Unification

Unification algorithm

–

the heart

of the computation model

of logic programs

Substitution

Finite set of the form

$$\theta = \{v_1 \mapsto t_1, \ldots, v_n \mapsto t_n\}$$

v_i-s

Distinct variables

t_i-s

Terms with $t_i \neq v_i$

Binding

$v_i \mapsto t_i$

Application of Substitution

Substitution

$$\theta = \{v_1 \mapsto t_1, \ldots, v_n \mapsto t_n\}$$

Applied to an expression E

$$E\theta$$

(Instance of E wrt θ)

Simultaneously replacing

each occurrence

of v_i in E

with t_i

Example (Application)

$$E = p(x, y, f(a))$$

$$\theta = \{x \mapsto b, y \mapsto x\}$$

$$E\theta = p(b, x, f(a))$$

Note that x not substituted second time

Composition

$$\theta = \{v_1 \mapsto t_1, \ldots, v_n \mapsto t_n\}$$

$$\sigma = \{u_1 \mapsto s_1, \ldots, u_m \mapsto s_m\}$$

Composition

$$\theta\sigma$$

Obtained from the set

$$\begin{aligned} \{ \quad & v_1 \mapsto t_1\sigma, \ldots, v_n \mapsto t_n\sigma, \\ & u_1 \mapsto s_1, \ldots, u_m \mapsto s_m \quad \} \end{aligned}$$

By deleting

all $u_i \mapsto s_i$-s with $u_i \in \{v_1, \ldots, v_n\}$,

all $v_i \mapsto t_i\sigma$-s with $v_i = t_i\sigma$.

Example (Composition)

$$\theta = \{x \mapsto f(y), y \mapsto z\}$$

$$\sigma = \{x \mapsto a, y \mapsto b, z \mapsto y\}$$

$$\theta\sigma = \{x \mapsto f(b), z \mapsto y\}$$

Empty Substitution

Empty substitution

$$\varepsilon$$

The substitution is the empty set

For all expressions E

$$E\varepsilon = E$$

Properties

$$\theta\varepsilon = \varepsilon\theta = \theta$$

$$(E\theta)\sigma = E(\theta\sigma)$$

$$(\theta\sigma)\lambda = \theta(\sigma\lambda)$$

Example (Properties)

Given

$$\theta = \{x \mapsto f(y), y \mapsto z\}$$

$$\sigma = \{x \mapsto a, z \mapsto b\}$$

$$E = p(x, y, g(z))$$

Then

$$\theta\sigma = \{x \mapsto f(y), y \mapsto b, z \mapsto b\}$$

$$E\theta = p(f(y), z, g(z))$$

$$(E\theta)\sigma = p(f(y), b, g(b))$$

$$E(\theta\sigma) = p(f(y), b, g(b))$$

Renaming Substitution

$$\theta = \{x_1 \mapsto y_1, \ldots, x_n \mapsto y_n\}$$

is a renaming substitution

iff

y_i-s are distinct variables.

Renaming an Expression

V – the set of variables of the expression E

$$\theta = \{x_1 \mapsto y_1, \ldots, x_n \mapsto y_n\}$$

is a renaming substitution for E

iff

θ is a renaming substitution and

$$\{x_1, \ldots, x_n\} \subseteq V$$

$$(V \setminus \{x_1, \ldots, x_n\}) \cap \{y_1, \ldots, y_n\} = \emptyset$$

Variants

Expression E

and

expression F

are variants

iff

there exist substitutions θ and σ

such that

$E\theta = F$

and

$F\sigma = E$

Variants and Renaming

E and F are variants

iff

there exist *renaming* substitutions θ and σ

such that

$E\theta = F$

and

$F\sigma = E$

More General

A substitution θ

is more general

than a substitution σ

iff

there exists a substitution λ

such that

$$\sigma = \theta\lambda$$

Example (More General)

$$\theta = \{x \mapsto y, u \mapsto f(y, z)\}$$

is more general than

$$\sigma = \{x \mapsto z, y \mapsto z, u \mapsto f(z, z)\}$$

because

$$\sigma = \theta\lambda$$

where $\lambda = \{y \mapsto z\}$

Unifier

Substitution θ

is a *unifier*

of expressions E and F

iff

$$E\theta = F\theta$$

Example (Unifier)

$$\theta = \{x \mapsto f(b), y \mapsto b, z \mapsto u\}$$

is a unifier of

$E = f(x, b, g(z))$ and $F = f(f(y), y, g(u))$:

$$E\theta = f(f(b), b, g(u))$$

$$F\theta = f(f(b), b, g(u))$$

Unifier (Contd.)

σ is a unifier

of a set of expression pairs

$\{\langle E_1, F_1\rangle, \ldots, \langle E_n, F_n\rangle\}$

iff

σ is a unifier of

E_i and F_i for each $1 \leq i \leq n$,

i.e., iff

$E_1\sigma = F_1\sigma$

$\ldots$

$E_n\sigma = F_n\sigma$

Most General Unifier (mgu)

A unifier θ of E and F

is most general

iff

θ is more general

than any other unifier of E and F

Example (mgu)

$$E = f(x, b, g(z)) \ F = f(f(y), y, g(u))$$

Unifiers of E and F

(infinitely many):

$$\theta_1 = \{x \mapsto f(b), y \mapsto b, z \mapsto u\}$$

$$\theta_2 = \{x \mapsto f(b), y \mapsto b, u \mapsto z\}$$

$$\theta_3 = \{x \mapsto f(b), y \mapsto b, z \mapsto a, u \mapsto a\}$$

$$\theta_4 = \{x \mapsto f(b), y \mapsto b, z \mapsto b, u \mapsto b\}$$

$$\theta_5 = \{x \mapsto f(b), y \mapsto b, z \mapsto u, w \mapsto d\}$$

$$\ldots$$

But

Example (mgu, Contd.)

mgu-s of E and F:

$$\theta_1 = \{x \mapsto f(b), y \mapsto b, z \mapsto u\}$$

$$\theta_2 = \{x \mapsto f(b), y \mapsto b, u \mapsto z\}$$

θ_1 is more general than θ_2:

$$\theta_2 = \theta_1\lambda_1 \text{ with } \lambda_1 = \{u \mapsto z\}$$

and

θ_2 is more general than θ_1:

$$\theta_1 = \theta_2\lambda_2 \text{ with } \lambda_2 = \{z \mapsto u\}$$

Note

λ_1 and λ_2 are renaming substitutions

Equivalence of mgu-s

Most general unifier of two expressions

is unique

up to variable renaming

Unification Algorithm

Rule-based approach

General form of rules:

$$P;\ \sigma \Longrightarrow Q;\ \theta$$

or

$$P;\ \sigma \Longrightarrow \bot$$

where

$\bot$ denotes failure

σ and θ are substitutions

P and Q are sets of expression pairs:

$$\{\langle E_1, F_1\rangle, \ldots, \langle E_n, F_n\rangle\}$$

Unification Rules

Trivial:

$$\{\langle s, s\rangle\} \cup P';\ \sigma \Longrightarrow P';\ \sigma$$

Decomposition:

$$\{\langle f(s_1, \ldots, s_n), f(t_1, \ldots, t_n)\rangle\} \cup P';\ \sigma \Longrightarrow$$
$$\{\langle s_1, t_1\rangle, \ldots, \langle s_n, t_n\rangle\} \cup P';\ \sigma$$

Symbol Clash:

$$\{\langle f(s_1, \ldots, s_n), g(t_1, \ldots, t_m)\rangle\} \cup P';\ \sigma \Longrightarrow \perp$$

if $f \neq g$.

Unification Rules (Contd.)

Orient:

$$\{\langle t, x\rangle\} \cup P';\ \sigma \Longrightarrow \{\langle x, t\rangle\} \cup P';\ \sigma$$

if t is not a variable.

Occurs Check:

$$\{\langle x, t\rangle\} \cup P';\ \sigma \Longrightarrow \bot$$

if x occurs in t and $x \neq t$.

Variable Elimination:

$$\{\langle x, t\rangle\} \cup P';\ \sigma \Longrightarrow P'\theta;\ \sigma\theta$$

if x does not occur in t, and $\theta = \{x \mapsto t\}$.

Unification Algorithm

In order to unify s and t

Create initial system $\{\langle s, t\rangle\}; \varepsilon$

Apply successively unification rules.

Termination

The unification algorithm

terminates

either with $\perp$

or with $\emptyset; \sigma$

Soundness

If

$$P;\ \varepsilon \Longrightarrow^{+} \emptyset;\ \sigma$$

then

σ is a unifier of P.

Completeness

For any unifier θ of P

the unification algorithm

finds a unifier σ of P

such that σ is more general than θ

Major Result

If two expressions are unifiable

unification algorithm

computes their mgu

Example 1

Unify $p(f(a), g(x))$ and $p(y, y)$.

$$
\begin{aligned}
\{\langle p(f(a), g(x)), p(y, y)\rangle\};\ \varepsilon &\quad \Longrightarrow_{\mathsf{Dec}} \\
\{\langle f(a), y\rangle, \langle g(x), y\rangle\};\ \varepsilon &\quad \Longrightarrow_{\mathsf{Or}} \\
\{\langle y, f(a)\rangle, \langle g(x), y\rangle\};\ \varepsilon &\quad \Longrightarrow_{\mathsf{VarEl}} \\
\{\langle g(x), f(a)\rangle\};\ \{x \mapsto f(a)\} &\quad \Longrightarrow_{\mathsf{SymCl}} \\
\perp &
\end{aligned}
$$

Not unifiable

Example 2

Unify $p(a, x, h(g(z)))$ and $p(z, h(y), h(y))$.

$$
\begin{aligned}
\{\langle p(a, x, h(g(z))), p(z, h(y), h(y))\rangle\};\ \varepsilon \quad &\Longrightarrow_{\mathsf{Dec}} \\
\{\langle a, z\rangle, \langle x, h(y)\rangle, \langle h(g(z)), h(y)\rangle\};\ \varepsilon \quad &\Longrightarrow_{\mathsf{Or}} \\
\{\langle z, a\rangle, \langle x, h(y)\rangle, \langle h(g(z)), h(y)\rangle\};\ \varepsilon \quad &\Longrightarrow_{\mathsf{VarEl}} \\
\{\langle x, h(y)\rangle, \langle h(g(a)), h(y)\rangle\};\ \{z \mapsto a\} \quad &\Longrightarrow_{\mathsf{VarEl}} \\
\{\langle h(g(a)), h(y)\rangle\};\ \{z \mapsto a, x \mapsto h(y)\} \quad &\Longrightarrow_{\mathsf{Dec}} \\
\{\langle g(a), y\rangle\};\ \{z \mapsto a, x \mapsto h(y)\} \quad &\Longrightarrow_{\mathsf{Or}} \\
\{\langle y, g(a)\rangle\};\ \{z \mapsto a, x \mapsto h(y)\} \quad &\Longrightarrow_{\mathsf{VarEl}} \\
\emptyset;\ \{z \mapsto a, x \mapsto h(g(a)), y \mapsto g(a)\} \quad &
\end{aligned}
$$

Answer: $\{z \mapsto a, x \mapsto h(g(a)), y \mapsto g(a)\}$

Example 3

Unify $p(x, x)$ and $p(y, f(y))$.

$$
\begin{aligned}
\{\langle p(x,x), p(y, f(y))\rangle\};\ \varepsilon \quad & \Longrightarrow_{\mathsf{Dec}} \\
\{\langle x, y\rangle, \langle x, f(y)\rangle\};\ \varepsilon \quad & \Longrightarrow_{\mathsf{VarEl}} \\
\{\langle y, h(y)\rangle\};\ \{x \mapsto y\} \quad & \Longrightarrow_{\mathsf{OccCh}} \\
\bot \quad &
\end{aligned}
$$

Not unifiable

Example 3 on Prolog

```
?- p(X,X)=p(Y,f(Y)).

X = f(f(f(f(f(f(f(f(f(f(...))))))))))

Y = f(f(f(f(f(f(f(f(f(f(...))))))))))

Yes
```

Occurrence Check

Prolog unification algorithm skips

Occurrence Check

Reason:

Occurrence Check can be expensive

Justification:

Most of the time this rule is not needed

Drawback:

Sometimes might lead to incorrect answers

Example

```
less(X,s(X)).

foo:-less(s(Y),Y).



?- foo.

Yes
```