

Unification

To apply inference rules inference system must be able to determine when two expressions are the **same** or *match*.

In **propositional calculus**, this is trivial: two expressions match if and only if they are syntactically identical.

In **predicate calculus**, the process of matching two sentences is complicated by the existence of variables in the expressions. Universal instantiation allows universally quantified variables to be replaced by terms from the domain.

Unification

Unification is an algorithm for determining the substitutions needed to make two predicate calculus expressions **match**.

Ex: Where `socrates` in `man(socrates)` was substituted for `X` in $\forall X(\text{man}(X) \Rightarrow \text{mortal}(X))$. This allowed the application of modus ponens and the conclusion **mortal(socrates)**.

Ex: The unification was seen previously when dummy variables were discussed. Because $p(X)$ and $p(Y)$ are equivalent, `Y` may be substituted for `X` to make the sentences match.

Unification

Unification and inference rules such as modus ponens allow us to make inferences on a set of logical assertions. To do this, the **logical database** must be expressed in an **appropriate form**.

An essential aspect of this form is the requirement that:

- All Variables Be Universally Quantified. This allows full freedom in computing substitutions.
- Existentially quantified variables may be eliminated from sentences in the database by replacing them with the constants that make the sentence true.

Unification

For example, $\exists X \text{parent}(X, \text{tom})$ could be replaced by the expression **parent(bob, tom)** or **parent(mary, tom)**, assuming that **bob and Mary are tom's parents under the interpretation.**

The process of eliminating existentially quantified variables is **complicated** by the fact that the value of these substitutions may depend on the value of other variables in the expression. For example, in the expression $\forall X \exists Y \text{mother}(X, Y)$, the value of the existentially quantified variable Y depends on the value of X . that why we use *Skolemization*

Unification

Some instances of the expression

$foo(X, a, goo(Y))$.

generated by legal substitutions are given below:

- 1) $foo(fred, a, goo(Z))$
- 2) $foo(W, a, goo(jack))$
- 3) $foo(Z, a, goo(moo(Z)))$

In this example, the substitution instances or *unifications* that would make the initial expression identical to each of the other three are written as the sets:

- 1) $\{fred/X, Z/Y\}$
- 2) $\{W/X, jack/Y\}$
- 3) $\{Z/X, moo(Z)/Y\}$

1. Expression must have the same name.
2. The number of attribute of two values must be equivalent .
3. Can replace variable with variable
4. Constant with variable
5. Expression with variable

Unification

In defining the unification algorithm that computes the substitutions required to match two expressions, a number of issues must be taken into account. **First**, although a constant may be systematically substituted for a variable, any constant is considered a “ground instance” and may not be replaced. Neither can two different ground instances be substituted for one variable. **Second**, a variable cannot be unified with a term containing that variable. X cannot be replaced by $p(X)$ as this creates an infinite expression: $p(p(p(p(\dots X)\dots)))$. The test for this situation is called the *occurs check*.

Unification

Furthermore, a problem-solving process often requires multiple inferences and, consequently, multiple successive unifications. Logic problem solvers must maintain **consistency** of variable substitutions. It is important that any unifying substitution be made consistently across all occurrences within the scope of the variable in both expressions being matched.

most general unifier: This is important, as will be seen in the next example, because, if generality is lost in the solution process, it may lessen the scope of the eventual solution or even eliminate the possibility of a solution entirely.

Unification

Unification Algorithm

- Need a procedure to find a MGU given a set of expressions
- Requirements:
 - stop after a finite number of steps
 - return an MGU if the set is unifiable
 - state that the set is not unifiable otherwise
- There are many possibilities
- We go for a recursive procedure.

Basic ideas

- Given a set of expressions $\{E_1, \dots, E_k\}$
- Find a **disagreement set**
- Build a substitution that can eliminate the disagreement

Unification

Definition (Disagreement Set)

The disagreement set of a nonempty set of expressions W is obtained by finding the first position (starting from the left) at which not all the expressions in the W have the same symbol. We then extract, from each expression, the sub-expression that begins with the symbol occupying that position. The set of these sub-expressions is the **Disagreement Set**.

Example (Disagreement Set)

Consider the set $\{P(a), P(x)\}$, the **Disagreement Set** is $\{a, x\}$. because the first position at which the string of symbols $P(a)$ and $P(x)$ differ is the position number 3. The sub-expression starting from position 3 is a and x respectively.

Unification

Example (Disagreement elimination)

Consider the set $\{P(a), P(x)\}$. These expressions are not identical.

- They disagree because of the arguments a and x
- The disagreement set here is $\{a, x\}$
- Since x is a variable, we can eliminate this disagreement by using the substitution $\theta = \{a/x\}$
- $P(a)\theta = P(x)\theta = P(a)$

Example (Disagreement Set)

Find the Disagreement Set for

$$W = \{P(x, f(y, z)), P(x, a), P(x, g(h(k(x))))\}$$

Sol.

$$D = \{f(y, z), a, g(h(k(x)))\}$$

Unification

Basic Steps

- 1 Set $k = 0$, $W_0 = W$ and $\sigma_0 = \epsilon$
- 2 If W_k is a **singleton**, STOP, σ_k is a MGU. Otherwise, find the disagreement set D_k for W_k .
- 3 If there is a pair $\langle v_k, t_k \rangle$ such that $v_k, t_k \in D_k$, v_k is a variable that **does not occur** in t_k go to step 4, otherwise STOP, W is not unifiable.
- 4 Let $\sigma_{k+1} = \sigma_k \circ \{t_k/v_k\}$ and $W_{k+1} = W_k\{t_k/v_k\}$.
- 5 Set $k = k + 1$ go to step 2.

Note

In step 4 $W_{k+1} = W_k\{t_k/v_k\} = W\sigma_{k+1}$ because composition of substitutions is associative.

Unification

Example: find M.g.u for

$$W = \{ p(a, x, f(g(y))), \\ P(z, f(z), f(u)) \}$$

Solution:

$$K=0$$

$$w_k = w = \{ \}$$

$$\sigma_0 = \varepsilon$$

Unification feasible = true

$$D_0 = \{ a, z \}$$

$$t_0 = a, v_0 = z \text{ such that } a/z$$

$$\sigma_1 = \varepsilon\{a/z\} = \{a/z\}$$

$$w_1 = \{ p(a, x, f(g(y))), p(a, f(a), f(u)) \}$$

Unification

$K=1$

$D1 = \{x, f(a)\}$

$t_1 = f(a)$

$v_1 = x$

Such that $f(a)/x$

$\sigma_2 = \sigma_1\{f(a)/x\} = \{a/z\}\{f(a)/x\} = \{a/z, f(a)/z\}$

$w_2 = \{p(a, f(a), f(g(y)), p(a, f(a), f(u)))\}$

$K=2$

$D2 = \{g(y), u\}$

$\sigma_3 = \sigma_2\{g(y)/u\} = \{a/z, f(a)/x, g(y)/u\}$

$w_3 = \{p(a, f(a), f(g(y)), p(a, f(a), f(g(y))))\}$

Singleton $M.g.u = \{a/z, f(a)/x, g(y)/u\}$

Unification

Find most general unifier

$W = \{p(a, Y), p(X, f(b))\}$

Solution

$k=0$;

$$w_k = w = \{\} \sigma_0 = \epsilon$$

Unification feasible= True

$$D_0 = \{a, X\}$$

$t_0 = a, v_0 = X$ such that $a|x$

$$\sigma_1 = \sigma_0\{a|X\} = \epsilon\{a|X\} = \{a|X\}$$

$$w_1 = \{p(a, Y), p(a, f(b))\}$$

$K=1$

$$D_1 = \{Y, f(b)\}$$

$t_1 = f(b), v_1 = Y$ such that $f(b)|Y$

$$\sigma_2 = \sigma_2\{a|X\} = \{a|X\}\{f(b)|Y\} = \{a|X, f(b)|Y\}$$

$$w_2 = \{p(a, f(b)), p(a, f(b))\}$$

Singleton $\therefore M. G. U = \{a|X, f(b)|Y\}$

Unification

Example (Unification Algorithm)

Find a most general unifier for the set

$$W = \{P(a, y), P(x, f(b))\}$$

Sol.

$$\theta = \{a/x, f(b)/y\}$$

Example (Unification Algorithm)

Find a most general unifier for the set

$$W = \{P(a, x, f(g(y))), P(z, f(z), f(u))\}$$

Sol.

$$\theta = \{a/z, f(a)/x, g(y)/u\}$$

Example (Unification Algorithm)

Determine whether or not the set

$$W = \{Q(f(a), g(x)), Q(y, y)\} \text{ is unifiable.}$$

Sol.

W is not unifiable

Unification

Exercise

Determine whether each of the following set of expressions is unifiable. If yes give a MGU

1 $W = \{Q(a, x, f(x)), Q(a, y, y)\}$

2 $W = \{Q(x, y, z), Q(u, h(v, v), u)\}$

3. $w = \{p(x, a), p(x, b)\}$

4. $w = \{p(x, f(a)), p(x, f(b))\}$