Part 3
SLD and coSLD resolution

Terminology

In part 2 we have seen that logic programs are specific inference systems
We used interchangeably terminology of inference systems and Prolog

From now on we adopt the Prolog terminology

- **logic program** = an inference system
- **Horn clause** = a meta-rule
- **ground instantiation of a Horn clause** = a rule
- **head of a clause** = conclusion of a meta-rule
- **body of a clause** = premises of a meta-rule
- **fact** = an axiom

Defining functions in Prolog

Functions as predicates

So far we have seen examples of predicates (functions returning *false* or *true*)

Problem: how can we define addition on natural numbers?

Solution: we introduce the predicate $add/3$ where the last argument is the result of the operation

Examples:

$add(s(z), s(s(z)), s(s(s(z))))$ holds

$add(s(z), s(s(z)), z)$ **does not hold**

More on functions in Prolog

Clauses defining *add*/3

(C1) $add(z, N, N)$.

(C2) $add(s(N), M, s(K))$:- $add(N, M, K)$.

Abstract and operational semantics

- The abstract syntax is concise and simple
- But useless for computing
- If we could only check whether ground atoms holds then we would not be able to compute functions
- More expressive queries are needed, and a corresponding operational semantics must be defined
- The operational semantics must be consistent with the abstract one

Queries (or goals)

Examples

The system must be able to solve **queries** (or **goals**)

?- $add(s(z), s(s(z)), N)$.

Meaning: find all substitutions $\{N \mapsto t\}$ s.t. $add(s(z), s(s(z)), t)$ holds (w.r.t. the abstract semantics)

Computed answer: $\{N \mapsto s(s(s(z)))\}$

Queries may involve more atoms

?- $geq(s(s(z)), N), add(s(z), M, N)$.

And there can be several computed answers

$\{M \mapsto z, N \mapsto s(z)\}, \{M \mapsto s(z), N \mapsto s(s(z))\}$

Most general substitution

Example

For capturing all answers the most general substitution must be computed

?- $add(z, N, M)$.

Computed answer: $\{N \mapsto M\}$

Meaning: all ground instantiations of $add(z, M, M)$ holds (w.r.t. the abstract semantics)

More on the operational semantics

Based on the SLD resolution

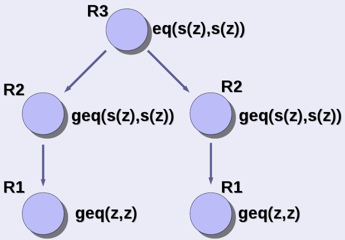
- resolution: inference rule based on refutation
- refutation: theorem proving technique (based on proof by contradiction)
- SLD means: Selective Linear Definite clause
- meaning of Selective and Linear explained later on

Why linear?

Example

Proof trees are linearized

A proof tree

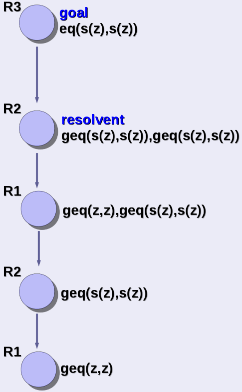


Why linear?

Example

Proof trees are linearized

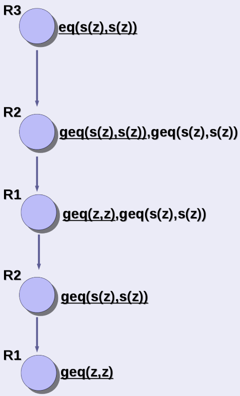
The same proof tree, but linearized.
Reasons: sequential process, easier implementation



Why selective?

Example

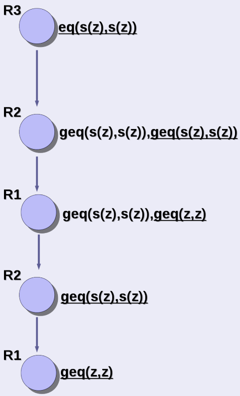
At each step one atom is selected according to a selection rule
Previous example obtained with left-most selection rule (as in Prolog)



Why selective?

Example

At each step one atom is selected according to a selection rule
Linearization obtained with right-most selection rule



Non ground goals

Example 1

?- $add(N, M, s(z))$. $\sigma_0 = \{\}$

Non ground goals

Example 1

?- $add(N, M, s(z))$. $\sigma_0 = \{\}$

Applied clause (C1) $\sigma_1 = mgu(add(N, M, s(z)), add(z, N_1, N_1))$

Non ground goals

Example 1

?- $add(N, M, s(z))$. $\sigma_0 = \{\}$

Applied clause (C1) $\sigma_1 = mgu(add(N, M, s(z)), add(z, N_1, N_1))$

?- *true*. $\sigma_0\sigma_1 = \{N \mapsto z, M \mapsto s(z), N_1 \mapsto s(z)\}$

Non ground goals

Example 1

?- $add(N, M, s(z))$. $\sigma_0 = \{\}$

Applied clause (C1) $\sigma_1 = mgu(add(N, M, s(z)), add(z, N_1, N_1))$

?- *true*. $\sigma_0\sigma_1 = \{N \mapsto z, M \mapsto s(z), N_1 \mapsto s(z)\}$

Computed answer: $\{N \mapsto z, M \mapsto s(z)\}$

Non ground goals

Example 2

?- $add(N, M, s(z))$. $\sigma_0 = \{\}$

Non ground goals

Example 2

?- $add(N, M, s(z))$. $\sigma_0 = \{\}$

Applied clause (C2) $\sigma_1 = mgu(add(N, M, s(z)), add(s(N_1), M_1, s(K_1)))$

Non ground goals

Example 2

?- $add(N, M, s(z))$. $\sigma_0 = \{\}$

Applied clause (C2) $\sigma_1 = mgu(add(N, M, s(z)), add(s(N_1), M_1, s(K_1)))$

?- $add(N_1, M_1, z)$. $\sigma_0\sigma_1 = \{N \mapsto s(N_1), M \mapsto M_1, K_1 \mapsto z\}$

Non ground goals

Example 2

?- $add(N, M, s(z))$. $\sigma_0 = \{\}$

Applied clause (C2) $\sigma_1 = mgu(add(N, M, s(z)), add(s(N_1), M_1, s(K_1)))$

?- $add(N_1, M_1, z)$. $\sigma_0\sigma_1 = \{N \mapsto s(N_1), M \mapsto M_1, K_1 \mapsto z\}$

Applied clause (C1) $\sigma_2 = mgu(add(N_1, M_1, z), add(z, N_2, N_2))$

Non ground goals

Example 2

?- $add(N, M, s(z))$. $\sigma_0 = \{\}$

Applied clause (C2) $\sigma_1 = mgu(add(N, M, s(z)), add(s(N_1), M_1, s(K_1)))$

?- $add(N_1, M_1, z)$. $\sigma_0\sigma_1 = \{N \mapsto s(N_1), M \mapsto M_1, K_1 \mapsto z\}$

Applied clause (C1) $\sigma_2 = mgu(add(N_1, M_1, z), add(z, N_2, N_2))$

?- $true$. $\sigma_0\sigma_1\sigma_2 = \{N \mapsto s(z), M \mapsto z, K_1 \mapsto z, N_1 \mapsto z, M_1 \mapsto z, N_2 \mapsto z\}$

Non ground goals

Example 2

?- $add(N, M, s(z))$. $\sigma_0 = \{\}$

Applied clause (C2) $\sigma_1 = mgu(add(N, M, s(z)), add(s(N_1), M_1, s(K_1)))$

?- $add(N_1, M_1, z)$. $\sigma_0\sigma_1 = \{N \mapsto s(N_1), M \mapsto M_1, K_1 \mapsto z\}$

Applied clause (C1) $\sigma_2 = mgu(add(N_1, M_1, z), add(z, N_2, N_2))$

?- $true$. $\sigma_0\sigma_1\sigma_2 = \{N \mapsto s(z), M \mapsto z, K_1 \mapsto z, N_1 \mapsto z, M_1 \mapsto z, N_2 \mapsto z\}$

Computed answer: $\{N \mapsto s(z), M \mapsto z\}$

Definition of SLD resolution

Meta-rules

$$\frac{sld(G, \emptyset, \sigma)}{sld(G, \sigma)} \qquad \frac{}{sld([\,], \sigma, \sigma)}$$

$$\frac{clause(A, \sigma_1, G2, \sigma_2) \quad sld(G2, \sigma_1\sigma_2, \sigma_3) \quad sld(G1, \sigma_3, \sigma)}{sld([A|G1], \sigma_1, \sigma)}$$

Explanations:

- $sld(G, \sigma)$: σ is the most general substitution s.t. all ground instances of $G\sigma$ holds
- $sld(G, \sigma_1, \sigma_2)$: σ_2 is the most general substitution s.t. $(G\sigma_1)\sigma_2$ holds; σ_1 corresponds to the substitution computed so far
- $[A|G]$ is a list where A is the left-most atom, G is the rest of the list
- $clause(A, \sigma_1, G, \sigma)$ holds is there exists a clause, with all fresh variables, where σ is the mgu between $A\sigma_1$ and the head, and G is the body
- $\sigma_1\sigma_2$ is the substitution σ s.t. $A\sigma = (A\sigma_1)\sigma_2$ for all atoms A

Definition of coSLD resolution

Meta-rules

$$\frac{sld([\,], G, \emptyset, \sigma)}{sld(G, \sigma)} \qquad \frac{}{sld(H, [\,], \sigma, \sigma)}$$

$$\frac{clause(A, \sigma_1, G2, \sigma_2) \quad sld([A|H], G2, \sigma_1\sigma_2, \sigma_3) \quad sld(H, G, \sigma_3, \sigma)}{sld(H, [A|G1], \sigma_1, \sigma)}$$

$$\frac{member(H, A, \sigma_1, \sigma_2) \quad sld(H, G, \sigma_1\sigma_2, \sigma)}{sld(H, [A|G], \sigma_1, \sigma)}$$

Explanations:

- $sld(H, G, \sigma_1, \sigma_2)$: σ_2 is the most general substitution s.t. $(G\sigma_1)\sigma_2$ holds; σ_1 corresponds to the substitution computed so far, H to the list of atoms resolved so far
- $member(H, A, \sigma_1, \sigma_2)$: there exists an atom A' in the list H s.t. σ_2 is the mgu between $A\sigma_1$ and A'