Dresden University of Technology

Computational Logic ∴ Group

Hannes Strass (based on slides by Michael Thielscher)

Faculty of Computer Science, Institute of Artificial Intelligence, Computational Logic Group

Correctness of SLD Resolution

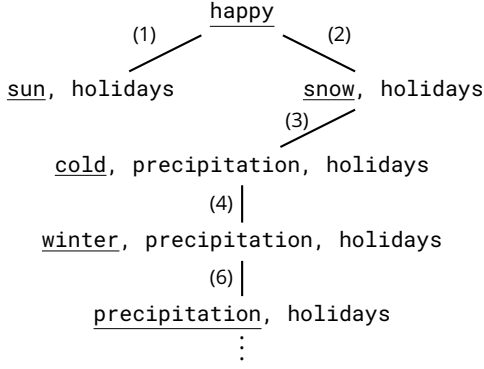
Lecture 4, 2nd Nov 2026 // Foundations of Logic Programming, WS 2026/27

Previously . . .

- A proof theory for (definite) logic programs is given by **SLD resolution**.
- A query is resolved with a (variant of a) program clause to another query.
- There are choices to be made (renaming of clause, mgu of query atom and clause, selected atom in query, program clause) with consequences.
- The search space can be visualized by (selection rule-induced) **SLD trees**.

```
(1) happy :- sun, holidays.
(2) happy :- snow, holidays.
(3) snow :- cold, precipitation.
(4) cold :- winter.
(5) precipitation :- holidays.
(6) winter.
(7) holidays.

| ?- happy.
```



Overview

Model Theory: Algebras, Interpretations, and Models

Soundness of SLD Resolution

Completeness of SLD Resolution

Model Theory: Algebras, Interpretations, and Models

Algebras (Semantics of Terms)

Definition

Let V be a set of variables, F be a ranked alphabet of function symbols.
An **algebra** J for F (or **F-algebra** or **pre-interpretation** for F) consists of:

1. A **domain**, a non-empty set D;
2. the assignment of a mapping $f_J \colon D^n \to D$ to every $f \in F^{(n)}$ with $n \geq 0$.

For $f \in F^{(0)}$, the constant function $f_J \colon D^0 \to D$ maps () to some $d \in D$.

Definition

A **state** σ over D is a mapping $\sigma \colon V \to D$.
The extension of σ to $TU_{F,V}$ by algebra J is the function $\sigma_J \colon TU_{F,V} \to D$ such that:

- $\sigma_J(v) := \sigma(v)$ for every $v \in V$, and
- $\sigma_J(f(t_1, \ldots, t_n)) := f_J(\sigma_J(t_1), \ldots, \sigma_J(t_n))$ for every $f \in F^{(n)}$.

In first-order predicate logic, a state is typically called a variable assignment.

Interpretations (Semantics of Programs)

Definition

Let F be a ranked alphabet of function symbols, Π be a ranked alphabet of predicate symbols.

An **interpretation** I for F and Π consists of :

1. An algebra J for F (with domain D);
2. for every $p \in \Pi^{(n)}$ with $n \geq 0$, the assignment of a relation

$$p_I \subseteq \underbrace{D \times \cdots \times D}_{n}$$

$\rightsquigarrow$ Standard definition of first-order logic interpretations.

Note

For $p \in \Pi^{(0)}$, we have $p_I \subseteq \{()\}$, that is, either $p_I = \emptyset$ (false) or $p_I = \{()\}$ (true).

Interpretations (Example)

$$add(x, 0, x) \leftarrow$$
$$add(x, s(y), s(z)) \leftarrow add(x, y, z)$$

Consider the addition program, P_{add}:
I_1, I_2, I_3, I_4, I_5, and I_6 are interpretations for $\{s, 0\}$ and $\{add\}$:

I_1: $D_{I_1} = \mathbf{N}$, $0_{I_1} = 0$, $s_{I_1} = \{n \mapsto n+1 \mid n \in \mathbf{N}\}$, $add_{I_1} = \{(m, n, m+n) \mid m, n \in \mathbf{N}\}$

I_2: $D_{I_2} = \mathbf{N}$, $0_{I_2} = 0$, $s_{I_2} = \{n \mapsto n+1 \mid n \in \mathbf{N}\}$, $add_{I_2} = \{(m, n, m*n) \mid m, n \in \mathbf{N}\}$

I_3: $D_{I_3} = HU_{\{s,0\}}$, $0_{I_3} = 0$, $s_{I_3} = \{t \mapsto s(t) \mid t \in HU_{\{s,0\}}\}$,
$add_{I_3} = \{(s^m(0), s^n(0), s^{m+n}(0)) \mid m, n \in \mathbf{N}\}$

I_4: $D_{I_4} = HU_{\{s,0\}}$, $0_{I_4} = 0$, $s_{I_4} = \{t \mapsto s(t) \mid t \in HU_{\{s,0\}}\}$, $add_{I_4} = \emptyset$

I_5: $D_{I_5} = HU_{\{s,0\}}$, $0_{I_5} = 0$, $s_{I_5} = \{t \mapsto s(t) \mid t \in HU_{\{s,0\}}\}$, $add_{I_5} = (HU_{\{s,0\}})^3$

I_6: $D_{I_6} = \{0, 1\}$, $0_{I_6} = 0$, $s_{I_6} = \{0 \mapsto 0, 1 \mapsto 1\}$, $add_{I_6} = \{(m, n, m) \mid m, n \in \{0, 1\}\}$

Logical Truth (1)

Definition

An **expression** E is an atom, a query, a clause, or a resultant.

Definition

Let E be an expression, I be an interpretation, σ be a state.
We say that E is **true in** I **under** σ and write $I, \sigma \Vdash E$ defined by case analysis on E:

$$
\begin{aligned}
I, \sigma \Vdash p(t_1, \ldots, t_n) &\quad :\Longleftrightarrow \quad (\sigma_I(t_1), \ldots, \sigma_I(t_n)) \in p_I \\
I, \sigma \Vdash A_1, \ldots, A_n &\quad :\Longleftrightarrow \quad I, \sigma \Vdash A_i \text{ for every } 1 \leq i \leq n \\
I, \sigma \Vdash A \leftarrow \vec{B} &\quad :\Longleftrightarrow \quad \text{if } I, \sigma \Vdash \vec{B} \text{ then } I, \sigma \Vdash A \\
I, \sigma \Vdash \vec{A} \leftarrow \vec{B} &\quad :\Longleftrightarrow \quad \text{if } I, \sigma \Vdash \vec{B} \text{ then } I, \sigma \Vdash \vec{A}
\end{aligned}
$$

Logical Truth (2)

Definition

Let E be an expression and I be an interpretation.

Furthermore, let $x_1, \ldots, x_k$ be the variables occuring in E.

- $\forall x_1, \ldots, \forall x_k\, E$ is the **universal closure** of E (abbreviated $\forall E$)
- $\exists x_1, \ldots, \exists x_k\, E$ is the **existential closure** of E (abbreviated $\exists E$)
- $I \Vdash \forall E \;:\Longleftrightarrow\; I, \sigma \Vdash E$ for every state σ
- $I \Vdash \exists E \;:\Longleftrightarrow\; I, \sigma \Vdash E$ for some state σ
- E is **true in** I (or: I is a **model of** E), written: $I \Vdash E \;:\Longleftrightarrow\; I \Vdash \forall E$

$\rightsquigarrow$ Standard first-order logic definition of logical truth (for expressions).

Logical Truth (3), Entailment, Correct Answers

Definition

Let S and T be sets of expressions and I be an interpretation.

- I is a **model of** S, written: $I \Vdash S \;:\Longleftrightarrow\; I \Vdash E$ for every $E \in S$
- T is a **logical consequence of** S, written: $S \models T$
  $:\Longleftrightarrow$ every model of S is a model of T

We sometimes refer to logical consequences as **semantic** consequences to stress their model-theoretic definition.

Definition

Let P be a program, Q_0 be a query, and θ be a substitution.

- $\theta|_{Var(Q_0)}$ is a **correct answer substitution** of $Q_0 \;:\Longleftrightarrow\; P \models Q_0\theta$
- $Q_0\theta$ is a **correct instance** of $Q_0 \;:\Longleftrightarrow\; P \models Q_0\theta$

$\rightsquigarrow$ Model-theoretic counterparts to *computed* answer substitutions/instances.

Models (Example)

Consider again P_{add}:

$$add(x, 0, x) \leftarrow$$
$$add(x, s(y), s(z)) \leftarrow add(x, y, z)$$

Furthermore, let I_1, I_2, I_3, I_4, I_5, and I_6 be the interpretations from slide 7.

- $I_1 \Vdash P_{add}$, since $I_1, \sigma \Vdash c$ for every clause $c \in P_{add}$ and state $\sigma\colon V \to \mathbb{N}$:
  1. $(\sigma(x), \sigma(0), \sigma(x)) \in add_{I_1}$ and
  2. if $(\sigma(x), \sigma(y), \sigma(z)) \in add_{I_1}$ then $(\sigma(x), \sigma(y) + 1, \sigma(z) + 1) \in add_{I_1}$.
- $I_2 \not\Vdash P_{add}$:
  (E.g. let $\sigma(x) = 1$, then $I_2, \sigma \not\Vdash add(x, 0, x)$ since $(\sigma(x), \sigma(0), \sigma(x)) = (1, 0, 1) \notin add_{I_2}$.)
- $I_3 \Vdash P_{add}$ (like for I_1; we call I_3 a (least) Herbrand model)
- $I_4 \not\Vdash P_{add}$ (e.g. let $\sigma(x) = s(0)$, then $I_4, \sigma \not\Vdash add(x, 0, x)$ since $(\sigma(x), \sigma(0), \sigma(x)) = (s(0), 0, s(0)) \notin add_{I_4}$)
- $I_5 \Vdash P_{add}$ (like for I_1; we call I_5 a Herbrand model)
- $I_6 \Vdash P_{add}$ (like for I_1)

$I_1\colon D_{I_1} = \mathbb{N}$, $0_{I_1} = 0$, $s_{I_1} = \{n \mapsto n + 1 \mid n \in \mathbb{N}\}$, $add_{I_1} = \{(n, 0, n), (n, s(n), s(m)) \mid (n, n, m) \in add_{I_1}\}$

Semantic Consequences (Example)

Consider again the addition program P_{add}.

- $P_{add} \models add(x, 0, x)$
  (For every interpretation I: if $I \Vdash P_{add}$ then $I \Vdash add(x, 0, x)$, since $add(x, 0, x) \in P_{add}$.)
- $P_{add} \models add(x, s(0), s(x))$
  (For every interpretation I: if $I \Vdash P_{add}$ then $I \Vdash add(x, 0, x)$ and $I \Vdash add(x, s(0), s(x)) \leftarrow add(x, 0, x)$ (instance of clause), thus $I \Vdash add(x, s(0), s(x))$.)
- $P_{add} \not\models add(0, x, x)$
  (Consider interpretation I_6 from slide 7 with $I_6 \Vdash P_{add}$; $I_6 \not\Vdash add(0, x, x)$, since e.g. $I_6, \sigma \not\Vdash add(0, x, x)$ for $\sigma(x) = 1$, since $(\sigma(0), \sigma(x), \sigma(x)) = (0, 1, 1) \notin add_{I_6}$.)

Quiz: Models and Consequences

Quiz

Consider the following logic program P where only x is a variable: …

Soundness of SLD Resolution

Towards Soundness of SLD Resolution (1)

Lemma 4.3 (i)

Let $Q \xrightarrow[c]{\theta} Q'$ be an SLD derivation step and $Q\theta \leftarrow Q'$ the resultant associated with it. Then

$$c \models Q\theta \leftarrow Q'$$

Proof.

Let $Q = \vec{A}, B, \vec{C}$ with selected atom B. Let $H \leftarrow \vec{B}$ be the input clause and $Q' = (\vec{A}, \vec{B}, \vec{C})\theta$. Then:

	$c \models H \leftarrow \vec{B}$	(variant of c)
implies	$c \models H\theta \leftarrow \vec{B}\theta$	(instance)
implies	$c \models B\theta \leftarrow \vec{B}\theta$	(θ is a unifier of B and H)
implies	$c \models (\vec{A}, B, \vec{C})\theta \leftarrow (\vec{A}, \vec{B}, \vec{C})\theta$	("context" unchanged)

□

Intuitively: The resultant is a logical consequence of the program clause.

Towards Soundness of SLD Resolution (2)

Lemma 4.3 (ii)

Let ξ be an SLD derivation of $P \cup \{Q_0\}$. For $i \geq 0$, let R_i be the resultant of level i of ξ. Then

$$P \models R_i$$

Proof.

Let $\xi = Q_0 \xrightarrow[c_1]{\theta_1} Q_1 \cdots Q_n \xrightarrow[c_{n+1}]{\theta_{n+1}} Q_{n+1} \cdots$. We use induction on $i \geq 0$:

$i = 0$: $R_0 = Q_0 \leftarrow Q_0$ is equivalent to true, thus $P \models R_0$

$i = 1$: $R_1 = Q_0\theta_1 \leftarrow Q_1$; by Lemma 4.3 (i): $P \models R_1$

$i \rightsquigarrow i+1$: By Lemma 4.3 (i), $c_{i+1} \models Q_i\theta_{i+1} \leftarrow Q_{i+1}$, thus $P \models Q_i\theta_{i+1} \leftarrow Q_{i+1}$.
By (IH), $P \models R_i$, that is, $P \models Q_0\theta_1 \cdots \theta_i \leftarrow Q_i$ and in particular
$P \models Q_0\theta_1 \cdots \theta_i\theta_{i+1} \leftarrow Q_i\theta_{i+1}$. In combination,
$P \models Q_0\theta_1 \cdots \theta_i\theta_{i+1} \leftarrow Q_{i+1}$, that is, $P \models R_{i+1}$. □

Soundness of SLD Resolution

Theorem 4.4

If there exists a successful SLD derivation of $P \cup \{Q_0\}$ with cas θ, then

$$P \models Q_0\theta$$

Proof.

Let $\xi = Q_0 \xrightarrow{\theta_1} \cdots \xrightarrow{\theta_n} \square$ be a successful SLD derivation.

Lemma 4.3 (ii) applied to the resultant of level n of ξ implies $P \models Q_0\theta_1 \cdots \theta_n$ and $Q_0\theta_1 \cdots \theta_n = Q_0(\theta_1 \cdots \theta_n|_{Var(Q_0)}) = Q_0\theta$. $\square$

Completeness of SLD Resolution

Towards Completeness of SLD Resolution

We aim to show a result of the following form, with P a program and Q a query:

$$\text{If } P \models Q, \text{ then there exists a successful SLD derivation of } P \cup \{Q\}.$$

To show this, we need to syntactically characterize the set of (semantically) entailed queries.

The concepts of term models and implication trees serve this purpose.

Definition

Let E be an expression and S be a set of expressions.

- $\mathit{inst}(E) \ :\Longleftrightarrow$ set of all instances of E
- $\mathit{inst}(S) \ :\Longleftrightarrow$ set of all instances of elements $E \in S$
- $\mathit{ground}(E) \ :\Longleftrightarrow$ set of all ground instances of E
- $\mathit{ground}(S) \ :\Longleftrightarrow$ set of all ground instances of elements $E \in S$

Term Models

Definition

Let V be a set of variables, F function symbols, Π predicate symbols.

The **term algebra** J for F is defined as follows:

1. domain $D = TU_{F,V}$,
2. mapping $f_J \colon (TU_{F,V})^n \to TU_{F,V}$ assigned to every $f \in F^{(n)}$ with $f_J(t_1, \ldots, t_n) := f(t_1, \ldots, t_n)$

Definition

A **term interpretation** I for F and Π consists of:

1. term algebra for F,
2. $I \subseteq TB_{\Pi,F,V}$ (set of atoms that are true; equivalently: assignment of a relation $p_I \subseteq (TU_{F,V})^n$ to every $p \in \Pi^{(n)}$).

I is a **term model** of a set S of expressions
$:\iff$ I term interpretation and model of S.

Herbrand Models

Definition

The **Herbrand algebra** J for F is defined as follows:

1. domain $D = HU_F$
2. mapping $f_J \colon (HU_F)^n \to HU_F$ assigned to every $f \in F^{(n)}$ with $f_J(t_1, \ldots, t_n) := f(t_1, \ldots, t_n)$

Definition

A **Herbrand interpretation** I for F and Π consists of:

1. Herbrand algebra for F,
2. $I \subseteq HB_{\Pi,F}$ (set of ground atoms that are true).

I is a **Herbrand model** of a set S of expressions

$:\iff$ I is a Herbrand interpretation and a model of S

Implication Trees

Definition

$\mathcal{T}$ **implication tree** w.r.t. program P

$$:\Longleftrightarrow$$

- tree $\mathcal{T}$ is finite
- nodes are atoms
- if A is a node with the direct descendants $B_1, \ldots, B_n$ then $A \leftarrow B_1, \ldots, B_n \in inst(P)$
- if A is a leaf, then $A \leftarrow \in inst(P)$

$\mathcal{T}$ **ground implication tree** w.r.t. program P

$:\Longleftrightarrow$ $\mathcal{T}$ implication tree w.r.t. P and all nodes are ground atoms

Implication Trees (Example)

Let P_{add} be the addition program, $n \in \mathbb{N}$, V set of variables, $t \in TU_{\{s,0\},V}$.
Consider the tree $\mathcal{T}$ given by

$$
\begin{array}{c}
add(t, s^n(0), s^n(t)) \\
| \\
add(t, s^{n-1}(0), s^{n-1}(t)) \\
\\
\vdots \\
\\
add(t, s(0), s(t)) \\
| \\
add(t, 0, t)
\end{array}
$$

$\mathcal{T}$ is an implication tree w.r.t. P_{add}.
If additionally $t \in HU_{\{s,0\}}$, then $\mathcal{T}$ is a ground implication tree w.r.t. P_{add}.

Implication Trees Constitute Term Models

Lemma 4.7

Consider term interpretation I, atom A, program P.

- $I \Vdash A$ iff $inst(A) \subseteq I$
- $I \Vdash P$ iff for every $A \leftarrow B_1, \ldots, B_n \in inst(P)$,

$$\{B_1, \ldots, B_n\} \subseteq I \text{ implies } A \in I$$

Lemma 4.12

The term interpretation

$$\mathcal{C}(P) := \{\, A \mid A \text{ is the root of some implication tree w.r.t. } P \,\}$$

is a model of P.

Ground Implication Trees Constitute Herbrand Models

Lemma 4.26

Consider Herbrand interpretation I, atom A, program P.

- $I \Vdash A$ iff $ground(A) \subseteq I$
- $I \Vdash P$ iff for every $A \leftarrow B_1, \ldots, B_n \in ground(P)$,

$$\{B_1, \ldots, B_n\} \subseteq I \text{ implies } A \in I$$

Lemma 4.28

The Herbrand interpretation

$$\mathcal{M}(P) := \{ A \mid A \text{ is the root of some ground implication tree w.r.t. } P \}$$

is a model of P.

Constituted Models (Example)

Consider again the addition program P_{add} and the set V of variables.
The term interpretation

$$\begin{aligned}\mathcal{C}(P_{add}) &= \left\{ add(t, s^n(0), s^n(t)) \;\middle|\; n \in \mathbb{N}, t \in TU_{\{s,0\},V} \right\} \\ &= \left\{ add(s^m(v), s^n(0), s^{n+m}(v)) \;\middle|\; m, n \in \mathbb{N}, v \in V \cup \{0\} \right\}\end{aligned}$$

and the Herbrand interpretation

$$\begin{aligned}\mathcal{M}(P_{add}) &= \left\{ add(t, s^n(0), s^n(t)) \;\middle|\; n \in \mathbb{N}, t \in HU_{\{s,0\}} \right\} \\ &= \left\{ add(s^m(0), s^n(0), s^{n+m}(0)) \;\middle|\; m, n \in \mathbb{N} \right\}\end{aligned}$$

are models of P_{add}.

Correct vs. Computed Answer Substitutions

Consider P_{add}

$$add(x, 0, x) \leftarrow$$
$$add(x, s(y), s(z)) \leftarrow add(x, y, z)$$

along with the query $Q = add(u, s(0), s(u))$.

- $\theta = \{u/s^2(v)\}$ is a correct answer substitution of Q,
  since $P_{add} \models Q\theta = add(s^2(v), s(0), s^3(v))$ (in analogy to slide 12 with $x = s^2(v)$)
- SLD derivation of $P_{add} \cup \{Q\}$: $add(u, s(0), s(u)) \xrightarrow[(2)]{\theta_1} add(u, 0, u) \xrightarrow[(1)]{\theta_2} \square$
  with $\theta_1 = \{x/u, y/0, z/u\}$ and $\theta_2 = \{x/u\}$,
  thus $\eta = (\theta_1\theta_2)|_{\{u\}} = \varepsilon$ is a computed answer substitution of Q.
- We observe that η is strictly more general than θ.
- In fact, no SLD derivation of $P_{add} \cup \{Q\}$ can deliver θ.

Completeness for Implication Trees (1)

Definition

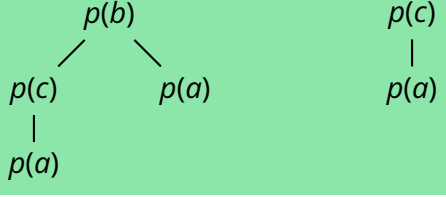
Query Q is **n-deep**

$$:\Longleftrightarrow$$

every atom in Q is the root of an implication tree, and n is the total number of nodes in these trees.

Example

Consider $P = \{p(a) \leftarrow, \quad p(c) \leftarrow p(a), \quad p(b) \leftarrow p(c), p(a)\}$. Then the query $Q = p(b), p(c)$ is 6-deep, as witnessed by these implication trees:



Completeness for Implication Trees (2)

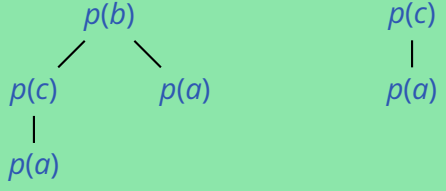
Lemma 4.15

Suppose that query $Q\theta$ is n-deep for some $n \geq 0$, where θ is a correct answer substitution of Q.

Then for every selection rule $\mathcal{R}$, there exists a successful SLD derivation of $P \cup \{Q\}$ with cas η such that η is at least as general as θ.

Example

Consider $P = \{p(a) \leftarrow, \quad p(c) \leftarrow p(a), \quad p(b) \leftarrow p(c), p(a)\}$ and implication trees



Completeness of SLD Resolution (1)

Theorem 4.13

Suppose that θ is a correct answer substitution of Q.

Then for every selection rule $\mathcal{R}$, there exists a successful SLD derivation of $P \cup \{Q\}$ with cas η such that η is at least as general as θ.

Proof.

Let $Q = A_1, \ldots, A_m$. Then:

	θ is a correct answer substitution of $A_1, \ldots, A_m$
implies	$P \models A_1\theta, \ldots, A_m\theta$
implies	for every interpretation I: if $I \Vdash P$, then $I \Vdash A_1\theta, \ldots, A_m\theta$
implies	$\mathcal{C}(P) \Vdash A_1\theta, \ldots, A_m\theta$ (since $\mathcal{C}(P) \Vdash P$ by Lemma 4.12)
implies	$inst(A_i\theta) \subseteq \mathcal{C}(P)$ for every $1 \leq i \leq m$ (by Lemma 4.7)
implies	$A_i\theta \in \mathcal{C}(P)$ for every $1 \leq i \leq m$
implies	$A_1\theta, \ldots, A_m\theta$ is n-deep for some $n \geq 0$ (by def. of $\mathcal{C}(P)$)
implies	claim (by Lemma 4.15) $\square$

Conclusion

Summary

- The semantics of (definite) logic programs is given by a standard first-order model theory, with logical entailment $\models$ defined as usual.
- SLD resolution is **sound**: For every successful SLD derivation of $P \cup \{Q_0\}$ with *computed* answer substitution θ, we have $P \models Q_0\theta$.
- SLD resolution is **complete**: If θ is a *correct* answer substitution of Q, then
  - for every selection rule
  - there exists a successful SLD derivation of $P \cup \{Q\}$ with cas η
  - such that η is at least as general as θ.

Suggested action points:

- Compare implication trees to SLD trees
- Clarify the distinction between *computed* and *correct* answer substitutions