

What is Unification?

- Let $\Psi_1 = \text{King}(x)$, $\Psi_2 = \text{King}(\text{John})$, $\rightarrow x \rightarrow \text{John}$

E.g. Let's say there are two different expressions, $P(x, y)$, and $P(a, f(z))$.

$$\begin{array}{ll} P(x, y) \dots\dots\dots (i) \\ \text{blue } P(a, f(z)) \dots\dots\dots (ii) \end{array}$$

- Substitute x with a , and y with $f(z)$ in the first expression, and it will be represented as a/x and $f(z)/y$.
- With both the substitutions, the first expression will be identical to the second expression and the substitution set will be: $[a/x, f(z)/y]$.

Conditions for Unification:

Following are some basic conditions for unification:

- Predicate symbol must be same, atoms or expression with different predicate symbol can never be unified.
- Number of Arguments in both expressions must be identical.
- Unification will fail if there are two similar variables present in the same expression.

Unification Algorithm:

Algorithm: Unify(Ψ_1, Ψ_2)

Step. 1: If Ψ_1 or Ψ_2 is a variable or constant, then:

- a) If Ψ_1 or Ψ_2 are identical, then return NIL.
- b) Else if Ψ_1 is a variable,
 - a. then if Ψ_1 occurs in Ψ_2 , then return FAILURE
 - b. Else return { (Ψ_2 / Ψ_1) }.
- c) Else if Ψ_2 is a variable,
 - a. If Ψ_2 occurs in Ψ_1 then return FAILURE,
 - b. Else return { (Ψ_1 / Ψ_2) }.
- d) Else return FAILURE.

Step.2: If the initial Predicate symbol in Ψ_1 and Ψ_2 are not same, then return FAILURE.

Step. 3: IF Ψ_1 and Ψ_2 have a different number of arguments, then return FAILURE.

Step. 4: Set Substitution set (SUBST) to NIL.

Step. 5: For $i=1$ to the number of elements in Ψ_1 .

- a) Call Unify function with the i th element of Ψ_1 and i th element of Ψ_2 , and put the result into S.
- b) If S = failure then returns Failure
- c) If S \neq NIL then do,
 - a. Apply S to the remainder of both L1 and L2.
 - b. SUBST= APPEND(S, SUBST).

Step.6: Return SUBST.

Implementation of the Algorithm

Step.1: Initialize the substitution set to be empty.

Step.2: Recursively unify atomic sentences:

- a. Check for Identical expression match.
- b. If one expression is a variable v_i , and the other is a term t_i which does not contain variable v_i , then:
 - a. Substitute t_i / v_i in the existing substitutions
 - b. Add t_i / v_i to the substitution setlist.
 - c. If both the expressions are functions, then function name must be similar, and the number of arguments must be the same in both the expression.

For each pair of the following atomic sentences find the most general unifier (If exist).

1. Find the MGU of $\{p(f(a), g(Y)) \text{ and } p(X, X)\}$

Sol: $S_0 \Rightarrow$ Here, $\Psi_1 = p(f(a), g(Y))$, and $\Psi_2 = p(X, X)$
 SUBST $\theta = \{f(a) / X\}$
 $S_1 \Rightarrow \Psi_1 = p(f(a), g(Y))$, and $\Psi_2 = p(f(a), f(a))$
 SUBST $\theta = \{f(a) / g(Y)\}$, **Unification failed.**

Unification is not possible for these expressions.

2. Find the MGU of $\{p(b, X, f(g(Z))) \text{ and } p(Z, f(Y), f(Y))\}$

Here, $\Psi_1 = p(b, X, f(g(Z)))$, and $\Psi_2 = p(Z, f(Y), f(Y))$
 $S_0 \Rightarrow \{p(b, X, f(g(Z))); p(Z, f(Y), f(Y))\}$
 SUBST $\theta = \{b/Z\}$

$S_1 \Rightarrow \{p(b, X, f(g(b))); p(b, f(Y), f(Y))\}$
 SUBST $\theta = \{f(Y)/X\}$

$S_2 \Rightarrow \{p(b, f(Y), f(g(b))); p(b, f(Y), f(Y))\}$
 SUBST $\theta = \{g(b)/Y\}$

$S_1 \Rightarrow \{p(b, X, f(g(b))); p(b, f(Y), f(Y))\}$
 SUBST $\theta = \{f(Y)/X\}$

$S_2 \Rightarrow \{p(b, f(Y), f(g(b))); p(b, f(Y), f(Y))\}$
 SUBST $\theta = \{g(b)/Y\}$

$S_2 \Rightarrow \{ p(b, f(g(b))), f(g(b)); p(b, f(g(b)), f(g(b))) \}$ **Unified Successfully.**
And Unifier = $\{ b/Z, f(Y) / X, g(b) / Y \}$.

3. Find the MGU of $\{p(X, X), \text{ and } p(Z, f(Z))\}$

Here, $\Psi_1 = \{p(X, X)\}$, and $\Psi_2 = \{p(Z, f(Z))\}$

$S_0 \Rightarrow \{p(X, X), p(Z, f(Z))\}$

SUBST $\theta = \{X/Z\}$

$S_1 \Rightarrow \{p(Z, Z), p(Z, f(Z))\}$

SUBST $\theta = \{f(Z) / Z\}$, **Unification Failed.**

Hence, unification is not possible for these expressions.

4. Find the MGU of UNIFY(prime(11), prime(y))

Here, $\Psi_1 = \{\text{prime}(11)\}$, and $\Psi_2 = \{\text{prime}(y)\}$

$S_0 \Rightarrow \{\text{prime}(11), \text{prime}(y)\}$

SUBST $\theta = \{11/y\}$

$S_1 \Rightarrow \{\text{prime}(11), \text{prime}(11)\}$, **Successfully unified.**

Unifier: $\{11/y\}$.

5. Find the MGU of $Q(a, g(x, a), f(y)), Q(a, g(f(b), a), x)$

Here, $\Psi_1 = \{Q(a, g(x, a), f(y))\}$, and $\Psi_2 = \{Q(a, g(f(b), a), x)\}$

$S_0 \Rightarrow \{Q(a, g(x, a), f(y)); Q(a, g(f(b), a), x)\}$

SUBST $\theta = \{f(b)/x\}$

$S_1 \Rightarrow \{Q(a, g(f(b), a), f(y)); Q(a, g(f(b), a), f(b))\}$

SUBST $\theta = \{b/y\}$

$S_1 \Rightarrow \{Q(a, g(f(b), a), f(b)); Q(a, g(f(b), a), f(b))\}$, **Successfully Unified.**

Unifier: $[a/a, f(b)/x, b/y]$.

6. UNIFY(knows(Richard, x), knows(Richard, John))

Here, $\Psi_1 = \{\text{knows(Richard, x)}\}$, and $\Psi_2 = \{\text{knows(Richard, John)}\}$

$S_0 \Rightarrow \{\text{knows(Richard, x); knows(Richard, John)}\}$

SUBST $\theta = \{\text{John}/x\}$

$S_1 \Rightarrow \{\text{knows(Richard, John); knows(Richard, John)}\}$, **Successfully Unified.**

Unifier: $\{\text{John}/x\}$

Resolution in FOL

Resolution

Resolution is a theorem proving technique that proceeds by building refutation proofs, i.e., proofs by contradictions. It was invented by a Mathematician John Alan Robinson in the year 1965.

Resolution is used, if there are various statements are given, and we need to prove a conclusion of those statements. Unification is a key concept in proofs by resolutions. Resolution is a single inference rule which can efficiently operate on the **conjunctive normal form or clausal form**.

Clause: Disjunction of literals (an atomic sentence) is called a **clause**. It is also known as a unit clause.

Conjunctive Normal Form: A sentence represented as a conjunction of clauses is said to be **conjunctive normal form** or **CNF**

The resolution inference rule:

The resolution rule for first-order logic is simply a lifted version of the propositional rule. Resolution can resolve two clauses if they contain complementary literals, which are assumed to be standardized apart so that they share no variables.

$$\frac{l_1 \vee \dots \vee l_k \quad m_1 \vee \dots \vee m_n}{\text{SUBST}(\theta, l_1 \vee \dots \vee l_{i-1} \vee l_{i+1} \vee \dots \vee l_k \vee m_1 \vee \dots \vee m_{j-1} \vee m_{j+1} \vee \dots \vee m_n)}$$

Where l_i and m_j are complementary literals.

This rule is also called the **binary resolution rule** because it only resolves exactly two literals.

Example:

We can resolve two clauses which are given below:

[Animal (g(x) \vee Loves (f(x), x)] and [\neg Loves(a, b) \vee \neg Kills(a, b)]

Where two complimentary literals are: **Loves (f(x), x) and \neg Loves (a, b)**

These literals can be unified with unifier **$\theta = [a/f(x), \text{ and } b/x]$** , and it will generate a resolvent clause:

[Animal (g(x) \vee \neg Kills(f(x), x)].

Steps for Resolution:

1. Conversion of facts into first-order logic.
2. Convert FOL statements into CNF
3. Negate the statement which needs to prove (proof by contradiction)
4. Draw resolution graph (unification).

To better understand all the above steps, we will take an example in which we will apply resolution.

Example:

- 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."

Step-1: Conversion of Facts into FOL

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

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

Step-2: Conversion of FOL into CNF

In First order logic resolution, it is required to convert the FOL into CNF as CNF form makes easier for resolution proofs.

○ Eliminate all implication (\rightarrow) and rewrite

1. $\forall x \neg \text{food}(x) \vee \text{likes}(\text{John}, x)$
2. $\text{food}(\text{Apple}) \wedge \text{food}(\text{vegetables})$
3. $\forall x \forall y \neg [\text{eats}(x, y) \wedge \neg \text{killed}(x)] \vee \text{food}(y)$
4. $\text{eats}(\text{Anil}, \text{Peanuts}) \wedge \text{alive}(\text{Anil})$
5. $\forall x \neg \text{eats}(\text{Anil}, x) \vee \text{eats}(\text{Harry}, x)$
6. $\forall x \neg [\neg \text{killed}(x)] \vee \text{alive}(x)$
7. $\forall x \neg \text{alive}(x) \vee \neg \text{killed}(x)$
8. $\text{likes}(\text{John}, \text{Peanuts})$

○ Move negation (\neg) inwards and rewrite

1. $\forall x \neg \text{food}(x) \vee \text{likes}(\text{John}, x)$

2. $\text{food}(\text{Apple}) \wedge \text{food}(\text{vegetables})$
3. $\forall x \forall y \neg \text{eats}(x, y) \vee \text{killed}(x) \vee \text{food}(y)$
4. $\text{eats}(\text{Anil}, \text{Peanuts}) \wedge \text{alive}(\text{Anil})$
5. $\forall x \neg \text{eats}(\text{Anil}, x) \vee \text{eats}(\text{Harry}, x)$
6. $\forall x \neg \text{killed}(x) \vee \text{alive}(x)$
7. $\forall x \neg \text{alive}(x) \vee \neg \text{killed}(x)$
8. $\text{likes}(\text{John}, \text{Peanuts})$.

○ **Rename variables or standardize variables**

1. $\forall x \neg \text{food}(x) \vee \text{likes}(\text{John}, x)$
2. $\text{food}(\text{Apple}) \wedge \text{food}(\text{vegetables})$
3. $\forall y \forall z \neg \text{eats}(y, z) \vee \text{killed}(y) \vee \text{food}(z)$
4. $\text{eats}(\text{Anil}, \text{Peanuts}) \wedge \text{alive}(\text{Anil})$
5. $\forall w \neg \text{eats}(\text{Anil}, w) \vee \text{eats}(\text{Harry}, w)$
6. $\forall g \neg \text{killed}(g) \vee \text{alive}(g)$
7. $\forall k \neg \text{alive}(k) \vee \neg \text{killed}(k)$
8. $\text{likes}(\text{John}, \text{Peanuts})$.

○ **Eliminate existential instantiation quantifier by elimination.**

In this step, we will eliminate existential quantifier \exists , and this process is known as **Skolemization**. But in this example problem since there is no existential quantifier so all the statements will remain same in this step.

- **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.

1. $\neg \text{food}(x) \vee \text{likes}(\text{John}, x)$
2. $\text{food}(\text{Apple})$
3. $\text{food}(\text{vegetables})$
4. $\neg \text{eats}(y, z) \vee \text{killed}(y) \vee \text{food}(z)$
5. $\text{eats}(\text{Anil}, \text{Peanuts})$
6. $\text{alive}(\text{Anil})$
7. $\neg \text{eats}(\text{Anil}, w) \vee \text{eats}(\text{Harry}, w)$
8. $\text{killed}(g) \vee \text{alive}(g)$
9. $\neg \text{alive}(k) \vee \neg \text{killed}(k)$
10. $\text{likes}(\text{John}, \text{Peanuts})$.

Note: Statements " $\text{food}(\text{Apple}) \wedge \text{food}(\text{vegetables})$ " and " $\text{eats}(\text{Anil}, \text{Peanuts}) \wedge \text{alive}(\text{Anil})$ " can be written in two separate statements.

- **Distribute conjunction \wedge over disjunction \neg .**

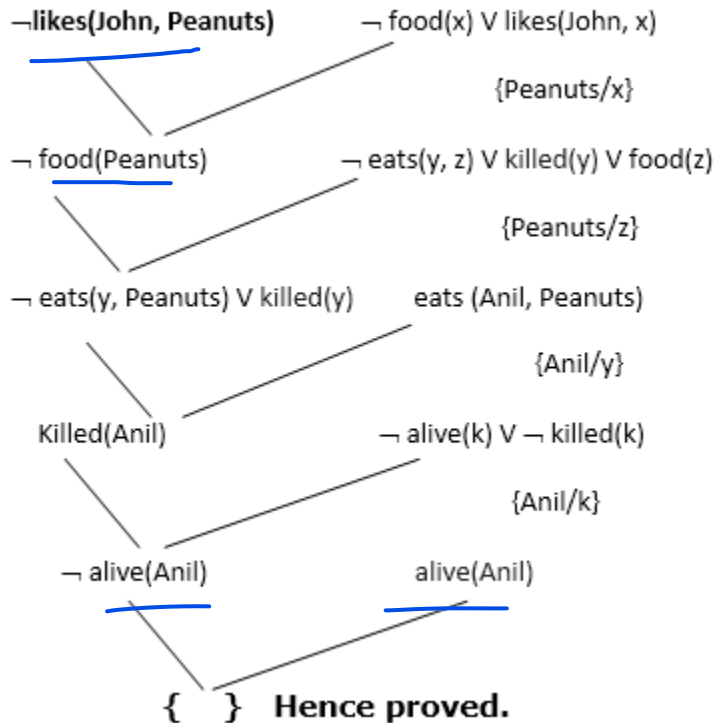
This step will not make any change in this problem.

Step-3: Negate the statement to be proved

In this statement, we will apply negation to the conclusion statements, which will be written as $\neg \text{likes}(\text{John}, \text{Peanuts})$

Step-4: Draw Resolution graph:

Now in this step, we will solve the problem by resolution tree using substitution. For the above problem, it will be given as follows:



Hence the negation of the conclusion has been proved as a complete contradiction with the given set of statements.

Explanation of Resolution graph:

- In the first step of resolution graph, $\neg \text{likes}(\text{John}, \text{Peanuts})$, and $\text{likes}(\text{John}, x)$ get resolved(canceled) by substitution of $\{ \text{Peanuts}/x \}$, and we are left with $\neg \text{food}(\text{Peanuts})$
- In the second step of the resolution graph, $\neg \text{food}(\text{Peanuts})$, and $\text{food}(z)$ get resolved (canceled) by substitution of $\{ \text{Peanuts}/z \}$, and we are left with $\neg \text{eats}(y, \text{Peanuts}) \vee \text{killed}(y)$.
- In the third step of the resolution graph, $\neg \text{eats}(y, \text{Peanuts})$ and $\text{eats}(\text{Anil}, \text{Peanuts})$ get resolved by substitution $\{ \text{Anil}/y \}$, and we are left with $\text{Killed}(\text{Anil})$.
- In the fourth step of the resolution graph, $\text{Killed}(\text{Anil})$ and $\neg \text{killed}(k)$ get resolve by substitution $\{ \text{Anil}/k \}$, and we are left with $\neg \text{alive}(\text{Anil})$.
- In the last step of the resolution graph $\neg \text{alive}(\text{Anil})$ and $\text{alive}(\text{Anil})$ get resolved.