



# Linguaggi di Programmazione

Roberta Gori

## Capitolo 2.1

# Inferenza

# Come applicare le regole SOS

1. un goal

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

# Come applicare le regole SOS

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

2. prendiamo una regola

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

# Come applicare le regole SOS

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

3. unifichiamo  
(se possibile)

$$E_0 = 1 \oplus 2$$

$$E_1 = 3 \oplus 4$$

$$n = m$$

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

# Come applicare le regole SOS

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

4. instanziamo

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

# Come applicare le regole SOS

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

5. ricorsivamente risolviamo  
i sotto-goal

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

# Come applicare le regole SOS

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

6. combiniamo i risultati

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



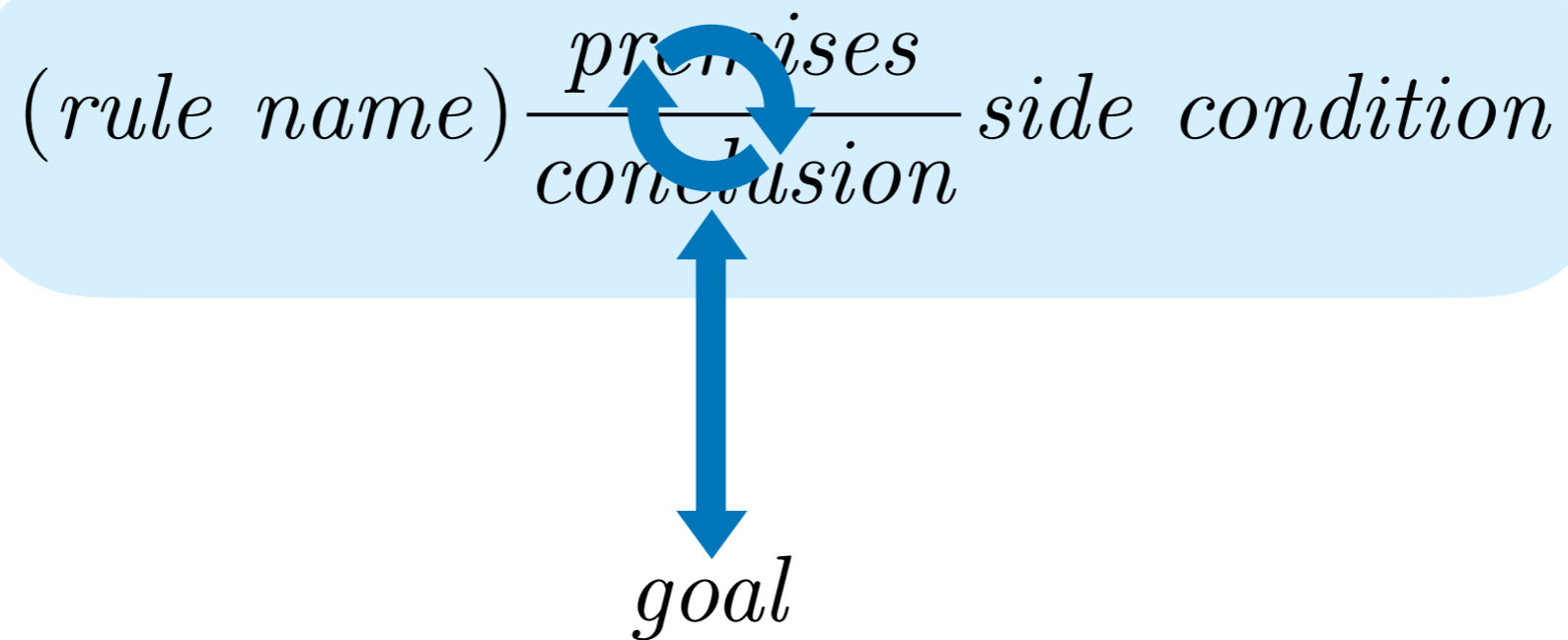
# SOS rule application?

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

7. restituiamo il risultato

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

# Processo di deduzione



Segnature

# Segnatura (senza tipi)

$(\Sigma, ar)$

un insieme di simboli di **funzione**  
(chiamati anche **operatori**)

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

**arieta'** di una funzione

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

ogni simbolo di funzione ha  
un'arieta'

## Esempio

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

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

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

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

# ....o equivalentemente

$$(\Sigma, ar)$$

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

una segnatura è una famiglia di insiemi di operatori indicizzati dall'arietà'

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

# Termini su una segnatura

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

$X = \{x, y, z, \dots\}$  un insieme infinito di variabili

$T_{\Sigma, X}$  denota l'insieme di tutti i **termini** su  $\Sigma, X$

e' il piu' piccolo insieme tale che:

- se  $x \in X$ , allora  $x \in T_{\Sigma, X}$
- se  $c \in \Sigma_0$ , allora  $c \in T_{\Sigma, X}$
- se  $f \in \Sigma_n$  e  $t_1, \dots, t_n \in T_{\Sigma, X}$ , allora  $f(t_1, \dots, t_n) \in T_{\Sigma, X}$

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

# Vars

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

$t \in T_{\Sigma, X}$        $vars(t)$  l'insieme di variabili che appare in  $t$

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

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

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

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

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

# Esempio

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

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

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

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



# Esercizio

completiamo lo schema

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

# Sostituzioni

$\rho : X \rightarrow T_{\Sigma, X}$  una **sostituzione** assegna un termine alle variabili

consideriamo sostituzioni che sono identità ovunque, tranne un numero finito di casi, scritto

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

tutte diverse

indichiamo allo stesso modo la funzione sui termini  $\rho : T_{\Sigma, X} \rightarrow T_{\Sigma, X}$

$\rho(t)$  denota il termine ottenuto dall'applicazione simultanea della sostituzione a tutte le occorrenze delle variabile in  $t$

$t\rho$  notazione alternativa

# Esempio

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

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

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

# la relazione mgt (more general than)

$t$  è più generale di  $t'$  se  $\exists \rho. t' = t\rho$

nel qual caso diciamo anche che  $t'$  è un istanza di  $t$

$\text{plus}(x, \text{succ}(y))$	mgt	$\text{plus}(0, \text{succ}(\text{succ}(z)))$
$\text{plus}(0, x)$	<del>mgt</del>	$\text{plus}(y, 0)$
$\text{plus}(y, 0)$	<del>mgt</del>	$\text{plus}(0, x)$
$\text{plus}(0, x)$ $\text{plus}(y, 0)$	mgt	$\text{plus}(0, 0)$

# relazione mgt

mgt e' transitiva e riflessiva

$$\frac{}{t \text{ mgt } t}$$

se  $(t_1 \text{ mgt } t_2)$  e  $(t_2 \text{ mgt } t_3)$  allora  $(t_1 \text{ mgt } t_3)$

ci sono termini  $t \neq t'$  tali che  $(t \text{ mgt } t')$  e  $(t' \text{ mgt } t)$

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

se consideriamo le classi di equivalenza rispetto al renaming sostituzioni che mappano variabili in variabili  
 $\rho = [x = y]$  otteniamo un ordine sui termini

mgt puo' essere esteso alle sostituzioni

e sulle sostituzioni

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

# Problema dell'unificazione

Il problema di unificazione nella sua forma più semplice (sintattica, del primo ordine) può essere espresso come segue:

Dato un insieme di possibili eguaglianze

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

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

Possiamo trovare una sostituzione  $\rho$

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

chiamiamo tale  $\rho$  una soluzione di  $\mathcal{G}$

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

# Problema dell'unificazione

In realta' possiamo risolvere il seguente problema

Dato un insieme di possibili eguaglianze

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

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

Possiamo trovare la sostituzione piu' generale  $\rho$  ?

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

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

# Algoritmo di unificazione

**Idea:** riduciamo iterativamente l'insieme  $\mathcal{G}$  applicando delle trasformazioni che preservano la soluzione finche' non **troviamo una soluzione** oppure possiamo **provare che tale soluzione non esiste**



Le soluzioni potrebbero non esistere  
e anche se esistono possono non essere uniche

Se esistono saranno uniche a meno di renaming!



# Terminazione

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

$\mathcal{G}$  e  $\mathcal{G}'$  sono equivalenti if  $sols(\mathcal{G}) = sols(\mathcal{G}')$

l'algoritmo termina con successo quando troviamo

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

tutte diverse

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

ogni siffatto  $\mathcal{G}'$  costituisce una soluzione  $[x = t_1, \dots, x_k = t_k]$

# Notazione

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

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

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

# Algoritmo di unificazione

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  $m \neq h$

# Esempio

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

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

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

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

**successo!**

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

# Esempio

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

**fallimento!**

# Esempio

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

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

fallimento

# Esercizio

delete

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

decompose

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

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

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  $m \neq h$

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

