Unification algorithm

Given in terms of steps

- each step considers one equation from the set
- steps performed in any order
- until finished: i.e., until no more steps can be applied
- results merged back into the set after each step

“x occurs in t”: \( x \neq t \) and \( x \in Var(t) \)
Unification algorithm

\[
f_1(s_1, \ldots, s_m) \overset{?}{=} f_2(t_1, \ldots, t_n)
\]
where \( f_1 \neq f_2 \)

\[
f(s_1, \ldots, s_m) \overset{?}{=} f(t_1, \ldots, t_m)
\]

\[
x \overset{?}{=} x
\]

\[
t \overset{?}{=} x \text{ where } t \text{ is not a variable}
\]

\[
x \overset{?}{=} t \text{ where } x \text{ occurs in } t
\]

\[
x \overset{?}{=} t \text{ where } x \text{ does not occur in } t
\]
and \( x \) occurs elsewhere

Fail ("function clash")

Replace by \( s_1 =? t_1, \ldots, s_m =? t_m \)

Delete

Replace by \( x =? t \)

Fail ("Occur-check failure")

Replace all \textit{other} occurrences of \( x \) by \( t \)
Unification

If the final set of equations is

\[ \{ x_1 = ? s_1, \ldots, x_k = ? s_k \} \]

then the most general unifier (mgu) is

\[ [s_1/x_1, \ldots, s_k/x_k] \]
\{\text{add}(X, X, s(s(0))) =? \text{add}(s(U), V, s(W))\}

\{X =? s(U), X =? V, s(s(0)) =? s(W)\}

\{X =? s(U), s(U) =? V, s(s(0)) =? s(W)\}

\{X =? s(U), V =? s(U), s(s(0)) =? s(W)\}

\{X =? s(U), V =? s(U), s(0) =? W\}

\{X =? s(U), V =? s(U), W =? s(0)\}

The unifier is \[s(U)/X, s(U)/V, s(0)/W\]
\{f(X_1, f(X, Y)) = f(f(Y, h(X)), X_1)\}
\{X_1 = f(Y, h(X)), f(X, Y) = X_1\}
\{X_1 = f(Y, h(X)), f(X, Y) = f(Y, h(X))\}
\{X_1 = f(Y, h(X)), X = Y, Y = h(X)\}
\{X_1 = f(Y, h(Y)), X = Y, Y = h(Y)\}

Occur-check failure
\[
\{ f(X_1, X_1) =? f(f(X_2, X_3), h(X_4)) \}
\]
\[
\{ X_1 =? f(X_2, X_3), \quad X_1 =? h(X_4) \}
\]
\[
\{ X_1 =? f(X_2, X_3), \quad f(X_2, X_3) =? h(X_4) \}
\]

Function clash