cancel{\text{omar}})$ ALI

$\{ \forall \{\neg \boxed{\text{Friend}(\text{omar}, \text{x})}, \text{LovesLogic}(x)\}, \forall \{\boxed{\text{Friend}(\text{y}, \text{ali})}\}, \{\neg \text{LovesLogic}(\cancel{\text{omar}})\} \}$ ALI

$\sigma =$

$\sigma = \{X/\text{ALI}, Y/\text{OMAR}\}$

FRIEND(OMAR, ALI) = FRIEND(OMAR, ALI)

{¬FRIEND(OMAR, ALI) ∪ LOVESLOGIC(ALI)}

{FRIEND(OMAR, ALI)}

{LOVESLOGIC(ALI)}

FRIEND(OMAR, X)

FRIEND(Y, ALI)

X/ALEX, Y/WAYNE

First-Order Logic: Resolution

$\forall x\, F \equiv \forall y\, F[x/y]$

Example 3b (with clash of variable names)

RENAME BOUND

All of Omar's friends love logic.	$\forall x1.\ \mathrm{Friend}(\mathrm{omar}, x1) \to \mathrm{LovesLogic}(x1)$
Ali is friends with everyone.	$\forall x2\ \mathrm{Friend}(x2, \mathrm{ali})$
Prove that ~~omar~~ ALI loves logic.	$\mathrm{LovesLogic}(\cancel{\mathrm{omar}}\ \mathrm{ALI})$

$\forall x(\mathrm{Friend}(\mathrm{omar}, x1) \to \mathrm{LovesLogic}(x1))\ \wedge\ \mathrm{Friend}(x2, \mathrm{ali}) \wedge \neg\, \mathrm{LovesLogic}(\cancel{\mathrm{omar}}\ \mathrm{ALI})$

$\{\ \{\neg\, \mathrm{Friend}(\mathrm{omar}, x1),\ \mathrm{LovesLogic}(x1)\ \},\ \{\ \mathrm{Friend}(x2, \mathrm{ali})\ \},\ \{\ \neg\, \mathrm{LovesLogic}(\cancel{\mathrm{omar}}\ \mathrm{ALI})\ \}\ \}$

FRIEND(OMAR, X1)

FRIEND(X2, ALI)

$\forall$. FRIEND(OMAR, _) → LL(_)

X1/ALI, X3/OMAR

X1 X2 X3

BEFORE RESOLVE 2 CLAUSES

RENAME ALL VARS.

First-Order Logic: Resolution

Example 3c (with variables renamed)

All of Omar's friends love logic.	$\forall x1.\ \mathrm{Friend}(\mathrm{omar}, \mathrm{x1}) \rightarrow \mathrm{LovesLogic}(x1)$
Ali is friends with everyone.	$\forall x2.\ \mathrm{Friend}(\mathrm{x2}, \mathrm{ali})$
Prove that omar loves logic.	LovesLogic(omar)

$\forall x(\mathrm{Friend}(\mathrm{omar}, x1) \rightarrow \mathrm{LovesLogic}(\mathrm{x1}))\ \wedge\ \mathrm{Friend}(\mathrm{x2}, \mathrm{ali}) \wedge \neg\ \mathrm{LovesLogic}(\mathrm{omar})$

$\{\ \{\neg\ \mathrm{Friend}(\mathrm{omar}, x1),\ \mathrm{LovesLogic}(x1)\ \}, \{\ \mathrm{Friend}(\mathrm{x2}, \mathrm{ali})\ \}, \{\ \neg\ \mathrm{LovesLogic}(\mathrm{omar})\ \}\ \}$

First-Order Logic: Resolution

Example 4 (with unification)

m(x) = mother of x (everyone has a mother)

P. 52

Ind

TEXT

Everyone knows their mother. $\forall x.\ \text{Knows}(x, m(x))$

Prove that henry knows someone. $\exists x.\ \text{Knows}(henry, x)$ Q

KB ⊨ Q

{KB ∪ {¬Q}} IS UNSAT.

$\neg Q = \neg\exists x.\ \text{Knows}(henry, x)$

$\forall x.\ \neg\text{Knows}(henry, x)$

$$\forall x(\text{Knows}(x, m(x)) \wedge \neg\text{Knows}(henry, x)$$

$$\{\ \{\ \text{Knows}(x_1, m(x_1))\ \},\ \{\neg\text{Knows}(henry, x_2)\ \}\ \}$$

{ }

Knows(x_1, m(x_1))

Knows(HENRY, x_2)

{ x_1/HENRY

Knows(HENRY, M(HENRY))

Knows(HENRY, x_2)

Unfortunately, unification can get much more complicated.....

$x_2 / M(\text{HENRY}) \}$

First-Order Logic: Unification

VARS: x, y, z, u

Unify the two terms: P(f(g(x)), y, z) P(u, u, f(u))

$u / f(g(x))$

$y / f(g(x))$

$z / f(f(g(x)))$

$P(f(g(x)), y, z)$ $P(f(g(x)), f(g(x)), f(f(g(x))))$

$P(f(g(x)), f(g(x)), \cancel{z})$ $f(f(g(x)))$ $P(f(g(x)), f(g(x)), f(f(g(x))))$

MGU MOST GENERAL UNIFIER

WHEN DOES UNIFICATION FAIL?

OCCURS CHECK

$f(x)$ $g(a_2)$ CLASH OF

x f(x)
a f(a)

SYMBOLS

First-Order Logic: Resolution Rule for FOL

Definition 3.7 Two literals are called *unifiable* if there is a substitution σ for all variables which makes the literals equal. Such a σ is called a *unifier*. A unifier is called the most general unifier (MGU) if all other unifiers can be obtained from it by substitution of variables.

Definition 3.8 The resolution rule for two clauses in conjunctive normal form reads

$$\frac{(A_1 \vee \cdots \vee A_m \vee B), \quad (\neg B' \vee C_1 \vee \cdots \vee C_n) \quad \sigma(B) = \sigma(B')}{(\sigma(A_1) \vee \cdots \vee \sigma(A_m) \vee \sigma(C_1) \vee \cdots \vee \sigma(C_n))}, \tag{3.6}$$

where σ is the MGU of B and B'.

$\sigma(B$

$$\frac{\sigma(B) = \sigma(B')}{\sigma(B) = \sigma(\neg B')}$$

First-Order Logic: Unification

First-Order Logic: Equivalent Formulae

Example 5:

a. **John likes all kind of food.**

b. **Apple and vegetable are food**

c. **Anything anyone eats and not killed is food.**

d. **Anil eats peanuts and still alive**

e. **Harry eats everything that Anil eats.**
   Prove by resolution that:

f. **John likes peanuts.**

First-Order Logic: Semantics

Step-1: Conversion of Facts into FOL

In the first step we will convert all the given statements into its first order logic.

a. **John likes all kind of food.**

b. **Apple and vegetable are food**

c. **Anything anyone eats and not killed is food.**

d. **Anil eats peanuts and still alive**

e. **Harry eats everything that Anil eats.**

   Prove by resolution that:

f. **John likes peanuts.**

a. $\forall x$: food(x) → likes(John, x)

b. food(Apple) Λ food(vegetables)

c. $\forall x \forall y$: eats(x, y) Λ ¬ killed(x) → food(y)

d. eats (Anil, Peanuts) Λ alive(Anil).

e. $\forall x$: eats(Anil, x) → eats(Harry, x)

f. $\forall x$: ¬ killed(x) → alive(x) } **added predicates.**

g. $\forall x$: alive(x) →¬ killed(x)

h. likes(John, Peanuts)

First-Order Logic: Semantics

Step-2: Conversion of FOL into CNF

a. $\forall x$: food(x) → likes(John, x)

b. food(Apple) Λ food(vegetables)

c. $\forall x$ $\forall y$: eats(x, y) Λ ¬ killed(x) → food(y)

d. eats (Anil, Peanuts) Λ alive(Anil).

e. $\forall x$: eats(Anil, x) → eats(Harry, x)

f. $\forall x$: ¬ killed(x) → alive(x) } **added predicates.**

g. $\forall x$: alive(x) →¬ killed(x)

h. likes(John, Peanuts)

- **Eliminate all implication (→) and rewrite**

  a. ∀x ¬ food(x) V likes(John, x)

  b. food(Apple) Λ food(vegetables)

  c. ∀x ∀y ¬ [eats(x, y) Λ ¬ killed(x)] V food(y)

  d. eats (Anil, Peanuts) Λ alive(Anil)

  e. ∀x ¬ eats(Anil, x) V eats(Harry, x)

  f. ∀x¬ [¬ killed(x)] V alive(x)

  g. ∀x ¬ alive(x) V ¬ killed(x)

  h. likes(John, Peanuts).

First-Order Logic: Semantics

a. $\forall x$: food(x) → likes(John, x)

b. food(Apple) Λ food(vegetables)

c. $\forall x$ $\forall y$: eats(x, y) Λ ¬ killed(x) → food(y)

d. eats (Anil, Peanuts) Λ alive(Anil).

e. $\forall x$: eats(Anil, x) → eats(Harry, x)

f. $\forall x$: ¬ killed(x) → alive(x)

g. $\forall x$: alive(x) →¬ killed(x)

added predicates.

h. likes(John, Peanuts)

- **Move negation (¬)inwards and rewrite**

a. ∀x ¬ food(x) V likes(John, x)

b. food(Apple) Λ food(vegetables)

c. ∀x ∀y ¬ eats(x, y) V killed(x) V food(y)

d. eats (Anil, Peanuts) Λ alive(Anil)

e. ∀x ¬ eats(Anil, x) V eats(Harry, x)

f. ∀x ¬killed(x)] V alive(x)

g. ∀x ¬ alive(x) V ¬ killed(x)

h. likes(John, Peanuts).

First-Order Logic: Semantics

a. $\forall x$: food(x) → likes(John, x)

b. food(Apple) ∧ food(vegetables)

c. $\forall x \forall y$: eats(x, y) ∧ ¬ killed(x) → food(y)

d. eats (Anil, Peanuts) ∧ alive(Anil).

e. $\forall x$: eats(Anil, x) → eats(Harry, x)

f. $\forall x$: ¬ killed(x) → alive(x)

g. $\forall x$: alive(x) →¬ killed(x)

added predicates.

h. likes(John, Peanuts)

- **Rename variables or standardize variables**

a. ∀x ¬ food(x) V likes(John, x)

b. food(Apple) Λ food(vegetables)

c. ∀y ∀z ¬ eats(y, z) V killed(y) V food(z)

d. eats (Anil, Peanuts) Λ alive(Anil)

e. ∀w¬ eats(Anil, w) V eats(Harry, w)

f. ∀g ¬killed(g)] V alive(g)

g. ∀k ¬ alive(k) V ¬ killed(k)

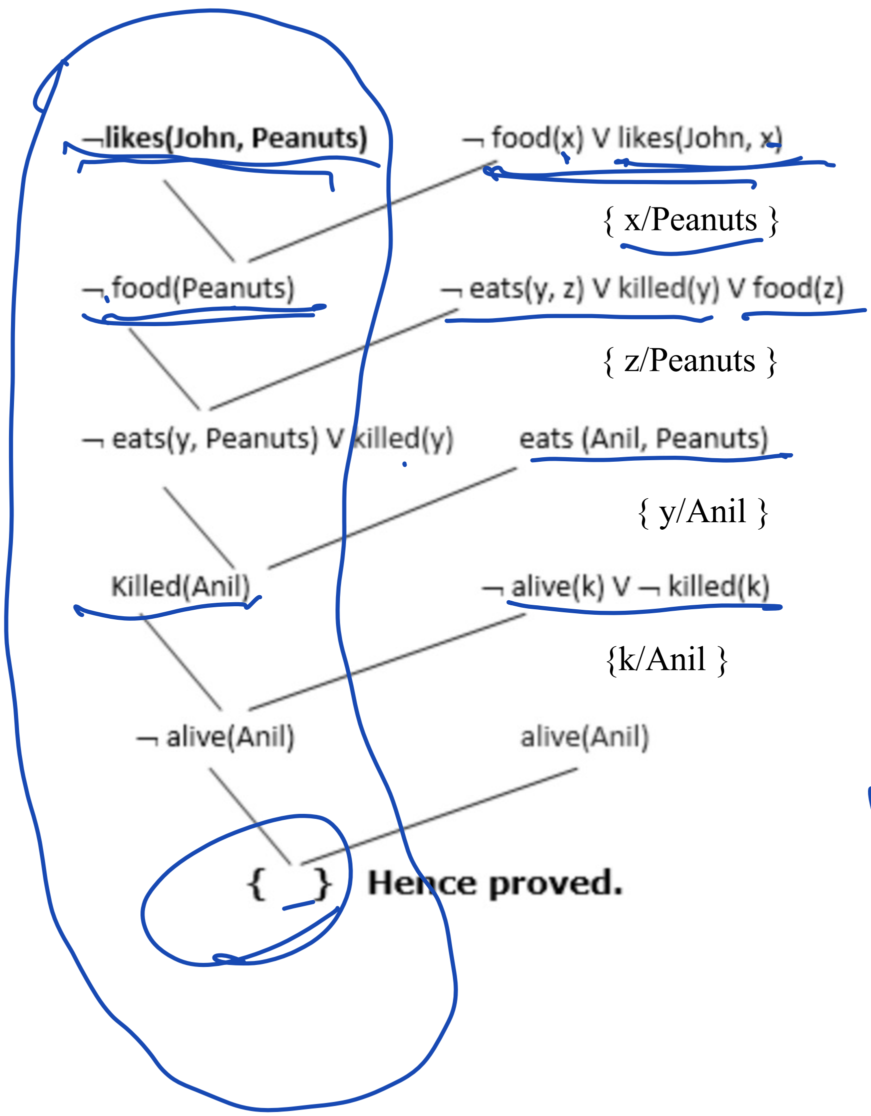
h. likes(John, Peanuts).

First-Order Logic: Semantics

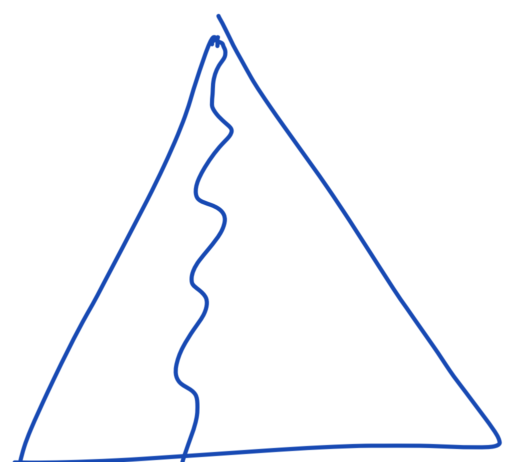
- **Drop Universal quantifiers.**
  In this step we will drop all universal quantifier since all the statements are not implicitly quantified so we don't need it.

  a. ¬ food(x) V likes(John, x)
  b. food(Apple)
  c. food(vegetables)
  d. ¬ eats(y, z) V killed(y) V food(z)
  e. eats (Anil, Peanuts)
  f. alive(Anil)
  g. ¬ eats(Anil, w) V eats(Harry, w)
  h. killed(g) V alive(g)
  i. ¬ alive(k) V ¬ killed(k)
  j. likes(John, Peanuts).

First-Order Logic: Semantics



LINEAR RESOLUTION



DFS