



<http://didawiki.di.unipi.it/doku.php/magistraleinformatica/psc/>

PSC 2020/21 (375AA, 9CFU)

Principles for Software Composition

Roberto Bruni

<http://www.di.unipi.it/~bruni/>

03 - Unification

Inference

SOS rule application?

1. a goal

$$(1 \oplus 2) \otimes (3 \oplus 4) \longrightarrow m$$

SOS rule application?

$$(prod) \frac{E_0 \longrightarrow n_0 \quad E_1 \longrightarrow n_1}{E_0 \otimes E_1 \longrightarrow n} \quad n = n_0 \cdot n_1$$

2. take a rule

$$(1 \oplus 2) \otimes (3 \oplus 4) \longrightarrow m$$

SOS rule application?

$$(prod) \frac{E_0 \longrightarrow n_0 \quad E_1 \longrightarrow n_1}{E_0 \otimes E_1 \longrightarrow n} \quad n = n_0 \cdot n_1$$

3. unify
(if possible)

$$E_0 = 1 \oplus 2$$

$$E_1 = 3 \oplus 4$$

$$n = m$$

$$(1 \oplus 2) \otimes (3 \oplus 4) \longrightarrow m$$

SOS rule application?

$$(prod) \frac{1 \oplus 2 \longrightarrow n_0 \quad 3 \oplus 4 \longrightarrow n_1}{(1 \oplus 2) \otimes (3 \oplus 4) \longrightarrow m} \quad m = n_0 \cdot n_1$$

4. instantiate

$$(1 \oplus 2) \otimes (3 \oplus 4) \longrightarrow m$$

SOS rule application?

$$(prod) \frac{\boxed{1 \oplus 2 \longrightarrow n_0} \quad \boxed{3 \oplus 4 \longrightarrow n_1}}{(1 \oplus 2) \otimes (3 \oplus 4) \longrightarrow m} \quad m = n_0 \cdot n_1$$

5. recursively solve subgoals

$$(1 \oplus 2) \otimes (3 \oplus 4) \longrightarrow m$$

SOS rule application?

$$(prod) \frac{1 \oplus 2 \longrightarrow \boxed{3} \quad 3 \oplus 4 \longrightarrow \boxed{7}}{(1 \oplus 2) \otimes (3 \oplus 4) \longrightarrow m} \quad m = \boxed{3 \cdot 7}$$

6. combine results

$$(1 \oplus 2) \otimes (3 \oplus 4) \longrightarrow m$$

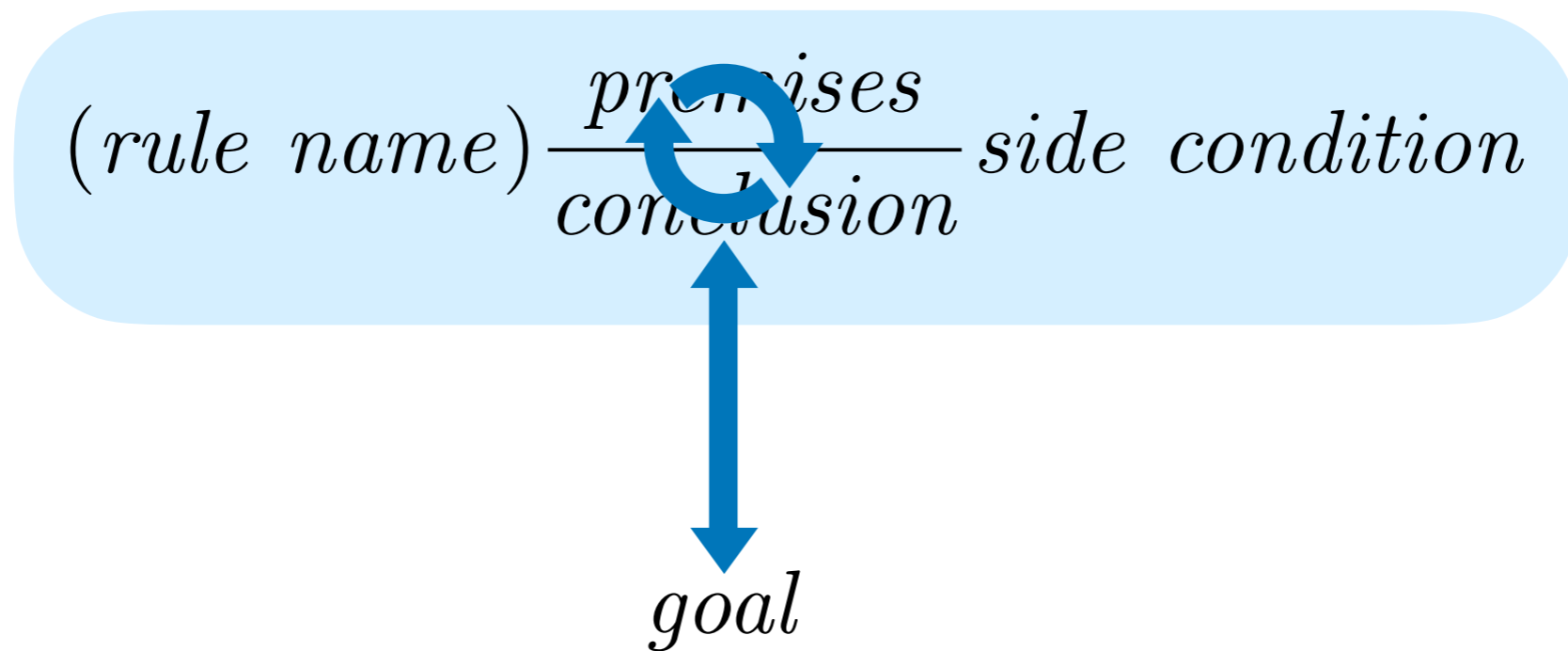
SOS rule application?

$$(prod) \frac{1 \oplus 2 \longrightarrow 3 \quad 3 \oplus 4 \longrightarrow 7}{(1 \oplus 2) \otimes (3 \oplus 4) \longrightarrow 21}$$

7. return results

$$(1 \oplus 2) \otimes (3 \oplus 4) \longrightarrow 21$$

Deduction process



Signatures

Unsorted signature

(Σ, ar)

a set of **function symbols**
(also called **operators**)

$$\Sigma = \{c, f, g, \dots\}$$

arity function

$$ar : \Sigma \rightarrow \mathbb{N}$$

each function symbol has an arity

Example

$$ar(c) = 0 \quad \text{constant}$$

$$ar(f) = 1 \quad \text{unary}$$

$$ar(g) = 2 \quad \text{binary}$$

$$ar(h) = 3 \quad \text{ternary}$$

Equivalently

$$(\Sigma, ar)$$

$$\begin{aligned} \text{let } \Sigma_n &\triangleq ar^{-1}(n) \\ &= \{f \in \Sigma \mid ar(f) = n\} \end{aligned}$$

a signature is an arity-indexed family of sets of operators

$$\Sigma = \{\Sigma_n\}_{n \in \mathbb{N}}$$

Terms over a signature

$\Sigma = \{\Sigma_n\}_{n \in \mathbb{N}}$ a signature

$X = \{x, y, z, \dots\}$ an infinite set of **variables**

$T_{\Sigma, X}$ denotes the set of all **terms** over Σ, X

it is the least set such that:

- if $x \in X$, then $x \in T_{\Sigma, X}$
- if $c \in \Sigma_0$, then $c \in T_{\Sigma, X}$
- if $f \in \Sigma_n$ and $t_1, \dots, t_n \in T_{\Sigma, X}$, then $f(t_1, \dots, t_n) \in T_{\Sigma, X}$

i.e. $T_{\Sigma, X} \ni t ::= x \mid c \mid f(t_1, \dots, t_n)$
 $x \in X \quad c \in \Sigma_0 \quad f \in \Sigma_n$

Vars

$$\Sigma = \{\Sigma_n\}_{n \in \mathbb{N}} \quad X = \{x, y, z, \dots\}$$

$t \in T_{\Sigma, X}$ $vars(t)$ set of variables that appears in t

$$vars : T_{\Sigma, X} \rightarrow \wp(X)$$

$$vars(x) \triangleq \{x\}$$

$$vars(c) \triangleq \emptyset$$

$$vars(f(t_1, \dots, t_n)) \triangleq \bigcup_{i=1}^n vars(t_i)$$

$$T_{\Sigma} \triangleq vars^{-1}(\emptyset) = \{t \in T_{\Sigma, X} \mid vars(t) = \emptyset\}$$

Running example

$$\Sigma_0 = \{0\}$$

$$\Sigma_1 = \{\text{succ}\}$$

$$\Sigma_2 = \{\text{plus}\}$$

$$\Sigma_n = \emptyset \quad \text{if } n > 2$$

Skill levels



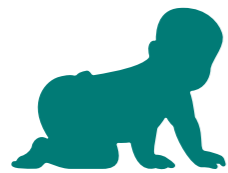
advanced
multitasking



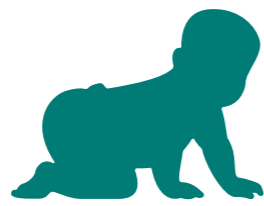
intermediate
does many things



novice
does one thing



beginner
does nothing



Exercise

let's complete the schema

t	$t \in ?$	$vars(t)$
0	$\begin{matrix} \bullet & \circ \\ T_\Sigma & T_{\Sigma, X} \end{matrix}$	\emptyset
x	$\begin{matrix} \circ & \bullet \\ T_\Sigma & T_{\Sigma, X} \end{matrix}$	$\{x\}$
$\text{succ}(0)$	$\begin{matrix} \circ & \circ \\ T_\Sigma & T_{\Sigma, X} \end{matrix}$	
$\text{succ}(x)$	$\begin{matrix} \circ & \circ \\ T_\Sigma & T_{\Sigma, X} \end{matrix}$	
$\text{succ}(\text{plus}(0), x)$	$\begin{matrix} \circ & \circ \\ T_\Sigma & T_{\Sigma, X} \end{matrix}$	
$\text{plus}(\text{succ}(x), 0)$	$\begin{matrix} \circ & \circ \\ T_\Sigma & T_{\Sigma, X} \end{matrix}$	
$\text{succ}(\text{succ}(0), \text{plus}(x))$	$\begin{matrix} \circ & \circ \\ T_\Sigma & T_{\Sigma, X} \end{matrix}$	
$\text{succ}(\text{plus}(w, z))$	$\begin{matrix} \circ & \circ \\ T_\Sigma & T_{\Sigma, X} \end{matrix}$	
$\text{plus}(\text{plus}(x, \text{succ}(y)), \text{plus}(0, \text{succ}(x)))$	$\begin{matrix} \circ & \circ \\ T_\Sigma & T_{\Sigma, X} \end{matrix}$	

Substitutions

$\rho : X \rightarrow T_{\Sigma, X}$ a **substitution** assigns terms to variables

we only consider substitutions that are identity everywhere, except for a finite number of cases, written

$$\rho = [x_1 = t_1, \dots, x_n = t_n] \quad \rho(x) = \begin{cases} t_i & \text{if } x = x_i \\ x & \text{otherwise} \end{cases}$$

all different

overloaded notation for the lifted function $\rho : T_{\Sigma, X} \rightarrow T_{\Sigma, X}$

$\rho(t)$ denotes the term obtained by simultaneous application of the substitution to all variable occurrences in t

$t\rho$ alternative notation

Example

$$\rho \triangleq [x = \text{succ}(y), y = 0]$$

$$t \triangleq \text{plus}(\text{plus}(x, y), \text{succ}(x))$$

$$t\rho = \text{plus}(\text{plus}(\text{succ}(y), 0), \text{succ}(\text{succ}(y)))$$

mgt relation

t is more general than t' if $\exists \rho. t' = t\rho$

in which case, we also say t' is an instance of t

$\text{plus}(x, \text{succ}(y))$	mgt	$\text{plus}(0, \text{succ}(\text{succ}(z)))$
$\text{plus}(0, x)$	mgt	$\text{plus}(y, 0)$
$\text{plus}(y, 0)$	mgt	$\text{plus}(0, x)$
$\text{plus}(0, x)$ $\text{plus}(y, 0)$	mgt	$\text{plus}(0, 0)$

mgt relation

mgt is transitive and reflexive

$$\begin{array}{c} / \\ t \text{ mgt } t \end{array}$$

if $(t_1 \text{ mgt } t_2)$ and $(t_2 \text{ mgt } t_3)$, then $(t_1 \text{ mgt } t_3)$

there are terms $t \neq t'$ such that $(t \text{ mgt } t')$ and $(t' \text{ mgt } t)$

$\text{succ}(x) \quad \text{succ}(y)$

mgt can be extended to substitutions pointwise

$\rho \text{ mgt } \rho'$ if $\exists \rho'' . \forall x . \rho'(x) = \rho''(\rho(x))$

Unification problem

The unification problem in its simplest form (syntactic, first-order) can be expressed as follows:

Given a set of potential equalities

$$\mathcal{G} = \{l_1 \stackrel{?}{=} r_1, \dots, l_n \stackrel{?}{=} r_n\}$$

where $l_1, \dots, l_n, r_1, \dots, r_n \in T_{\Sigma, X}$

can we find a substitution ρ such that

$$\forall i \in [1, n]. \rho(l_i) = \rho(r_i) \quad ?$$

we call such a ρ a solution of \mathcal{G}

$$\text{sols}(\mathcal{G}) \triangleq \{\rho \mid \forall i \in [1, n]. \rho(l_i) = \rho(r_i)\}$$

Unification problem

More interestingly, can we solve the following problem

Given a set of potential equalities

$$\mathcal{G} = \{l_1 \stackrel{?}{=} r_1, \dots, l_n \stackrel{?}{=} r_n\}$$

where $l_1, \dots, l_n, r_1, \dots, r_n \in T_{\Sigma, X}$

can we find a most general solution ρ ?

$$\rho \in \text{sols}(\mathcal{G}) \quad \text{and}$$

$$\forall \rho' \in \text{sols}(\mathcal{G}). \rho \text{ mgt } \rho'$$

Unification algorithm

Idea: we iteratively reduce the set \mathcal{G}
by solution-preserving transformations until
either a solution is found
or we can prove there is no solution



Solutions may not exist
and even if they exist may not be unique

Termination

$$\mathcal{G} = \{l_1 \stackrel{?}{=} r_1, \dots, l_n \stackrel{?}{=} r_n\}$$

\mathcal{G} and \mathcal{G}' are equivalent if $\text{sols}(\mathcal{G}) = \text{sols}(\mathcal{G}')$

the algorithm terminates successfully when we reach

$$\mathcal{G}' = \{x_1 \stackrel{?}{=} t_1, \dots, x_k \stackrel{?}{=} t_k\} \text{ equivalent to } \mathcal{G}$$

all different
 k

$$\{x_1, \dots, x_k\} \cap \bigcup_{i=1}^k \text{vars}(t_i) = \emptyset$$

any such \mathcal{G}' defines a straightforward solution $[x_1 = t_1, \dots, x_k = t_k]$

Notation

$$\mathcal{G} = \{l_1 \stackrel{?}{=} r_1, \dots, l_n \stackrel{?}{=} r_n\}$$

$$\text{vars}(\mathcal{G}) \stackrel{\Delta}{=} \bigcup_{i=1}^n (\text{vars}(l_i) \cup \text{vars}(r_i))$$

$$\mathcal{G}\rho \stackrel{\Delta}{=} \{l_1\rho \stackrel{?}{=} r_1\rho, \dots, l_n\rho \stackrel{?}{=} r_n\rho\}$$

Unification algorithm

delete

$\mathcal{G} \cup \{t \stackrel{?}{=} t\}$
becomes
 \mathcal{G}

eliminate

$\mathcal{G} \cup \{x \stackrel{?}{=} t\}$
becomes if $x \in \text{vars}(\mathcal{G}) \setminus \text{vars}(t)$
 $\mathcal{G}[x = t] \cup \{x \stackrel{?}{=} t\}$

swap

$\mathcal{G} \cup \{f(t_1, \dots, t_m) \stackrel{?}{=} x\}$
becomes
 $\mathcal{G} \cup \{x \stackrel{?}{=} f(t_1, \dots, t_m)\}$

decompose

$\mathcal{G} \cup \{f(t_1, \dots, t_m) \stackrel{?}{=} f(u_1, \dots, u_m)\}$
becomes
 $\mathcal{G} \cup \{t_1 \stackrel{?}{=} u_1, \dots, t_m \stackrel{?}{=} u_m\}$

occur-check

$\mathcal{G} \cup \{x \stackrel{?}{=} f(t_1, \dots, t_m)\}$
fails if $x \in \text{vars}(f(t_1, \dots, t_m))$

conflict

$\mathcal{G} \cup \{f(t_1, \dots, t_m) \stackrel{?}{=} g(u_1, \dots, u_h)\}$
fails if $f \neq g$ or $m \neq h$

Example

$$\{\text{plus}(\text{succ}(x), x) \stackrel{?}{=} \text{plus}(y, 0)\}$$

decompose

$$\mathcal{G} \cup \{f(t_1, \dots, t_m) \stackrel{?}{=} f(u_1, \dots, u_m)\}$$

becomes

$$\mathcal{G} \cup \{t_1 \stackrel{?}{=} u_1, \dots, t_m \stackrel{?}{=} u_m\}$$

$$\{\text{succ}(x) \stackrel{?}{=} y, x \stackrel{?}{=} 0\}$$

eliminate

$$\mathcal{G} \cup \{x \stackrel{?}{=} t\}$$

becomes if $x \in \text{vars}(\mathcal{G}) \setminus \text{vars}(t)$

$$\mathcal{G}[x = t] \cup \{x \stackrel{?}{=} t\}$$

$$\{\text{succ}(0) \stackrel{?}{=} y, x \stackrel{?}{=} 0\}$$

swap

$$\mathcal{G} \cup \{f(t_1, \dots, t_m) \stackrel{?}{=} x\}$$

becomes

$$\mathcal{G} \cup \{x \stackrel{?}{=} f(t_1, \dots, t_m)\}$$

$$\{y \stackrel{?}{=} \text{succ}(0), x \stackrel{?}{=} 0\}$$

success!

$$\rho = [y = \text{succ}(0), x = 0]$$

Example

$\{\text{plus}(0, x) \stackrel{?}{=} \text{succ}(y)\}$ **plus \neq succ**

conflict

$\mathcal{G} \cup \{f(t_1, \dots, t_m) \stackrel{?}{=} g(u_1, \dots, u_h)\}$
fails if $f \neq g$ or $m \neq h$

failure!

Example

$$\{\text{succ}(x) \stackrel{?}{=} y, \text{succ}(y) \stackrel{?}{=} x\}$$

swap

$$\mathcal{G} \cup \{f(t_1, \dots, t_m) \stackrel{?}{=} x\}$$

becomes

$$\mathcal{G} \cup \{x \stackrel{?}{=} f(t_1, \dots, t_m)\}$$

$$\{\text{succ}(x) \stackrel{?}{=} y, x \stackrel{?}{=} \text{succ}(y)\}$$

eliminate

$$\mathcal{G} \cup \{x \stackrel{?}{=} t\}$$

becomes if $x \in \text{vars}(\mathcal{G}) \setminus \text{vars}(t)$

$$\mathcal{G}[x = t] \cup \{x \stackrel{?}{=} t\}$$

$$\{\text{succ}(\text{succ}(y)) \stackrel{?}{=} y, x \stackrel{?}{=} \text{succ}(y)\}$$

swap

$$\mathcal{G} \cup \{f(t_1, \dots, t_m) \stackrel{?}{=} x\}$$

becomes

$$\mathcal{G} \cup \{x \stackrel{?}{=} f(t_1, \dots, t_m)\}$$

$$\{y \stackrel{?}{=} \text{succ}(\text{succ}(y)), x \stackrel{?}{=} \text{succ}(y)\}$$

$y \in \text{vars}(\text{succ}(\text{succ}(y)))$

occur-check

$$\mathcal{G} \cup \{x \stackrel{?}{=} f(t_1, \dots, t_m)\}$$

fails if $x \in \text{vars}(f(t_1, \dots, t_m))$

failure!



Exercise

$$\{\text{plus}(x, \text{succ}(x)) \stackrel{?}{=} \text{plus}(0, y), \text{plus}(y, z) \stackrel{?}{=} \text{plus}(z, w)\}$$

delete

$$\mathcal{G} \cup \{t \stackrel{?}{=} t\}$$

becomes

$$\mathcal{G}$$

eliminate

$$\mathcal{G} \cup \{x \stackrel{?}{=} t\}$$

becomes if $x \in \text{vars}(\mathcal{G}) \setminus \text{vars}(t)$

$$\mathcal{G}[x = t] \cup \{x \stackrel{?}{=} t\}$$

swap

$$\mathcal{G} \cup \{f(t_1, \dots, t_m) \stackrel{?}{=} x\}$$

becomes

$$\mathcal{G} \cup \{x \stackrel{?}{=} f(t_1, \dots, t_m)\}$$

decompose

$$\mathcal{G} \cup \{f(t_1, \dots, t_m) \stackrel{?}{=} f(u_1, \dots, u_m)\}$$

becomes

$$\mathcal{G} \cup \{t_1 \stackrel{?}{=} u_1, \dots, t_m \stackrel{?}{=} u_m\}$$

occur-check

$$\mathcal{G} \cup \{x \stackrel{?}{=} f(t_1, \dots, t_m)\}$$

fails if $x \in \text{vars}(f(t_1, \dots, t_m))$

conflict

$$\mathcal{G} \cup \{f(t_1, \dots, t_m) \stackrel{?}{=} g(u_1, \dots, u_h)\}$$

fails if $f \neq g$ or $m \neq h$