Unification
In Predicate Logic

By Dr. U. B. Chavan

Unification

- Unification is a process of finding a substitution of terms for variables in two logical expressions so that the expressions become **identical**.
- This process is used in many areas of artificial intelligence, including theorem proving, natural language processing, and machine learning.

Unification in Propositional Logic

- **Easy to Determine That Two Literals Can Not Both Be True at the Same Time**

  For example
  The literals Man(Sunil) and ~Man(Sunil) are contradictory,
  while the literals Man(Sunil) and ~Man(Arun) are not contradictory.

Unification in Predicate Logic

- **Matching Process Is More Complicated, Since Bindings of Variables Must Be Considered**

  Two literals are said to be unifiable if there exists a substitution of terms for variables that makes the literals identical

Unification Algorithm

- **To make two expressions look identical by using substitution.**

UNIFY Algorithm:

1. Takes two expressions as input.

2. Returns a unifier if one exists, otherwise returns fail.

3. Substitution means replacing one variable with another term.

4. UNIFY(p, q)=0 where SUBST(0, p) = SUBST(e, q).

Unification Example 1

- Consider the following two logical expressions:

  P(x, y)

  P(a, f(z))

- These two expressions are not initially identical, but they can be made identical by using unification. To do this, we need to find a substitution of terms for variables that makes the two expressions identical.

- In this case, the substitution {x = a, y = f(z)} makes the two expressions identical.

- first expression will be identical to the second expression and the substitution set will be: [a/x, f(z)/y]

Conditions for unification:

The two expressions must have the same number of arguments.

The two expressions must have the same predicate symbol.

Unification fails, if there are two similar variables present in same expression.

Thank You!