Basic Approaches to the Semantics of Computation (BaSC)

Lecture 2: Inference and Unification

Jorge A. Pérez

Bernoulli Institute for Mathematics, Computer Science, and AI
University of Groningen, Groningen, the Netherlands

November 13, 2025

$$\frac{E_0 \longrightarrow n_0 \quad E_1 \longrightarrow n_1}{E_0 \otimes E_1 \longrightarrow n}\,(prod)\; n = n_0 \cdot n_1 \quad \overset{?}{\rightsquigarrow} \quad \frac{1 \oplus 2 \longrightarrow 3 \quad 3 \oplus 4 \longrightarrow 7}{(1 \oplus 2) \otimes (3 \oplus 4) \longrightarrow 21}\,(prod)$$

Inference Rule

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

$$\overset{?}{\rightsquigarrow}$$

Rule Instance

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

Inference Rule

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

$$\overset{?}{\rightsquigarrow}$$

Rule Instance

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

Today:

- Goal-oriented (or bottom-up) derivations
- Signatures and substitutions
- Unification (key ideas)
- Inference rules, derivations, an inline notation
- Logic programming (key ideas)

Applying SOS Rules

Step 1. A goal

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

Applying SOS Rules

Step 2. Take a rule

(prod)

$$\frac{E_0 \longrightarrow n_0 \qquad E_1 \longrightarrow n_1}{E_0 \otimes E_1 \longrightarrow n} \; n = n_0 \cdot n_1$$

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

Applying SOS Rules

Step 3. Unify (if possible)

(*prod*)

$$\frac{E_0 \longrightarrow n_0 \qquad E_1 \longrightarrow n_1}{E_0 \otimes E_1 \longrightarrow n} \; n = n_0 \cdot n_1$$

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

$$\begin{aligned} E_0 &= 1 \oplus 2 \\ E_1 &= 3 \oplus 4 \\ n &= m \end{aligned}$$

Applying SOS Rules

Step 4. Instantiate

(*prod*)

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

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

Applying SOS Rules

Step 5. Recursively solve sub-goals

(*prod*)

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

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

Applying SOS Rules

Step 6. Combine results

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

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

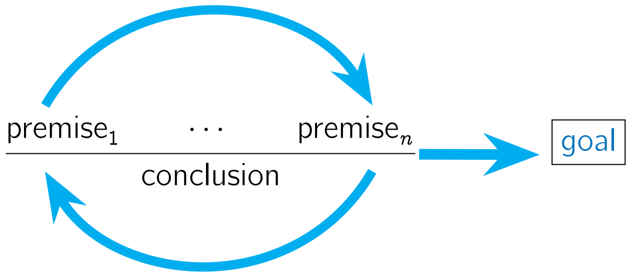
Applying SOS Rules

Step 7. Return results

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

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

Deduction Process: Goal-Oriented


$\dfrac{\text{premise}_1 \quad \cdots \quad \text{premise}_n}{\text{conclusion}}$
goal

Signatures

Unsorted Signature

(Σ, ar)

a set of function symbols
(also called operators)

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

arity function

$$\mathsf{ar} : \Sigma \to \mathbb{N}$$

Each symbol has an arity. Examples:

- $\mathsf{ar}(c) = 0$: constant
- $\mathsf{ar}(f) = 1$: unary
- $\mathsf{ar}(g) = 2$: binary
- $\mathsf{ar}(h) = 3$: ternary

Equivalently

- Given (Σ, ar), we can define

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

  That is, Σ_n denotes the set of operators of arity n

- A signature can then be defined as a family of sets of operators, indexed by their arity:

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

Terms Over a Signature

Consider given:

$\Sigma = \{\Sigma_n\}_{n \in \mathbb{N}}$	a signature
$X = \{x, y, z, \ldots\}$	an infinite set of variables

Let $T_{\Sigma,X}$ denote the set of all terms over Σ, X.

Terms Over a Signature

Consider given:

$\Sigma = \{\Sigma_n\}_{n\in\mathbb{N}}$	a signature
$X = \{x, y, z, \dots\}$	an infinite set of variables

Let $T_{\Sigma,X}$ denote the set of all terms over Σ, X. 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 $f(t_1, \dots, t_n) \in T_{\Sigma,X}$

Terms Over a Signature

Consider given:

$\Sigma = \{\Sigma_n\}_{n\in\mathbb{N}}$	a signature
$X = \{x, y, z, \dots\}$	an infinite set of variables

Let $T_{\Sigma,X}$ denote the set of all terms over Σ, X. 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 $f(t_1, \dots, t_n) \in T_{\Sigma,X}$

Put differently:

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

Variables

Assume

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

Given $t \in T_{\Sigma,X}$, we write vars(t) to denote the set of variables that appear in t.

$$\text{vars} \;:\; T_{\Sigma,X} \to \mathcal{P}(X)$$

$$\begin{aligned} \text{vars}(x) &\triangleq \{x\} \\ \text{vars}(c) &\triangleq \emptyset \\ \text{vars}(f(t_1,\ldots,t_n)) &\triangleq \bigcup_{i=1}^{n} \text{vars}(t_i) \end{aligned}$$

Variables

Assume

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

Given $t \in T_{\Sigma,X}$, we write $\mathsf{vars}(t)$ to denote the set of variables that appear in t.

$$\mathsf{vars} : T_{\Sigma,X} \to \mathcal{P}(X)$$

$$\begin{aligned} \mathsf{vars}(x) &\triangleq \{x\} \\ \mathsf{vars}(c) &\triangleq \emptyset \\ \mathsf{vars}(f(t_1, \ldots, t_n)) &\triangleq \bigcup_{i=1}^{n} \mathsf{vars}(t_i) \end{aligned}$$

We can also define closed terms:

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

Complete the Schema (1/2)

A signature:

$\Sigma_0 = \{0\}$

$\Sigma_1 = \{\mathsf{succ}\}$

$\Sigma_2 = \{\mathsf{plus}\}$

$\Sigma_n = \emptyset \qquad (\text{if } n > 2)$

t	$t \in ?$		$\mathsf{vars}(t)$
0	☒ T_Σ	☐ $T_{\Sigma,X}$	$\emptyset$
x	☐ T_Σ	☒ $T_{\Sigma,X}$	$\{x\}$
$\mathsf{succ}(0)$	☐ T_Σ	☐ $T_{\Sigma,X}$	
$\mathsf{succ}(x)$	☐ T_Σ	☐ $T_{\Sigma,X}$	
$\mathsf{succ}(\mathsf{plus}(0), x)$	☐ T_Σ	☐ $T_{\Sigma,X}$	

Complete the Schema (2/2)

t	$t \in ?$	$\mathsf{vars}(t)$
$\mathsf{plus}(\mathsf{succ}(0), x)$	☐ T_Σ ☐ $T_{\Sigma,X}$	
$\mathsf{succ}(\mathsf{succ}(0), \mathsf{plus}(x))$	☐ T_Σ ☐ $T_{\Sigma,X}$	
$\mathsf{succ}(\mathsf{plus}(w, z))$	☐ T_Σ ☐ $T_{\Sigma,X}$	
$\mathsf{plus}(\mathsf{plus}(x, \mathsf{succ}(y)), \mathsf{plus}(0, \mathsf{succ}(x)))$	☐ T_Σ ☐ $T_{\Sigma,X}$	

Substitutions

- A substitution ρ assigns terms to variables: $\rho : X \to T_{\Sigma,X}$
- We only consider substitutions that are identity everywhere, except for a finite number of cases, written:

$$\rho = [\underbrace{x_1}_{} = t_1, \cdots, \underbrace{x_n}_{\text{all different}} = t_n] \qquad \rho(x) = \begin{cases} t_i & \text{if } x = x_i, \\ x & \text{otherwise.} \end{cases}$$

Substitutions

- A substitution ρ assigns terms to variables: $\rho : X \to T_{\Sigma,X}$
- We only consider substitutions that are identity everywhere, except for a finite number of cases, written:

$$\rho = [\, x_1 = t_1, \cdots, x_n = t_n]$$

all different

$$\rho(x) = \begin{cases} t_i & \text{if } x = x_i, \\ x & \text{otherwise.} \end{cases}$$

- Overloaded notation for the lifted function $\rho : T_{\Sigma,X} \to T_{\Sigma,X}$
- $\rho(t)$ denotes the term obtained by simultaneous application of ρ to all variable occurrences in t. An alternative notation: $t\rho$.

Example

Given

$$\rho \triangleq \left[x = \mathsf{succ}(y) \,,\; y = 0 \right]$$
$$t \triangleq \mathsf{plus}(\mathsf{plus}(x, y), \mathsf{succ}(x))$$

Then

$$t\rho =$$

Example

Given

$$\rho \triangleq \left[x = \mathsf{succ}(y)\,,\ y = 0 \right]$$
$$t \triangleq \mathsf{plus}(\mathsf{plus}(x, y), \mathsf{succ}(x))$$

Then

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

More General Than (mgt) Relation

- We say that t is more general than t', written $t \text{ mgt } t'$, if $\exists\rho.\ t' = t\rho$.
- If $t \text{ mgt } t'$ then t' is an instance of t

Example

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

More General Than (mgt) Relation

- We say that t is more general than t', written t mgt t', if $\exists\rho.\ t' = t\rho$.
- If t mgt t' then t' is an instance of t

Example

	mgt?	
$\mathsf{plus}(x, \mathsf{succ}(y))$	✓	$\mathsf{plus}(0, \mathsf{succ}(\mathsf{succ}(z)))$
$\mathsf{plus}(0, x)$		$\mathsf{plus}(y, 0)$
$\mathsf{plus}(y, 0)$		$\mathsf{plus}(0, x)$
$\mathsf{plus}(0, x),\ \mathsf{plus}(y, 0)$		$\mathsf{plus}(0, 0)$

More General Than (mgt) Relation

- We say that t is more general than t', written t mgt t', if $\exists \rho.\ t' = t\rho$.
- If t mgt t' then t' is an instance of t

Example

	mgt?	
$\mathrm{plus}(x, \mathrm{succ}(y))$	✓	$\mathrm{plus}(0, \mathrm{succ}(\mathrm{succ}(z)))$
$\mathrm{plus}(0, x)$	✗	$\mathrm{plus}(y, 0)$
$\mathrm{plus}(y, 0)$		$\mathrm{plus}(0, x)$
$\mathrm{plus}(0, x)$, $\mathrm{plus}(y, 0)$		$\mathrm{plus}(0, 0)$

More General Than (mgt) Relation

- We say that t is more general than t', written t mgt t', if $\exists\rho.\ t' = t\rho$.
- If t mgt t' then t' is an instance of t

Example

	mgt?	
$\mathsf{plus}(x, \mathsf{succ}(y))$	✓	$\mathsf{plus}(0, \mathsf{succ}(\mathsf{succ}(z)))$
$\mathsf{plus}(0, x)$	✗	$\mathsf{plus}(y, 0)$
$\mathsf{plus}(y, 0)$	✗	$\mathsf{plus}(0, x)$
$\mathsf{plus}(0, x)$, $\mathsf{plus}(y, 0)$		$\mathsf{plus}(0, 0)$

More General Than (mgt) Relation

- We say that t is more general than t', written t mgt t', if $\exists \rho.\ t' = t\rho$.
- If t mgt t' then t' is an instance of t

Example

	mgt?	
$\mathrm{plus}(x, \mathrm{succ}(y))$	✓	$\mathrm{plus}(0, \mathrm{succ}(\mathrm{succ}(z)))$
$\mathrm{plus}(0, x)$	✗	$\mathrm{plus}(y, 0)$
$\mathrm{plus}(y, 0)$	✗	$\mathrm{plus}(0, x)$
$\mathrm{plus}(0, x)$, $\mathrm{plus}(y, 0)$	✓	$\mathrm{plus}(0, 0)$

mgt Relation

- mgt is transitive and reflexive
  - t mgt t
  - if (t_1 mgt t_2) and (t_3 mgt t_3) then (t_1 mgt t_3)
- There are terms $t \neq t'$ such that (t mgt t') and (t' mgt t). Example:

$$\mathsf{succ}(x) \qquad \mathsf{succ}(y)$$

- mgt extends to substitutions pointwise:

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

The Unification Problem

The unification problem (syntactic, first-order) can be stated as follows:

Given a set of *potential* equalities

$$\mathcal{G} = \{\, l_1 \overset{?}{=} r_1, \ldots, l_n \overset{?}{=} r_n \,\}$$

where $l_1, \ldots, l_n, r_1, \ldots, r_n \;\in\; T_{\Sigma,X}$.

Can we find a ρ such that $\forall i \in [1, n].\ \rho(l_i) = \rho(r_i)$?

The Unification Problem

The unification problem (syntactic, first-order) can be stated as follows:

Given a set of *potential* equalities

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

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

Can we find a ρ such that $\forall i \in [1, n].\ \rho(l_i) = \rho(r_i)$?

The set of solutions of $\mathcal{G}$: $\mathsf{sols}(\mathcal{G}) \triangleq \{ \rho \mid \forall i \in [1, n].\ \rho(l_i) = \rho(r_i) \}$.

The Unification Problem

The unification problem (syntactic, first-order) can be stated as follows:

Given a set of *potential* equalities

$$\mathcal{G} = \{\, l_1 \overset{?}{=} r_1, \ldots, l_n \overset{?}{=} r_n \,\}$$

where $l_1, \ldots, l_n, r_1, \ldots, r_n \;\in\; T_{\Sigma,X}$.

Can we find a ρ such that $\forall i \in [1,n].\ \rho(l_i) = \rho(r_i)$?

The set of solutions of $\mathcal{G}$: $\mathsf{sols}(\mathcal{G}) \triangleq \{\, \rho \mid \forall i \in [1,n].\ \rho(l_i) = \rho(r_i) \,\}$.

Another problem is finding the most general unifier for $\mathcal{G}$:
can we find a $\rho \in \mathsf{sols}(\mathcal{G})$ such that ρ mgt ρ' for every $\rho' \in \mathsf{sols}(\mathcal{G})$?

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.

Note A solution may no exist and even if exists it may not be unique.

Algorithm: Termination Criteria

- We say $\mathcal{G}$ and $\mathcal{G}'$ are equivalent if $\mathsf{sols}(\mathcal{G}) = \mathsf{sols}(\mathcal{G}')$.
- Given $\mathcal{G} = \{\, l_1 \overset{?}{=} r_1, \ldots, l_n \overset{?}{=} r_n \,\}$, the algorithm terminates successfully when we reach:

$\mathcal{G}' = \{\, x_1 \overset{?}{=} t_1, \ldots, x_k \overset{?}{=} t_k \,\}$ such that
- $\mathcal{G}'$ is equivalent to $\mathcal{G}$
- $\underbrace{\{x_1, \ldots, x_k\}}_{\text{all different}} \cap \bigcup_{i=1}^{k} \mathsf{vars}(t_i) = \varnothing$

- Any such $\mathcal{G}'$ determines a solution $[x_1 = t_1, \ldots, x_k = t_k]$.

Algorithm: Notation

Suppose $\mathcal{G} = \{\, l_1 \overset{?}{=} r_1, \ldots, l_n \overset{?}{=} r_n \,\}$. We define:

$$\begin{aligned} \mathsf{vars}(\mathcal{G}) &\triangleq \bigcup_{i=1}^{n} \big(\mathsf{vars}(l_i) \cup \mathsf{vars}(r_i)\big) \\ \mathcal{G}\rho &\triangleq \{\, l_1\rho \overset{?}{=} r_1\rho, \cdots, l_n\rho \overset{?}{=} r_n\rho \,\} \end{aligned}$$

Unification Algorithm

delete

$$\mathcal{G} \cup \{\, t \overset{?}{=} t \,\}$$

<u>becomes</u>

$$\mathcal{G}$$

eliminate

$$\mathcal{G} \cup \{\, x \overset{?}{=} t \,\}$$

<u>becomes</u>

$$\mathcal{G}[x = t] \cup \{\, x \overset{?}{=} t \,\} \quad \text{if } x \in \mathsf{vars}(\mathcal{G}) \setminus \mathsf{vars}(t)$$

Unification Algorithm

delete

$$\mathcal{G} \cup \{\, t \overset{?}{=} t \,\}$$

<u>becomes</u>

$$\mathcal{G}$$

eliminate

$$\mathcal{G} \cup \{\, x \overset{?}{=} t \,\}$$

<u>becomes</u>

$$\mathcal{G}[x = t] \cup \{\, x \overset{?}{=} t \,\} \quad \text{if } x \in \mathsf{vars}(\mathcal{G}) \setminus \mathsf{vars}(t)$$

swap

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

<u>becomes</u>

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

decompose

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

<u>becomes</u>

$$\mathcal{G} \cup \{\, t_1 \overset{?}{=} u_1, \ldots, u_1 \overset{?}{=} u_m \,\}$$

Unification Algorithm

delete

$$\mathcal{G} \cup \{\, t \overset{?}{=} t \,\}$$

<u>becomes</u>

$$\mathcal{G}$$

eliminate

$$\mathcal{G} \cup \{\, x \overset{?}{=} t \,\}$$

<u>becomes</u>

$$\mathcal{G}[x = t] \cup \{\, x \overset{?}{=} t \,\} \quad \text{if } x \in \mathsf{vars}(\mathcal{G}) \setminus \mathsf{vars}(t)$$

swap

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

<u>becomes</u>

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

decompose

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

<u>becomes</u>

$$\mathcal{G} \cup \{\, t_1 \overset{?}{=} u_1, \ldots, u_1 \overset{?}{=} u_m \,\}$$

occur-check

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

<u>fails</u> if $x \in \mathsf{vars}(f(t_1, \ldots, t_m))$

conflict

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

<u>fails</u> if $f \neq g$ or $m \neq h$

Unification Algorithm: Example

$$\{\mathsf{plus}(\mathsf{succ}(x), x) \overset{?}{=} \mathsf{plus}(y, \mathsf{0})\}$$

Unification Algorithm: Example

$$\{\mathsf{plus}(\mathsf{succ}(x), x) \overset{?}{=} \mathsf{plus}(y, 0)\}$$

decompose

$$\{\mathsf{succ}(x) \overset{?}{=} y, x \overset{?}{=} 0\}$$

Unification Algorithm: Example

$$\{\mathsf{plus}(\mathsf{succ}(x), x) \overset{?}{=} \mathsf{plus}(y, 0)\}$$

decompose

$$\{\mathsf{succ}(x) \overset{?}{=} y, x \overset{?}{=} 0\}$$

eliminate

$$\{\mathsf{succ}(0) \overset{?}{=} y, x \overset{?}{=} 0\}$$

Unification Algorithm: Example

$$\{\mathsf{plus}(\mathsf{succ}(x), x) \overset{?}{=} \mathsf{plus}(y, 0)\}$$

decompose

$$\{\mathsf{succ}(x) \overset{?}{=} y, x \overset{?}{=} 0\}$$

eliminate

$$\{\mathsf{succ}(0) \overset{?}{=} y, x \overset{?}{=} 0\}$$

swap

$$\{y \overset{?}{=} \mathsf{succ}(0), x \overset{?}{=} 0\}$$

Unification Algorithm: Example

$$\{\mathsf{plus}(\mathsf{succ}(x), x) \overset{?}{=} \mathsf{plus}(y, 0)\}$$

decompose

$$\{\mathsf{succ}(x) \overset{?}{=} y, x \overset{?}{=} 0\}$$

eliminate

$$\{\mathsf{succ}(0) \overset{?}{=} y, x \overset{?}{=} 0\}$$

swap

$$\{y \overset{?}{=} \mathsf{succ}(0), x \overset{?}{=} 0\}$$

✓ **success**: $\rho = [y = \mathsf{succ}(0), x = 0]$

Unification Algorithm: Example

$$\{\mathsf{plus}(0, x) \overset{?}{=} \mathsf{succ}(y)\}$$

Unification Algorithm: Example

$$\{\mathsf{plus}(0, x) \overset{?}{=} \mathsf{succ}(y)\}$$

conflict: $\mathsf{plus} \neq \mathsf{succ}$

Unification Algorithm: Example

$$\{\mathsf{plus}(0, x) \overset{?}{=} \mathsf{succ}(y)\}$$

conflict: $\mathsf{plus} \neq \mathsf{succ}$

✗ fail

Unification: Another Example

$$\{\mathsf{succ}(x) \overset{?}{=} y, \mathsf{succ}(y) \overset{?}{=} x\}$$

Unification: Another Example

$$\{\mathsf{succ}(x) \overset{?}{=} y, \mathsf{succ}(y) \overset{?}{=} x\}$$

swap

$$\{\mathsf{succ}(x) \overset{?}{=} y, x \overset{?}{=} \mathsf{succ}(y)\}$$

Unification: Another Example

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

swap

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

eliminate

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

Unification: Another Example

$$\{\mathsf{succ}(x) \overset{?}{=} y, \mathsf{succ}(y) \overset{?}{=} x\}$$

swap

$$\{\mathsf{succ}(x) \overset{?}{=} y, x \overset{?}{=} \mathsf{succ}(y)\}$$

eliminate

$$\{\mathsf{succ}(\mathsf{succ}(y)) \overset{?}{=} y, x \overset{?}{=} \mathsf{succ}(y)\}$$

swap

$$\{y \overset{?}{=} \mathsf{succ}(\mathsf{succ}(y)), x \overset{?}{=} \mathsf{succ}(y)\}$$

Unification: Another Example

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

swap

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

eliminate

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

swap

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

occur-check: $y \in \mathsf{vars}(\mathsf{succ}(\mathsf{succ}(y)))$

✗ fail

Exercise

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

Inference Rules

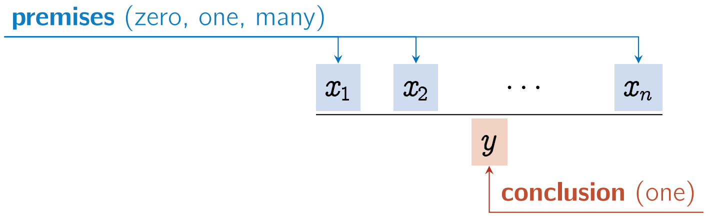
Inference Rules

An inference rule:

$$\frac{x_1 \qquad x_2 \qquad \cdots \qquad x_n}{y}$$

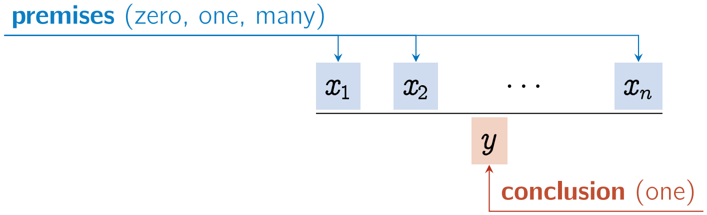
Inference Rules

An inference rule:



Inference Rules

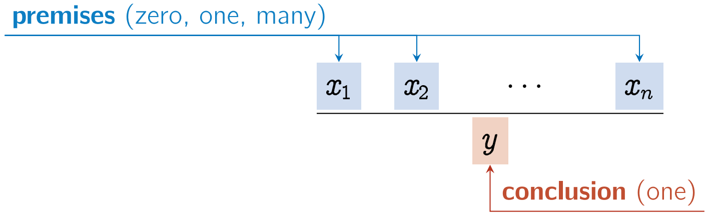
An inference rule:



- Intuition: If the premises are valid, then the conclusion is also valid.
- In an axiom there are no premises: the conclusion is always valid (it is a fact)
- Above, $x_1, x_2, \ldots, y$ are formulas. Any variable in them is universally quantified (implicitly).

Inference Rules

An inference rule:



- Intuition: If the premises are valid, then the conclusion is also valid.
- In an axiom there are no premises: the conclusion is always valid (it is a fact)
- Above, $x_1, x_2, \ldots, y$ are formulas. Any variable in them is universally quantified (implicitly).
- A rule instance is obtained by applying some substitution ρ to $x_1, x_2, \ldots, y$.

Rule Instances

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

Two instances of (prod):

$$(\mathit{prod})\ \frac{1 \longrightarrow 1 \quad 1 \oplus 2 \longrightarrow 3}{1 \otimes (1 \oplus 2) \longrightarrow 3}\ 3 = 1 \cdot 3$$

$$(\mathit{prod})\ \frac{1 \longrightarrow 3 \quad 1 \oplus 2 \longrightarrow 5}{1 \otimes (1 \oplus 2) \longrightarrow 3}\ 15 = 3 \cdot 5$$

Not all instances are valid!

Rule Instances

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

Another instance:

$$(\textit{prod})\ \frac{E \otimes 2 \longrightarrow k \qquad E \oplus 1 \longrightarrow 3}{(E \otimes 2) \otimes (E \oplus 1) \longrightarrow 3k}$$

Rule Instances

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

Another instance:

variables can be shared

$$(\mathit{prod})\ \frac{\boxed{E} \otimes 2 \longrightarrow k \qquad \boxed{E} \oplus 1 \longrightarrow 3}{(\boxed{E} \otimes 2) \otimes (\boxed{E} \oplus 1) \longrightarrow 3k}$$

variables can be shared

Logical System

A logical system is a set of axioms and inference rules:

$$R = \left\{ \frac{}{z}, \frac{x_1 \quad \cdots \quad x_n}{y}, \ldots \right\}$$

If an inference rule contains some variables, we assume all its instances are in R

Derivations

Given a logical system R, a derivation in R, is written

$$d \Vdash_R y$$

where

- either $d = \left(\dfrac{}{y}\right)$ is an axiom of R;
- or $d = \left(\dfrac{d_1 \quad \cdots \quad d_n}{y}\right)$ for some derivations $d_1 \Vdash_R x_1, \cdots, d_n \Vdash_R x_n$

  such that $\left(\dfrac{x_1 \quad \cdots \quad x_n}{y}\right)$ is an inference rule of R.

Put differently: a derivation is a proof tree whose leaves are axioms.

Example

$$R = \left\{ \frac{}{N \longrightarrow n}, \quad \frac{E_0 \longrightarrow n_0 \quad E_1 \longrightarrow n_1}{E_0 \oplus E_1 \longrightarrow n_0 + n_1}, \quad \frac{E_0 \longrightarrow n_0 \quad E_1 \longrightarrow n_1}{E_0 \otimes E_1 \longrightarrow n_0 \cdot n_1} \right\}$$

$$d = \cfrac{\cfrac{\cfrac{}{1 \longrightarrow 1} \qquad \cfrac{}{2 \longrightarrow 2}}{(1 \oplus 2) \longrightarrow 3} \qquad \cfrac{\cfrac{}{3 \longrightarrow 3} \qquad \cfrac{}{4 \longrightarrow 4}}{(3 \oplus 4) \longrightarrow 7}}{(1 \oplus 2) \otimes (3 \oplus 4) \longrightarrow 21}$$

Hence

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

Theorems

- Given a logical system R, a theorem of R is written

$$\Vdash_R y$$

  That is, y is a formula for which we can find a derivation d in R.
- The set of all theorems of R is denoted by I_R:

$$I_R \triangleq \{y \mid \Vdash_R y\}$$

An Inline Notation for Derivations

$$d = \dfrac{\dfrac{\dfrac{}{1 \longrightarrow 1} \qquad \dfrac{}{2 \longrightarrow 2}}{(1 \oplus 2) \longrightarrow 3} \qquad \dfrac{\dfrac{}{3 \longrightarrow 3} \qquad \dfrac{}{4 \longrightarrow 4}}{(3 \oplus 4) \longrightarrow 7}}{(1 \oplus 2) \otimes (3 \oplus 4) \longrightarrow 21}$$

$$\begin{aligned}
(1 \oplus 2) \otimes (3 \oplus 4) \longrightarrow 21 &\nwarrow (1 \oplus 2) \longrightarrow 3, (3 \oplus 4) \longrightarrow 7 \\
&\nwarrow 1 \longrightarrow 1, 2 \longrightarrow 2, (3 \oplus 4) \longrightarrow 7 \\
&\nwarrow 2 \longrightarrow 2, (3 \oplus 4) \longrightarrow 7 \\
&\nwarrow (3 \oplus 4) \longrightarrow 7 \\
&\nwarrow 3 \longrightarrow 3, 4 \longrightarrow 4 \\
&\nwarrow 4 \longrightarrow 4 \\
&\nwarrow \square \qquad \text{[the empty goal: nothing left to prove]}
\end{aligned}$$

Backtracking

A goal oriented derivation (depth-first):

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

$$\nwarrow \ (1 \oplus 2) \longrightarrow 7, (3 \oplus 4) \longrightarrow 3$$

$$\nwarrow \ 1 \longrightarrow 1, 2 \longrightarrow 6, (3 \oplus 4) \longrightarrow 3$$

$$\nwarrow \ 2 \longrightarrow 6, (3 \oplus 4) \longrightarrow 3$$

fail! need to backtrack to the last choice and retry

$$\nwarrow \ 1 \longrightarrow 2, 2 \longrightarrow 5, (3 \oplus 4) \longrightarrow 3$$

fail! need to backtrack to the last choice and retry

...

Logic Programming

PROLOG

- PROLOG (‘PROgrammation en LOGique’) is a simple, yet powerful declarative programming language, based on first-order predicate logic
- Key ideas:

Algorithm =	Logic	+	Control
	What (problem description)		How (steps to reach a solution)
	Horn clauses		Resolution
	Database		Interpreter

Formulas

- Base sets:

$X = \{x, y, \ldots\}$	a set of variables
$\Sigma = \{\Sigma_n\}_{n \in \mathbb{N}}$	a signature of function symbols $c, f, g, \ldots$
$\Pi = \{\Pi_n\}_{n \in \mathbb{N}}$	a signature of predicate symbols $p, q, \ldots$

- Atomic formula

$$a = p(t_1, \ldots, t_n)$$

  where $p \in \Pi_n$ and $t_1, \ldots t_n \in T_{\Sigma,X}$

- A formula

$$a_1, \ldots, a_n$$

  is a possibly empty conjunction of atomic formulas

Logic Programs

A logic program serves to answer the question: given a formula g that we want to prove (a goal), what are the valid instances of g?

Logic Programs

A logic program serves to answer the question: given a formula g that we want to prove (a goal), what are the valid instances of g? Example:

```
% Rules
grandparent(X, Y) :- parent(X, Z), parent(Z, Y).

% Facts
parent(jan, merel).
parent(merel, sandra).

:- initialization(main).
main :-
    grandparent(X, sandra),
    write('Sandras grandparent: '), write(X), nl,  halt.
```

Logic Programs, Formally

- A Horn clause:

$$\underbrace{l}_{\text{an atomic formula: the } \textbf{head}} \;{:}\!-\; \overbrace{r}^{\text{a formula: the } \textbf{body}}$$

- Having $a :- a_1, \ldots, a_n$ is analogous to $\dfrac{a_1 \quad \cdots \quad a_n}{a}$.
- A logic program is a set / list of Horn clauses:

$$L = \left\{ \begin{array}{c} \ldots \\ l :- r \\ \ldots \end{array} \right\}$$

SLD Resolution

Idea Iteratively reduce the initial goal g by applying one of the Horn clauses in L to one of the atomic formulas in the goal

Each application

- computes a most general unifier (mgu)
- replaces the selected formula with the body of the selected clause and
- applies the mgu to the new goal

$$
\begin{array}{lll}
?\!-g & \nwarrow_{\sigma_1} & g_1 \\
 & \nwarrow_{\sigma_2} & g_2 \\
 & \nwarrow_{\sigma_3} & \cdots \\
 & \nwarrow_{\sigma_m} & \square
\end{array}
$$

Then $g\sigma_1\sigma_2\cdots\sigma_m$ is a theorem.

SLD Resolution

Assume given:

$?{-}a_1, \cdots, a_i, \cdots, a_k$ $\qquad$ $L = \{\cdots, h \mathrel{:-} b_1, \ldots, b_n, \cdots\}$

Repeat until no goal is left:

1. Select a clause of the goal a_i (e.g. the first)

SLD Resolution

Assume given:

$?- a_1, \cdots, a_i, \cdots, a_k$ $\qquad$ $L = \{\cdots, h :- b_1, \ldots, b_n, \cdots\}$

Repeat until no goal is left:

1. Select a clause of the goal a_i (e.g. the first)
2. Select a Horn clause $h :- b_1, \ldots, b_n$ from L whose head unifies with a_i

SLD Resolution

Assume given:

$?- a_1, \cdots, a_i, \cdots, a_k$ $\qquad L = \{\cdots, h :- b_1, \ldots, b_n, \cdots\}$

Repeat until no goal is left:

1. Select a clause of the goal a_i (e.g. the first)
2. Select a Horn clause $h :- b_1, \ldots, b_n$ from L whose head unifies with a_i
3. Let σ be a most general unifier $(a_i\sigma = h\sigma)$

SLD Resolution

Assume given:

$?\!-a_1, \cdots, a_i, \cdots, a_k$ $\qquad$ $L = \{\cdots, h \mathrel{:-} b_1, \ldots, b_n, \cdots\}$

Repeat until no goal is left:

1. Select a clause of the goal a_i (e.g. the first)
2. Select a Horn clause $h \mathrel{:-} b_1, \ldots, b_n$ from L whose head unifies with a_i
3. Let σ be a most general unifier $(a_i\sigma = h\sigma)$
4. Replace a_i with $b_1, \ldots, b_n$

SLD Resolution

Assume given:

$?\!-a_1, \cdots, a_i, \cdots, a_k$ $\qquad$ $L = \{\cdots, h :\!- b_1, \ldots, b_n, \cdots\}$

Repeat until no goal is left:

1. Select a clause of the goal a_i (e.g. the first)
2. Select a Horn clause $h :\!- b_1, \ldots, b_n$ from L whose head unifies with a_i
3. Let σ be a most general unifier $(a_i\sigma = h\sigma)$
4. Replace a_i with $b_1, \ldots, b_n$
5. Apply the substitution σ to the revised goal $(a_1, \cdots, b_1, \ldots, b_n, \cdots, a_k)\sigma$.
   This ensures that σ is propagated uniformly, as goals may share variables.

SLD Resolution: Variables Matter

Note The same clause can be reused many times: each time its variables must be renamed (before unification) with fresh identifiers, to avoid clashes.

Repeat until no goal is left:

1. Select a clause of the goal a_i (e.g. the first)
2. Select a Horn clause $h \mathrel{:-} b_1, \ldots, b_n$ from M whose head unifies with a_i

SLD Resolution: Variables Matter

Note The same clause can be reused many times: each time its variables must be renamed (before unification) with fresh identifiers, to avoid clashes.

Repeat until no goal is left:

1. Select a clause of the goal a_i (e.g. the first)
2. Select a Horn clause $h \coloneq b_1, \ldots, b_n$ from M whose head unifies with a_i
3. Let $\rho : X \to X$ be a renaming of vars$(h \coloneq b_1, \ldots, b_n)$ to fresh variables

   The renamed clause $(h \coloneq b_1, \ldots, b_n)\rho$ is a variant of the original one

SLD Resolution: Variables Matter

Note The same clause can be reused many times: each time its variables must be renamed (before unification) with fresh identifiers, to avoid clashes.

Repeat until no goal is left:

1. Select a clause of the goal a_i (e.g. the first)
2. Select a Horn clause $h :- b_1, \ldots, b_n$ from M whose head unifies with a_i
3. Let $\rho : X \to X$ be a renaming of vars$(h :- b_1, \ldots, b_n)$ to fresh variables
   The renamed clause $(h :- b_1, \ldots, b_n)\rho$ is a variant of the original one
4. Let σ be a most general unifier $(a_i\sigma = (h\rho)\sigma)$
5. Replace a_i with $(b_1, \ldots, b_n)\rho$
6. Apply σ to the revised goal $(a_1, \cdots, (b_1, \ldots, b_n)\rho, \cdots, a_k)\sigma$.
   This ensures that σ is propagated uniformly, as goals may share variables.

SLD Resolution: Substitutions

In the computed substitution, we are only interested in the variables that appear in the goal. We therefore define a ‘partial substitution’:

Given $\sigma : X \to T_{\Sigma,X}$ and $Y \subseteq X$, we define:

$$\sigma_{|Y} \triangleq \begin{cases} \sigma(x) & \text{if } x \in Y \\ x & \text{otherwise} \end{cases}$$

In resolution we then use σ and $\widehat{\sigma}$:

$$a_1, \ldots, a_i, \ldots, a_k \nwarrow_{\widehat{\sigma}} (a_1, \ldots, b_1, \ldots, a_n, \ldots, a_k)\, \sigma$$

where $\widehat{\sigma} \triangleq \sigma_{|Y}$ with $Y = \mathrm{vars}(a_1, \ldots, a_k)$.

Example: Summation (1/4)

- Define:

$$\Sigma_0 = \{0, \ldots\} \quad \Pi_3 = \{\mathsf{sum}, \ldots\}$$
$$\Sigma_1 = \{\mathsf{succ}, \ldots\}$$

(We write '$\mathsf{s}(t)$' instead of '$\mathsf{succ}(t)$', for convenience.)

Example: Summation (1/4)

- Define:

$$\Sigma_0 = \{0, \ldots\} \quad \Pi_3 = \{\mathsf{sum}, \ldots\}$$
$$\Sigma_1 = \{\mathsf{succ}, \ldots\}$$

(We write '$\mathsf{s}(t)$' instead of '$\mathsf{succ}(t)$', for convenience.)

- Sum as a predicate: $\mathsf{sum}(x, y, z)$ means '$x + y = z$'.
- The set L:

$$\mathsf{sum}(0, y, y).$$
$$\mathsf{sum}(\mathsf{s}(x), y, \mathsf{s}(z)) :\!- \mathsf{sum}(x, y, z).$$

Example: Summation (1/4)

- Define:

$$\Sigma_0 = \{0, \ldots\} \quad \Pi_3 = \{\mathsf{sum}, \ldots\}$$
$$\Sigma_1 = \{\mathsf{succ}, \ldots\}$$

  (We write '$\mathsf{s}(t)$' instead of '$\mathsf{succ}(t)$', for convenience.)
- Sum as a predicate: $\mathsf{sum}(x, y, z)$ means '$x + y = z$'.
- The set L:

$$\mathsf{sum}(0, y, y).$$
$$\mathsf{sum}(\mathsf{s}(x), y, \mathsf{s}(z)) \coloneq \mathsf{sum}(x, y, z).$$

- Let's target the goal: $?\!- \mathsf{sum}(\mathsf{s}(\mathsf{s}(0)), \mathsf{s}(\mathsf{s}(0)), n)$. (That is: '$2 + 2 = ?$')

Example: Summation (2/4)

Given our goal $\mathsf{sum}(\mathsf{s}(\mathsf{s}(0)), \mathsf{s}(\mathsf{s}(0)), n)$ and the set L:

- $\{\mathsf{sum}(\mathsf{s}(\mathsf{s}(0)), \mathsf{s}(\mathsf{s}(0)), n) \overset{?}{=} \mathsf{sum}(0, y', y')\}$ **fails!**

Example: Summation (2/4)

Given our goal $\mathsf{sum}(\mathsf{s}(\mathsf{s}(0)), \mathsf{s}(\mathsf{s}(0)), n)$ and the set L:

- $\{\mathsf{sum}(\mathsf{s}(\mathsf{s}(0)), \mathsf{s}(\mathsf{s}(0)), n) \overset{?}{=} \mathsf{sum}(0, y', y')\}$ **fails!**
- $\{\mathsf{sum}(\mathsf{s}(\mathsf{s}(0)), \mathsf{s}(\mathsf{s}(0)), n) \overset{?}{=} \mathsf{sum}(\mathsf{s}(x_1), y_1, \mathsf{s}(z_1))\}$ **succeeds!**

  We have

$$\sigma_1 = [x_1 = \mathsf{s}(0), y_1 = \mathsf{s}(\mathsf{s}(0)), n = \mathsf{s}(z_1)]$$
$$\widehat{\sigma_1} = [n = \mathsf{s}(z_1)]$$

Example: Summation (2/4)

Given our goal $\mathsf{sum}(\mathsf{s}(\mathsf{s}(0)), \mathsf{s}(\mathsf{s}(0)), n)$ and the set L:

- $\{\mathsf{sum}(\mathsf{s}(\mathsf{s}(0)), \mathsf{s}(\mathsf{s}(0)), n) \overset{?}{=} \mathsf{sum}(0, y', y')\}$ **fails!**
- $\{\mathsf{sum}(\mathsf{s}(\mathsf{s}(0)), \mathsf{s}(\mathsf{s}(0)), n) \overset{?}{=} \mathsf{sum}(\mathsf{s}(x_1), y_1, \mathsf{s}(z_1))\}$ **succeeds!**

  We have

$$\sigma_1 = [x_1 = \mathsf{s}(0), y_1 = \mathsf{s}(\mathsf{s}(0)), n = \mathsf{s}(z_1)]$$
$$\widehat{\sigma_1} = [n = \mathsf{s}(z_1)]$$

  Therefore:

$$\begin{aligned} \mathsf{sum}(\mathsf{s}(\mathsf{s}(0)), \mathsf{s}(\mathsf{s}(0)), n) \nwarrow_{\widehat{\sigma_1}} & (\mathsf{sum}(x_1, y_1, z_1))\sigma_1 \\ &= \mathsf{sum}(\mathsf{s}(0), \mathsf{s}(\mathsf{s}(0)), z_1) \end{aligned}$$

Example: Summation (2/4)

Given our goal $\mathsf{sum}(\mathsf{s}(\mathsf{s}(0)), \mathsf{s}(\mathsf{s}(0)), n)$ and the set L:

- $\{\mathsf{sum}(\mathsf{s}(\mathsf{s}(0)), \mathsf{s}(\mathsf{s}(0)), n) \overset{?}{=} \mathsf{sum}(0, y', y')\}$ **fails!**
- $\{\mathsf{sum}(\mathsf{s}(\mathsf{s}(0)), \mathsf{s}(\mathsf{s}(0)), n) \overset{?}{=} \mathsf{sum}(\mathsf{s}(x_1), y_1, \mathsf{s}(z_1))\}$ **succeeds!**

  We have

$$\begin{aligned} \sigma_1 &= [x_1 = \mathsf{s}(0), y_1 = \mathsf{s}(\mathsf{s}(0)), n = \mathsf{s}(z_1)] \\ \widehat{\sigma_1} &= [n = \mathsf{s}(z_1)] \end{aligned}$$

  Therefore:

$$\begin{aligned} \mathsf{sum}(\mathsf{s}(\mathsf{s}(0)), \mathsf{s}(\mathsf{s}(0)), n) \nwarrow_{\widehat{\sigma_1}} & (\mathsf{sum}(x_1, y_1, z_1))\sigma_1 \\ &= \mathsf{sum}(\mathsf{s}(0), \mathsf{s}(\mathsf{s}(0)), z_1) \end{aligned}$$

  Note: we write $\nwarrow_{\widehat{\sigma_1}}$ because $\widehat{\sigma_1}$ gives the least condition for the derivation.

Example: Summation (3/4)

Given our new goal $\mathsf{sum}(\mathsf{s}(0), \mathsf{s}(\mathsf{s}(0)), z_1)$ and the set L:

- $\{\mathsf{sum}(\mathsf{s}(0), \mathsf{s}(\mathsf{s}(0)), z_1) \overset{?}{=} \mathsf{sum}(0, y', y')\}$ **fails!**

Example: Summation (3/4)

Given our new goal $\mathsf{sum}(\mathsf{s}(0), \mathsf{s}(\mathsf{s}(0)), z_1)$ and the set L:

- $\{\mathsf{sum}(\mathsf{s}(0), \mathsf{s}(\mathsf{s}(0)), z_1) \overset{?}{=} \mathsf{sum}(0, y', y')\}$ **fails!**
- $\{\mathsf{sum}(\mathsf{s}(0), \mathsf{s}(\mathsf{s}(0)), z_1) \overset{?}{=} \mathsf{sum}(\mathsf{s}(x_2), y_2, \mathsf{s}(z_2))\}$ **succeeds!**

  We have

$$\sigma_2 = [x_2 = 0, y_2 = \mathsf{s}(\mathsf{s}(0)), n = \mathsf{s}(z_2)]$$
$$\widehat{\sigma_2} = [z_1 = \mathsf{s}(z_2)]$$

Example: Summation (3/4)

Given our new goal $\mathsf{sum}(\mathsf{s}(0), \mathsf{s}(\mathsf{s}(0)), z_1)$ and the set L:

- $\{\mathsf{sum}(\mathsf{s}(0), \mathsf{s}(\mathsf{s}(0)), z_1) \overset{?}{=} \mathsf{sum}(0, y', y')\}$ **fails!**
- $\{\mathsf{sum}(\mathsf{s}(0), \mathsf{s}(\mathsf{s}(0)), z_1) \overset{?}{=} \mathsf{sum}(\mathsf{s}(x_2), y_2, \mathsf{s}(z_2))\}$ **succeeds!**

  We have

$$\begin{aligned} \sigma_2 &= [x_2 = 0, y_2 = \mathsf{s}(\mathsf{s}(0)), n = \mathsf{s}(z_2)] \\ \widehat{\sigma_2} &= [z_1 = \mathsf{s}(z_2)] \end{aligned}$$

  Therefore:

$$\begin{aligned} \mathsf{sum}(\mathsf{s}(\mathsf{s}(0)), \mathsf{s}(\mathsf{s}(0)), n) \nwarrow_{\widehat{\sigma_1}} & \ \mathsf{sum}(\mathsf{s}(0), \mathsf{s}(\mathsf{s}(0)), z_1) \\ \nwarrow_{\widehat{\sigma_2}} & \ (\mathsf{sum}(x_2, y_2, z_2))\sigma_2 \\ = & \ \mathsf{sum}(0, \mathsf{s}(\mathsf{s}(0)), z_2) \end{aligned}$$

Example: Summation (4/4)

Given our new goal $\mathsf{sum}(0, \mathsf{s}(\mathsf{s}(0)), z_2)$ and the set L:

- $\{\mathsf{sum}(0, \mathsf{s}(\mathsf{s}(0)), z_2) \overset{?}{=} \mathsf{sum}(0, y_3, y_3)\}$ **succeeds!**

  We have

$$\sigma_3 = [y_3 = \mathsf{s}(\mathsf{s}(0)), z_2 = \mathsf{s}(\mathsf{s}(0))]$$
$$\widehat{\sigma_3} = [z_2 = \mathsf{s}(\mathsf{s}(0))]$$

Example: Summation (4/4)

Given our new goal $\mathsf{sum}(0, \mathsf{s}(\mathsf{s}(0)), z_2)$ and the set L:

- $\{\mathsf{sum}(0, \mathsf{s}(\mathsf{s}(0)), z_2) \overset{?}{=} \mathsf{sum}(0, y_3, y_3)\}$ **succeeds!**

  We have

$$\sigma_3 = [y_3 = \mathsf{s}(\mathsf{s}(0)), z_2 = \mathsf{s}(\mathsf{s}(0))]$$
$$\widehat{\sigma_3} = [z_2 = \mathsf{s}(\mathsf{s}(0))]$$

  Therefore:

$$\begin{aligned} \mathsf{sum}(\mathsf{s}(\mathsf{s}(0)), \mathsf{s}(\mathsf{s}(0)), n) &\nwarrow_{\widehat{\sigma_1}} \mathsf{sum}(\mathsf{s}(0), \mathsf{s}(\mathsf{s}(0)), z_1) \\ &\nwarrow_{\widehat{\sigma_2}} \mathsf{sum}(0, \mathsf{s}(\mathsf{s}(0)), z_2) \\ &\nwarrow_{\widehat{\sigma_3}} \square \end{aligned}$$

Example: Summation (4/4)

Given our new goal $\mathsf{sum}(0, \mathsf{s}(\mathsf{s}(0)), z_2)$ and the set L:

- $\{\mathsf{sum}(0, \mathsf{s}(\mathsf{s}(0)), z_2) \stackrel{?}{=} \mathsf{sum}(0, y_3, y_3)\}$ **succeeds!**

  We have

$$\sigma_3 = [y_3 = \mathsf{s}(\mathsf{s}(0)), z_2 = \mathsf{s}(\mathsf{s}(0))]$$
$$\widehat{\sigma_3} = [z_2 = \mathsf{s}(\mathsf{s}(0))]$$

  Therefore:

$$\begin{aligned} \mathsf{sum}(\mathsf{s}(\mathsf{s}(0)), \mathsf{s}(\mathsf{s}(0)), n) &\nwarrow_{\widehat{\sigma_1}} \mathsf{sum}(\mathsf{s}(0), \mathsf{s}(\mathsf{s}(0)), z_1) \\ &\nwarrow_{\widehat{\sigma_2}} \mathsf{sum}(0, \mathsf{s}(\mathsf{s}(0)), z_2) \\ &\nwarrow_{\widehat{\sigma_3}} \square \end{aligned}$$

- Recall: $\widehat{\sigma_1} = [n = \mathsf{s}(z_1)]$, $\widehat{\sigma_2} = [z_1 = \mathsf{s}(z_2)]$, $\widehat{\sigma_3} = [z_2 = \mathsf{s}(\mathsf{s}(0))]$.
  We conclude: $\widehat{\sigma_1} \cdot \widehat{\sigma_2} \cdot \widehat{\sigma_3} = [n = \mathsf{s}(\mathsf{s}(\mathsf{s}(\mathsf{s}(0))))]$, as desired.

The End