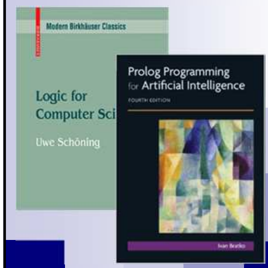


FLORIDA INTERNATIONAL UNIVERSITY

First Order Logic (IV)

Dr. Antonio L. Bajuelos

FIU School of Computing & Information Sciences

Resolution in FOL. Preliminaries

Some **preliminaries definitions**:

- If a formula G results from certain **substitutions** from a formula F, then G is called an **instance** of F.
- **Substitutions** which make a formula **variable free** are called **ground substitutions.**
- The result of applying a **ground substitution** to a formula is a **ground instance** of that formula.

Resolution. Preliminaries (cont…)

- Remember that:

Let F be a set of clauses. Then $Res(F)$ is defined as

$$Res(F) = F \cup \{R \mid R \text{ is a resolvent of two clauses in } F\}.$$

Furthermore, define

$$\begin{aligned} Res^0(F) &= F \\ Res^{n+1}(F) &= Res(Res^n(F)) \quad \text{for } n \geq 0 \,. \end{aligned}$$

and finally, let

$$Res^*(F) = \bigcup_{n \geq 0} Res^n(F) \,.$$

Resolution. Ground Resolution Procedure

- **Ground Resolution Procedure**

Instance: a closed formula F in Skolem form with its matrix F^* in **CNF**

$i := 0$;
$M := \emptyset$;
repeat
    $i := i + 1$;
    $M := M \cup \{F_i\}$;
    $M := Res^*(M)$;
until $\square \in M$;
Output "unsatisfiable" and halt;

Resolution. The first example

- **Example:** Consider the following **unsatisfiable** formula

$$F = \forall x(P(x) \land \neg P(f(x))).$$

Here we have,

$$F^* = (P(x) \land \neg P(f(x))),$$

which is written in clause form,

$$F^* = \{\{P(x)\}, \{\neg P(f(x))\}\}.$$

Furthermore,

$$\{(P(a) \land \neg P(f(a))), (P(f(a)) \land \neg P(f(f(a)))), \ldots\}$$

- Already the first two ground substitutions **[x/a]** and **[x/f(a)]** lead to a **unsatisfiable** clause set. Why?

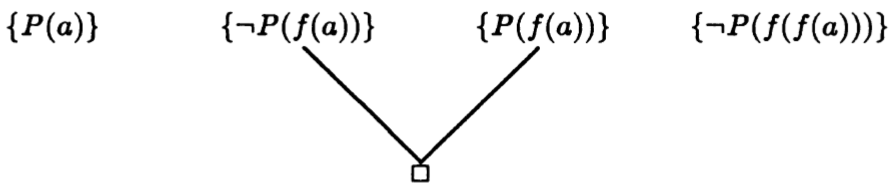
5

Resolution. The first example (cont...)

- **Example:** Consider the following **unsatisfiable** formula

$$F = \forall x(P(x) \land \neg P(f(x))).$$

$$\{(P(a) \land \neg P(f(a))), (P(f(a)) \land \neg P(f(f(a)))), \ldots\}$$



- In this example, already two clauses are generated which are not needed for the **resolution refutation**.
- Therefore, we conclude that it suffices to consider **ground substitutions** that are applied **individually** to the clauses of the original formula F*.

6

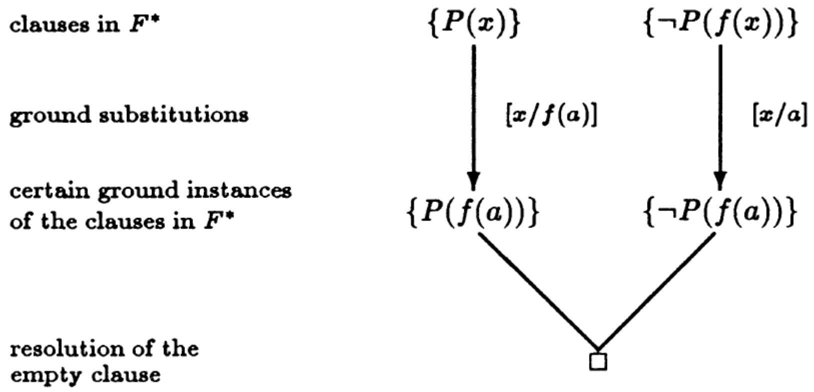
Resolution. The first example (cont…)

- **Example:** Consider the following **unsatisfiable** formula

$$F = \forall x(P(x) \wedge \neg P(f(x))).$$

$$\{(P(a) \wedge \neg P(f(a))), (P(f(a)) \wedge \neg P(f(f(a)))), \dots\}$$

- Note that it suffices to consider **ground substitutions** that are applied **individually** to the clauses of the original formula F*.



Resolution. The second example

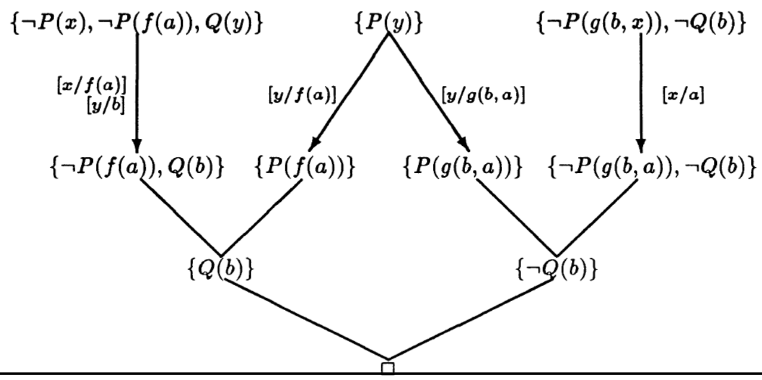
- **Example:** Let us consider a more complex example. Let

$$F = \forall x \forall y((\neg P(x) \vee \neg P(f(a)) \vee Q(y)) \wedge P(y) \wedge (\neg P(g(b,x)) \vee \neg Q(b))).$$

- Then we obtain the following clause representation of F^*,

$$F^* = \{\{\neg P(x), \neg P(f(a)), Q(y)\}, \{P(y)\}, \{\neg P(g(b,x)), \neg Q(b)\}\}.$$

- This formula F is **unsatisfiable**. A proof for the **unsatisfiability** of F is given by the following diagram.



8

Resolution. The FOL version

- Now we introduce the **FOL** version of **resolution** which was invented by J. A. **Robinson**.
- The **new idea**:
  - to **resolve clauses** in **FOL** to clauses in **FOL**, where each resolution step is accompanied by a **substitution**.
- A **substitution** is a [v/t], where v is a variable and t is a term, different from v.
- A set of substitutions is a finite set $\{[v_1/t_1], \ldots, [v_n/t_n]\}$, where $[v_i/t_i]\}$, $1 \leq i \leq n$ is a substitution and no two elements in the set have the same variable before the symbol /.

9

Resolution. The FOL version

- For example, in the case of

$$\{\{P(x), \neg Q(g(x))\}, \{\neg P(f(y))\}\},$$

  - it suffices to use the **substitution** [x/f(y)] to obtain the **resolvent** $\{\neg Q(g(f(y))\}$.
  - There is no need at this point to substitute anything for the variable y.

10

Resolution. Unifier

- **Definition (unifier, most general unifier)**
  - A **substitution sub** is a **unifier** for a (finite) set of literals

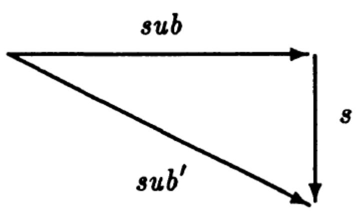
    $$L = \{L_1, L_2, \ldots, L_k\},$$

    $$\textbf{if } L_1\textbf{sub} = L_2\textbf{sub} = \ldots = L_k\textbf{sub}.$$

  - That is, by applying **sub** to every literal in the set **L**, one and only one literal is obtained.
  - If **Lsub** expresses the set obtained by applying **sub** to every literal in the set **L**, then this situation can be formally expressed by **|Lsub|=1**
  - If a substitution **sub** exists with the property that **|Lsub|=1**, then we say **L** is **unifiable**.

Resolution. Unifier (cont…)

- A unifier **sub** for some literal set **L** is called a **most general unifier** if
  - for every unifier **sub'** there is a substitution s such that **sub'= sub** o **s** . ("**sub** o **s" is a composition of sub and s)**
- Here, the equality **sub'= sub** o **s** means that for every formula F, **Fsub' = Fsub** o **s**.
- The following diagram describes the situation:



Unification Theorem

Theorem (Robinson):

- Every unifiable set of literals has a **most general unifier (MGU)**.

- **Definition (Disagreement Set):**
  - Let S be a finite set of simple expressions.
  - Locate the leftmost symbol position at which not all expressions in S have the same symbol and
  - Extract from each expression in S the subexpression beginning at that symbol position.
  - The set of all such subexpressions is the **disagreement set**.
- **Example:**

  Let S = {P(f(x),h(y),a), P(f(x),z,a), P(f(x),h(y),b)},

  then the disagreement set, D, is {h(y),z}

13

Unification Algorithm (pseudocode)

Input: A non-empty set of literals L.
sub := []; (this is the empty substitution)
Lsub := L;
while |Lsub| > 1 do {
    Scan the literals in Lsub from left to right, until the disagreement set **D** is not an empty set. Let L1 and L2 the corresponding symbols which are different
    if none of these symbols is a variable then
        output "non-unifiable" and halt
    else {
        Let x be the variable, and let t be a term that in D;
        if x occurs in t then
            output "non-unifiable" and halt;
        else
            sub := sub o [x/t];
            (this means the composition of the substitutions sub and [x/t])
        }
}
output sub as a **most general unifier** of L ;

{P(f(x),h(y),a), P(f(x),z,a), P(f(x),h(y),b)}, D = {z, h(y)}
{P(f(x),h(y),a), P(f(x),z,a), P(f(x),h(y),b)}, D = {a, b}

14

Unification Algorithm. Examples

- Find a most general unifier (**MGU**) for the following clause sets:

  (a) S = {P(a,y), P(x,f(b))}

  Sol: MGU = {[x/a], [y/f(b)]}

  (b) S = {P(a,x,f(g(y))), P(z,f(z),f(u))}

  Sol: MGU = {[z/a], [x/f(a)], [u/g(y)]}

  (c) S = {Q(f(a),g(x)), Q(y,y)}

  Sol: S is NOT unificable

15

Unification Algorithm. Examples

- We want to apply the unification algorithm to the set of literals:

$$\mathbf{L} = \{\neg P(f(z, g(a, y)), h(z)), \neg P(f(f(u, v), w), h(f(a, b)))\}$$

- Then we obtain in the first step:

$$\neg P(f(z, g(a, y)), h(z))$$
$$\neg P(f(f(u, v), w), h(f(a, b)))$$
$$\uparrow$$

  which results in the substitution **sub** - **[z/f(u,v)]**.
- In the second step, after applying **sub**, we obtain:

$$\neg P(f(f(u, v), g(a, y)), h(f(u, v)))$$
$$\neg P(f(f(u, v), w), h(f(a, b)))$$
$$\uparrow$$

- Therefore, the substitution is extended by **[w/g(a,y)]**.

$$\neg P(f(f(u, v), g(a, y)), h(f(u, v)))$$
$$\neg P(f(f(u, v), g(a, y)), h(f(a, b)))$$
$$\uparrow$$

16

Unification Algorithm. Examples

- We want to apply the unification algorithm to the set of literals:

$$\mathbf{L} = \{\neg P(f(z, g(a, y)), h(z)), \neg P(f(f(u, v), w), h(f(a, b)))\}$$

- ...

$$\neg P(f(f(u, v), g(a, y)), h(f(u, v)))$$
$$\neg P(f(f(u, v), g(a, y)), h(f(a, b)))$$
$$\uparrow$$

- Now sub is extended by **[u/a].** In the fourth step

$$\neg P(f(f(a, v), g(a, y)), h(f(a, v)))$$
$$\neg P(f(f(a, v), g(a, y)), h(f(a, b)))$$
$$\uparrow$$

- Now sub is extended by **[v/b].** In the next step we obtain the final substitution **sub** - **[z/f(u,v)][w/g(a,y)][u/a][v/b].**

- This is a **MGU ={[z/f(u,v)],[w/g(a,y)],[u/a],[v/b]}** for L, and we have:

$$\mathbf{L}sub = \{\neg P(f(f(a, b)), g(a, y)), h(f(a, b)))\}$$

17

Unification Algorithm. Examples

- **Example:** Determine the **MGU**

$$S = \{P(x, f(y), z), P(g(a), f(w), u), P(v, f(b), c)\}$$

- 1. $S_0 = S$, $D(S_0) = \{x, g(a), v\}$. Possible substitution are $[x/g(a)]$,$[x/v]$,$[v/g(a)]$, and $[v/x]$. $\Phi_1 = [x/g(a)]$

$$S_1 = S_0\phi_1 = \{P(g(a), f(y), z), P(g(a), f(w), u), P(v, f(b), c)\}$$

2. $D(S_1) = \{g(a), v\}$, $\phi_2 = [v/g(a)]$.

$$S_2 = S_1\phi_2 = \{P(g(a), f(y), z), P(g(a), f(w), u), P(g(a), f(b), c)\}$$

3. $D(S_2) = \{y, w, b\}$, $\phi_3 = [y/w]$.

$$S_3 = S_2\phi_2 = \{P(g(a), f(w), z), P(g(a), f(w), u), P(g(a), f(b), c)\}$$

4. $D(S_3) = \{w, b\}$, $\phi_3 = [w/b]$.

$$S_4 = S_3\phi_3 = \{P(g(a), f(b), z), P(g(a), f(b), u), P(g(a), f(b), c)\}$$

5. $D(S_4) = \{z, u, c\}$, $\phi_3 = [z/u]$.

$$S_5 = S_4\phi_4 = \{P(g(a), f(b), u), P(g(a), f(b), u), P(g(a), f(b), c)\}$$

6. $D(S_5) = \{u, c\}$, $\phi_3 = [u/c]$.

$$S_6 = S_5\phi_5 = \{P(g(a), f(b), c)\}$$

- $\boxed{\textbf{MGU} = [x/g(a)][v/g(a)][y/w][w/b][z/u][u/c]}$

18

Resolution. Example

- **Consider the following facts:**
  - Jack owns a dog.
    $\exists x : Dog(x) \wedge Owns(Jack, x)$
  - Every dog owner is an animal lover.
    $\forall x;\ (\exists y\ \ Dog(y) \wedge Owns(x, y)) \rightarrow AnimalLover(x)$
  - No animal lover kills an animal.
    $\forall x;\ \ AnimalLover(x) \rightarrow (\forall y\ \ Animal(y) \rightarrow \neg Kills(x, y))$
  - Either Jack or Curiosity killed the cat, who is named Tuna.
    $Kills(Jack, Tuna) \vee Kills(Curiosity, Tuna)$

Did Curiosity kill the cat?

Additional facts:

$Cat(Tuna)$

$\forall x :\ \ Cat(x) \rightarrow Animal(x)$

19

Resolution. Example (cont…)

- **Consider the following facts:**
  - Jack owns a dog.
  - Every dog owner is an animal lover.
  - No animal lover kills an animal.
  - Either Jack or Curiosity killed the cat, who is named Tuna.
  - Did Curiosity kill the cat?
- Let *D* a placeholder for the dogs unknown name. Then we have

$Dog(D)$

$Owns(Jack, D)$

20

Resolution. Example (cont…)

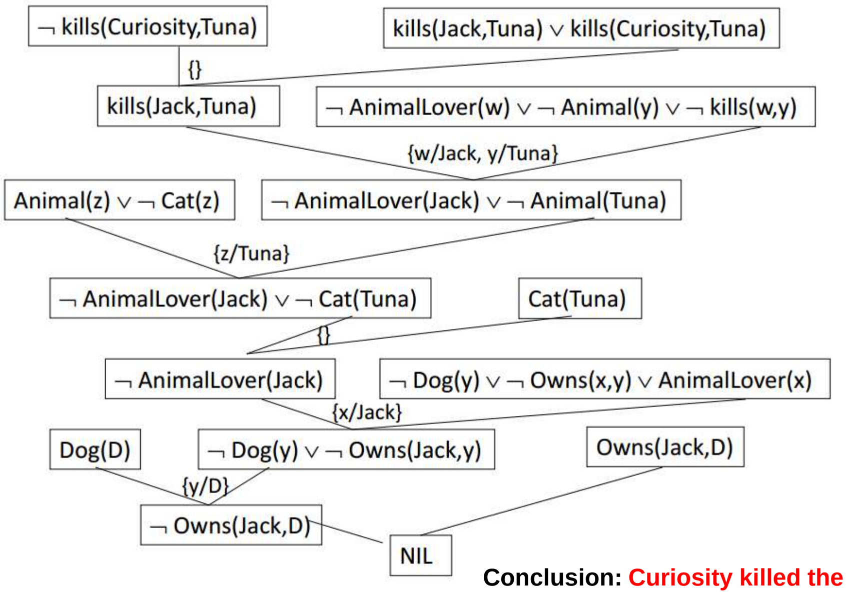
- **The set of clauses in CNF is:**

$$Dog(D)$$
$$Owns(Jack, D)$$
$$\neg Dog(y) \vee \neg Owns(x, y) \vee AnimalLover(x)$$
$$\neg AnimalLover(w) \vee \neg Animal(y) \vee \neg Kills(w, y)$$
$$Kills(Jack, Tuna) \vee Kills(Curiosity, Tuna)$$
$$Cat(Tuna)$$
$$\neg Cat(z) \vee Animal(z)$$
$$\neg Kills(Curiosity, Tuna)$$

21

Resolution. Example (cont…)

- **The resolution process:**



Conclusion: **Curiosity killed the cat!**