

#### Substitutions

• A  $T(\mathcal{F}, \mathcal{V})$ -substitution: A function  $\sigma : \mathcal{V} \to T(\mathcal{F}, \mathcal{V})$ , whose domain

 $\mathcal{D}om(\sigma) := \{ x \mid \sigma(x) \neq x \}$ 

is finite.

• Range of a substitution  $\sigma$ :

 $\mathcal{R}an(\sigma) := \{ \sigma(x) \mid x \in \mathcal{D}om(\sigma) \}.$ 

• Variable range of a substitution  $\sigma$ :

 $\mathcal{VR}an(\sigma) := \mathcal{V}ar(\mathcal{R}an(\sigma)).$ 

▶ Notation: lower case Greek letters  $\sigma, \vartheta, \varphi, \psi, \ldots$ Identity substitution:  $\varepsilon$ .

#### Unification

Solving term equations: Given: Two terms s and t. Find: A substitution  $\sigma$  such that  $\sigma(s)=\sigma(t).$ 

#### Substitutions

▶ Notation: If  $\mathcal{D}om(\sigma) = \{x_1, \ldots, x_n\}$ , then  $\sigma$  can be written as the set

$$\{x_1 \mapsto \sigma(x_1), \ldots, x_n \mapsto \sigma(x_n)\}.$$

► Example:

$$\{x \mapsto i(y), y \mapsto e\}.$$

#### Substitutions

 $\blacktriangleright$  The substitution  $\sigma$  can be extended to a mapping

 $\sigma: T(\mathcal{F}, \mathcal{V}) \to T(\mathcal{F}, \mathcal{V})$ 

by induction:

$$\sigma(f(t_1,\ldots,t_n)) = f(\sigma(t_1),\ldots,\sigma(t_n)).$$

► Example:

$$\begin{split} \sigma &= \{x \mapsto i(y), y \mapsto e\}.\\ t &= f(y, f(x, y))\\ \sigma(t) &= f(e, f(i(y), e)) \end{split}$$

• Sub : The set of substitutions.

5 / 1

#### More Notions about Substitutions

Algorithm for obtaining a set representation of a composition of two substitutions in a set form.

► Given:

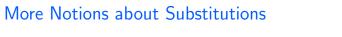
 $\theta = \{x_1 \mapsto t_1, \dots, x_n \mapsto t_n\}$  $\sigma = \{y_1 \mapsto s_1, \dots, y_m \mapsto s_m\},\$ 

the set representation of their composition  $\sigma\theta$  is obtained from the set

 $\{x_1 \mapsto \sigma(t_1), \ldots, x_n \mapsto \sigma(t_n), y_1 \mapsto s_1, \ldots, y_m \mapsto s_m\}$ 

by deleting

- all  $y_i \mapsto s_i$ 's with  $y_i \in \{x_1, \dots, x_n\}$ ,
- all  $x_i \mapsto \sigma(t_i)$ 's with  $x_i = \sigma(t_i)$ .



• Composition of  $\vartheta$  and  $\sigma$ :

 $\sigma\vartheta(x) := \sigma(\vartheta(x)).$ 

- Composition of two substitutions is again a substitution.
- Composition is associative but not commutative.

More Notions about Substitutions

Example (Composition)

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

#### More Notions about Substitutions

• t is an instance of s iff there exists a  $\sigma$  such that

 $\sigma(s) = t.$ 

- Notation:  $t \gtrsim s$  (or  $s \lesssim t$ ).
- Reads: t is more specific than s, or s is more general than t.
- $\blacktriangleright$   $\gtrsim$  is a quasi-order.
- ► Strict part: >.
- Example:  $f(e, f(i(y), e)) \gtrsim f(y, f(x, y))$ , because

 $\sigma(f(y, f(x, y))) = f(e, f(i(y), e)$ 

for  $\sigma = \{x \mapsto i(y), y \mapsto e\}$ 

9/1

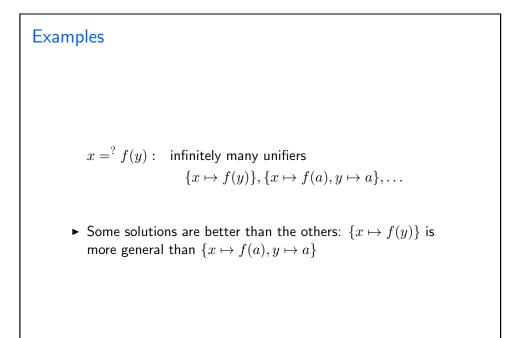
#### Examples

## $\begin{array}{rl} f(x) = \stackrel{?}{} f(a): & \mbox{exactly one unifier } \{x \mapsto a\} \\ & x = \stackrel{?}{} f(y): & \mbox{infinitely many unifiers} \\ & & \{x \mapsto f(y)\}, \{x \mapsto f(a), y \mapsto a\}, \dots \\ f(x) = \stackrel{?}{} g(y): & \mbox{no unifiers} \\ & x = \stackrel{?}{} f(x): & \mbox{no unifiers} \end{array}$

#### Unification

Syntactic unification:
Given: Two terms s and t.
Find: A substitution σ such that σ(s) = σ(t).
★ σ: a unifier of s and t.
★ σ: a solution of the equation s =? t.





#### Substitutions

#### Instantiation Quasi-Ordering

- A substitution  $\sigma$  is more general than  $\vartheta$ , written  $\sigma \lesssim \vartheta$ , if there exists  $\eta$  such that  $\eta \sigma = \vartheta$ .
- $\vartheta$  is called an instance of  $\sigma$ .
- ► The relation ≤ is quasi-ordering (reflexive and transitive binary relation), called instantiation quasi-ordering.
- $\blacktriangleright$   $\sim$  is the equivalence relation corresponding to  $\lesssim$  , i.e., the relation  $\lesssim \cap \gtrsim$  .

#### Example

Let 
$$\sigma = \{x \mapsto y\}, \ \rho = \{x \mapsto a, y \mapsto a\}, \ \vartheta = \{y \mapsto x\}.$$

- $\sigma \lesssim \rho$ , because  $\{y \mapsto a\}\sigma = \rho$ .
- $\sigma \lesssim \vartheta$ , because  $\{y \mapsto x\}\sigma = \vartheta$ .

• 
$$\vartheta \lesssim \sigma$$
, because  $\{x \mapsto y\} \vartheta = \sigma$ .

 $\blacktriangleright \ \sigma \sim \vartheta.$ 

13/1

#### Substitutions

#### Definition (Idempotent Substitution)

A substitution  $\sigma$  is idempotent iff  $\sigma \sigma = \sigma$ .

#### Example

Let  $\sigma = \{x \mapsto f(z), y \mapsto z\}, \ \vartheta = \{x \mapsto f(y), y \mapsto z\}.$ 

- $\sigma$  is idempotent.
- $\vartheta$  is not:  $\vartheta \vartheta = \sigma \neq \vartheta$ .

#### Substitutions

#### Definition (Variable Renaming)

A substitution  $\sigma = \{x_1 \mapsto y_1, x_2 \mapsto y_2, \dots, x_n \mapsto y_n\}$  is called variable renaming iff  $\{x_1, \dots, x_n\} = \{y_1, \dots, y_n\}$ . (Permuting the domain variables.)

#### Example

- $\blacktriangleright \ \{x\mapsto y, y\mapsto z, z\mapsto x\} \text{ is a variable renaming}.$
- $\{x \mapsto a\}$ ,  $\{x \mapsto y\}$ , and  $\{x \mapsto z, y \mapsto z, z \mapsto x\}$  are not.

#### Substitutions

#### Lemma

 $\sigma \sim \vartheta$  iff there exists a variable renaming  $\rho$  such that  $\rho \sigma = \vartheta$ .

#### Proof.

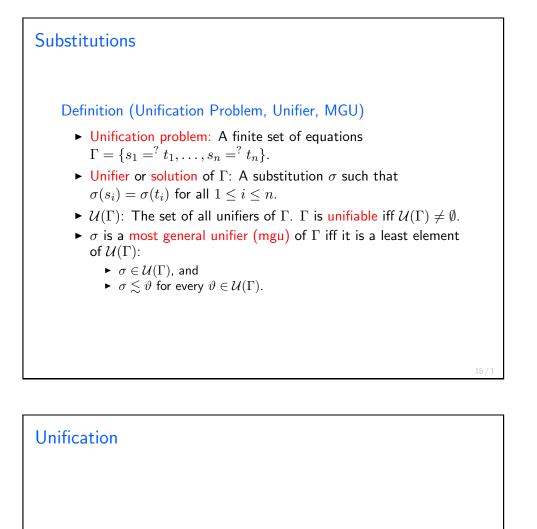
#### Exercise.

#### Example

- $\bullet \ \sigma = \{x \mapsto y\}.$
- $\blacktriangleright \ \vartheta = \{ y \mapsto x \}.$
- $\blacktriangleright \ \sigma \sim \vartheta.$
- $\blacktriangleright \ \{x \mapsto y, y \mapsto x\} \sigma = \vartheta.$



# $$\begin{split} \sigma &:= \{x \mapsto y\} \text{ is an mgu of } x = {}^{?} y. \\ \text{For any other unifier } \vartheta \text{ of } x = {}^{?} y, \sigma \lesssim \vartheta \text{ because} \\ \bullet \ \vartheta(x) &= \vartheta(y) = \vartheta \sigma(x). \\ \bullet \ \vartheta(y) &= \vartheta \sigma(y). \\ \bullet \ \vartheta(z) &= \vartheta \sigma(z) \text{ for any other variable } z. \\ \sigma' &:= \{x \mapsto z, y \mapsto z\} \text{ is a unifier but not an mgu of } x = {}^{?} y. \\ \bullet \ \sigma' &= \{y \mapsto z\}\sigma. \\ \bullet \ \{z \mapsto y\}\sigma' &= \{x \mapsto y, z \mapsto y\} \neq \sigma. \\ \sigma'' &= \{x \mapsto y, z_1 \mapsto z_2, z_2 \mapsto z_1\} \text{ is an mgu of } x = {}^{?} y. \\ \bullet \ \sigma &= \{z_1 \mapsto z_2, z_2 \mapsto z_1\}\sigma''. \\ \bullet \ \sigma'' \text{ is not idempotent.} \end{split}$$



Question: How to compute an mgu of an unification problem?

#### Rule-Based Formulation of Unification

- Unification algorithm in a rule-base way.
- Repeated transformation of a set of equations.
- The left-to-right search for disagreements: modeled by term decomposition.

#### The Inference System ${\mathfrak U}$

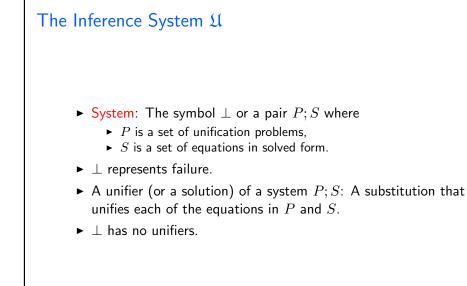
• A set of equations in solved form:

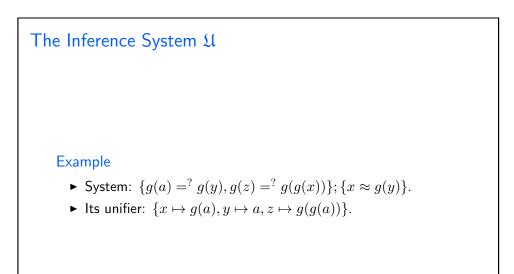
 $\{x_1 \approx t_1, \ldots, x_n \approx t_n\}$ 

where each  $x_i$  occurs exactly once.

- For each idempotent substitution there exists exactly one set of equations in solved form.
- ► Notation:
  - $[\sigma]$  for the solved form set for an idempotent substitution  $\sigma$ .
  - $\sigma_S$  for the idempotent substitution corresponding to a solved form set S.

.....





#### The Inference System ${\mathfrak U}$

Six transformation rules on systems:<sup>1</sup>

Trivial:

$$\{s = ?s\} \uplus P'; S \Leftrightarrow P'; S$$

**Decomposition:** 

 $\{f(s_1, \dots, s_n) \stackrel{?}{=} f(t_1, \dots, t_n)\} \uplus P'; S \Leftrightarrow$  $\{s_1 \stackrel{?}{=} t_1, \dots, s_n \stackrel{?}{=} t_n\} \cup P'; S, \text{ where } n \ge 0.$ 

Symbol Clash:

$$\{f(s_1,\ldots,s_n) = {}^? g(t_1,\ldots,t_m)\} \uplus P'; S \Leftrightarrow \bot, \text{ if } f \neq g.$$

 $^1 \oplus$  stands for disjoint union.

25 /

Unification with  ${\mathfrak U}$ 

In order to unify s and t:

- 1. Create an initial system  $\{s = {}^{?} t\}; \emptyset$ .
- 2. Apply successively rules from  $\mathfrak{U}$ .

The system  ${\mathfrak U}$  is essentially the Herbrand's Unification Algorithm.

#### The Inference System ${\mathfrak U}$

**Orient:**  $\{t = {}^? x\} \uplus P'; S \Leftrightarrow \{x = {}^? t\} \cup P'; S, \text{ if } t \notin \mathcal{V}.$ 

**Occurs Check:** 

 $\{x = {}^? t\} \uplus P'; S \Leftrightarrow \bot \text{ if } x \in \mathcal{V}ar(t) \text{ but } x \neq t.$ 

Variable Elimination:

 $\{x = {}^? t\} \uplus P'; S \Leftrightarrow \{x \mapsto t\}(P'); \{x \mapsto t\}(S) \cup \{x \approx t\},$  if  $x \notin \mathcal{V}ar(t).$ 

26 / 1

### Examples Example (Failure) Unify p(f(a), g(x)) and p(y, y). $\{p(f(a), g(x)) = {}^{?} p(y, y)\}; \emptyset \Longrightarrow_{\mathsf{Dec}}$ $\{f(a) = {}^{?} y, g(x) = {}^{?} y\}; \emptyset \Longrightarrow_{\mathsf{Or}}$ $\{y = {}^{?} f(a), g(x) = {}^{?} y\}; \emptyset \Longrightarrow_{\mathsf{VarEl}}$ $\{g(x) = {}^{?} f(a)\}; \{y \approx f(a)\} \Longrightarrow_{\mathsf{SymCl}}$ $\bot$

27 / 2

#### Examples

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

$$\begin{split} \{p(a, x, h(g(z))) =^{?} p(z, h(y), h(y))\}; & \emptyset \Longrightarrow_{\mathsf{Dec}} \\ \{a =^{?} z, x =^{?} h(y), h(g(z)) =^{?} h(y)\}; & \emptyset \Longrightarrow_{\mathsf{Or}} \\ \{z =^{?} a, x =^{?} h(y), h(g(z)) =^{?} h(y)\}; & \emptyset \Longrightarrow_{\mathsf{VarEl}} \\ \{x =^{?} h(y), h(g(a)) =^{?} h(y)\}; & \{z \approx a\} \Longrightarrow_{\mathsf{VarEl}} \\ \{h(g(a)) =^{?} h(y)\}; & \{z \approx a, x \approx h(y)\} \Longrightarrow_{\mathsf{Dec}} \\ & \{g(a) =^{?} y\}; & \{z \approx a, x \approx h(y)\} \Longrightarrow_{\mathsf{Or}} \\ & \{y =^{?} g(a)\}; & \{z \approx a, x \approx h(y)\} \Longrightarrow_{\mathsf{VarEl}} \\ & \emptyset; & \{z \approx a, x \approx h(g(a)), y \approx g(a)\}. \end{split}$$

 $29 \, / \, 1$ 

#### Properties of $\mathfrak{U}$ : Termination

#### Lemma

For any finite set of equations P, every sequence of transformations in  $\mathfrak U$ 

 $P; \emptyset \Leftrightarrow P_1; S_1 \Leftrightarrow P_2; S_2 \Leftrightarrow \cdots$ 

terminates either with  $\perp$  or with  $\emptyset$ ; S, with S in solved form.

#### **Examples**

Example (Failure) Unify p(x, x) and p(y, f(y)).

 $\begin{array}{l} \{p(x,x) =^? p(y,f(y))\}; \ \emptyset \Longrightarrow_{\mathsf{Dec}} \\ \{x =^? y,x =^? f(y)\}; \ \emptyset \Longrightarrow_{\mathsf{VarEl}} \\ \{y =^? f(y)\}; \ \{x \approx y\} \Longrightarrow_{\mathsf{OccCh}} \\ \bot \end{array}$ 

#### Properties of $\mathfrak{U}$ : Termination

#### Proof.

Complexity measure on the set P of equations:  $\langle n_1,n_2,n_3\rangle$ , ordered lexicographically on triples of naturals, where

 $n_1 =$  The number of distinct variables in P.

 $n_2 =$  The number of symbols in P.

 $n_3$  = The number of equations in P of the form  $t = {}^? x$  where t is not a variable.

#### Properties of $\mathfrak{U}$ : Termination

#### Proof [Cont.]

Each rule in  ${\mathfrak U}$  strictly reduces the complexity measure.

Rule	$n_1$	$n_2$	$n_3$
Trivial	$\geq$	>	
Decomposition	=	>	
Orient	=	=	>
Variable Elimination	>		

#### Properties of $\mathfrak{U}$ : Correctness

Notation:  $\Gamma$  for systems.

#### Lemma

For any transformation  $P; S \Leftrightarrow \Gamma$ , a substitution  $\vartheta$  unifies P; S iff it unifies  $\Gamma$ .

#### Properties of $\mathfrak{U}$ : Termination

#### Proof [Cont.]

- A rule can always be applied to a system with non-empty P.
- The only systems to which no rule can be applied are  $\bot$  and  $\emptyset; S.$
- ► Whenever an equation is added to S, the variable on the left-hand side is eliminated from the rest of the system, i.e. S<sub>1</sub>, S<sub>2</sub>,... are in solved form.

#### Corollary

If  $P; \emptyset \Leftrightarrow^+ \emptyset; S$  then  $\sigma_S$  is idempotent.

4/1

#### Properties of $\mathfrak{U}$ : Correctness

#### Proof.

**Occurs Check:** If  $x \in Var(t)$  and  $x \neq t$ , then

- x contains fewer symbols than t,
- $\vartheta(x)$  contains fewer symbols than  $\vartheta(t)$  (for any  $\vartheta$ ).

Therefore,  $\vartheta(x)$  and  $\vartheta(t)$  can not be unified.

Variable Elimination: From  $\vartheta(x) = \vartheta(t)$ , by structural induction on u:

$$\vartheta(u) = \vartheta\{x \mapsto t\}(u)$$

for any term, equation, or set of equations u. Then

 $\vartheta(P') = \vartheta\{x \mapsto t\}(P'), \qquad \vartheta(S') = \vartheta\{x \mapsto t\}(S').$ 

35 / 1

#### Properties of $\mathfrak{U}$ : Correctness

#### Theorem (Soundness)

If  $P; \emptyset \Leftrightarrow^+ \emptyset; S$ , then  $\sigma_S$  unifies any equation in P.

#### Proof.

By induction on the length of derivation, using the previous lemma and the fact that  $\sigma_S$  unifies S.

#### Observations

- ► 𝔅 computes an idempotent mgu.
- ► The choice of rules in computations via 𝔅 is "don't care" nondeterminism (the word "any" in Completeness Theorem).
- Any control strategy will result to an mgu for unifiable terms, and failure for non-unifiable terms.
- Any practical algorithm that proceeds by performing transformations of \$\mathcal{L}\$ in any order is
  - sound and complete,
  - generates mgus for unifiable terms.
- ▶ Not all transformation sequences have the same length.
- ► Not all transformation sequences end in exactly the same mgu.

#### Properties of $\mathfrak{U}$ : Correctness

#### Theorem (Completeness)

If  $\vartheta$  unifies every equation in P, then any maximal sequence of transformations  $P; \emptyset \Leftrightarrow \cdots$  ends in a system  $\emptyset; S$  such that  $\sigma_S \lesssim \vartheta$ .

#### Proof.

Such a sequence must end in  $\emptyset$ ; *S* where  $\vartheta$  unifies *S* (why?). For every binding  $x \mapsto t$  in  $\sigma_S$ ,  $\vartheta \sigma_S(x) = \vartheta(t) = \vartheta(x)$  and for every  $x \notin \mathcal{D}om(\sigma_S)$ ,  $\vartheta \sigma_S(x) = \vartheta(x)$ . Hence,  $\vartheta = \vartheta \sigma_S$ .

#### Corollary

If P has no unifiers, then any maximal sequence of transformations from  $P; \emptyset$  must have the form  $P; \emptyset \Leftrightarrow \cdots \Leftrightarrow \bot$ .

#### Example ?? in Prolog

Recall: Unification algorithm fails for p(x,x) = p(y,f(y)) because of the occurrence check.

But Prolog behaves differently:

Example (Infinite Terms)

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

X = f(\*\*), Y = f(\*\*).

In some versions of Prolog output looks like this: X = f(f(f(f(f(f(f(f(f(...)))))))))

#### 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 unexpected answers.

#### Occurrence Check

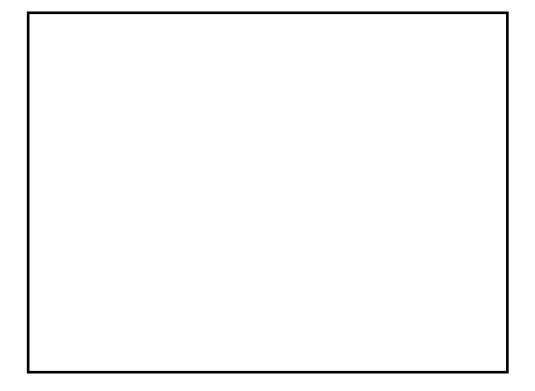
#### Example

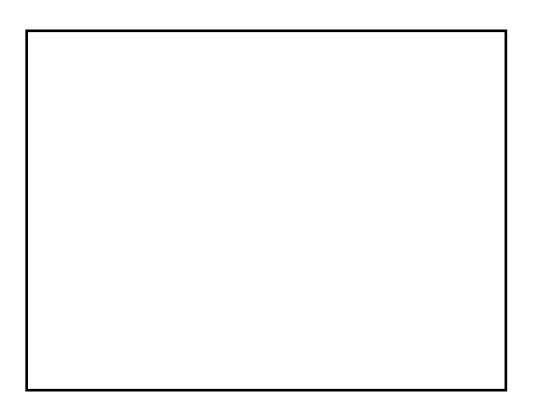
less(X,s(X)).foo:-less(s(Y),Y).

?- foo.

Yes

41/1





42/1