Computational Logic

SLD resolution

Damiano Zanardini

UPM European Master in Computational Logic (EMCL)
School of Computer Science
Technical University of Madrid
damiano@fi.upm.es

Academic Year 2008/2009

Introduction

SLD: Selection function in Linear resolution for Definite clauses

- combines linear, input, directed and ordered strategies on a particular class of clauses

Horn clauses

- at most one non-negated literal (if it exists, it's the first in the clause)
  - $A \vee \neg B_1 \vee \neg B_2$
  - A
  - $\neg B_1 \vee \neg B_2$
- clauses *without* the non-negated literal form the *goal set*
- clauses *with* the non-negated literal form the *support set*

Introduction

Definition (SLD)

An SLD derivation of C_m from a set $\{C_1, .., C_n\}$ of Horn clauses (with the non-negated literal in the first place, if it exists) is a sequence $C_1, .., C_i, .., C_n, C_{n+1}, .., C_m$ such that

- C_{n+1} is the resolvent of C_i (goal clause) and another $C \in \{C_1, .., C_n\}$
- for every $j > n + 1$, C_j is the resolvent of C_{j-1} and another $C \in \{C_1, .., C_n\}$
- every resolution step takes the form
  $L' \vee C', \ \neg L'' \vee C'' \ \rightsquigarrow (C' \vee C'')(MGU(L', L''))$

Properties: SLD resolution is

- linear
- input
- directed
- ordered

LUSH resolution

The selection rule

In SLD, the rule requires the factor to be the first literal in both clauses

- as a consequence, the goal clause does not contain a non-negated literal and has to resolve with a clause whose first literal in non-negated

LUSH: Linear resolution with Unrestricted Selection for Horn clauses

- linear, input and directed but not ordered: every literal can be resolved with any other

LUSH resolution

Example: goal $C_7 : \neg C(x) \vee \neg E(x)$

C_1	:	$D(x) \vee \neg A(x)$	C_2	:	$E(x) \vee \neg A(x)$
C_3	:	$A(a)$	C_4	:	$B(a)$
C_5	:	$C(x) \vee \neg D(x) \vee \neg B(x)$	C_6	:	$B(x) \vee \neg D(x) \vee \neg C(x)$

SLD

C_7 with C_5 gives $\neg D(x) \vee \neg B(x) \vee \neg E(x)$

$\neg D(x) \vee \neg B(x) \vee \neg E(x)$ with C_1 gives $\neg A(x) \vee \neg B(x) \vee \neg E(x)$

$\neg A(x) \vee \neg B(x) \vee \neg E(x)$ with C_3 gives $\neg B(a) \vee \neg E(a)$

$\neg B(a) \vee \neg E(a)$ with C_4 gives $\neg E(a)$

$\neg E(a)$ with C_2 gives $\neg A(a)$

$\neg A(a)$ with C_3 gives $\square$

LUSH

C_7 with C_2 gives $\neg A(x) \vee \neg C(x)$

$\neg A(x) \vee \neg C(x)$ with C_3 gives $\neg C(a)$

$\neg C(a)$ with C_5 gives $\neg D(a) \vee \neg B(a)$

$\neg D(a) \vee \neg B(a)$ with C_1 gives $\neg A(a) \vee \neg B(a)$

$\neg A(a) \vee \neg B(a)$ with C_3 gives $\neg B(a)$

$\neg B(a)$ with C_4 gives $\square$

SLD resolution

Lemma

The support set of a set of Horn Clauses is satisfiable

Proof.

1. the clauses of the support set have a non-negated literal
2. an interpretation which assigns **t** to such literals makes the set true

Corollary

If there exists a refutation of a set of Horn clauses, then there exists a directed refutation on the support set

Lemma

If there exists a LUSH refutation of a set of Horn clauses, then there exists an SLD refutation of the same set

SLD resolution

Lemma

If there exists a refutation of a set H of Horn clauses, then there exists an SLD refutation of the same set

Proof.

1. there exists a refutation of H
2. there exists a directed refutation $\mathcal{R}$ (the support set is satisfiable)
   - every step involves a goal clause or an intermediate resolvent
3. $\mathcal{R}$ is an input refutation
   - every step requires a clause with a non-negated literal, i.e., a support clause
   - support clauses are input clauses
4. if there exists an input refutation, then there exists a linear input one $\mathcal{R}'$
   - $\mathcal{R}'$ is directed, input and linear, that is, LUSH
5. there exists an SLD refutation $\mathcal{R}''$ (lemma above)

SLD resolution

Theorem

SLD resolution is complete for Horn clauses: a set of Horn clauses is unsatisfiable iff there exists an SLD refutation for it

Proof.

$\rightarrow$ previous lemma

$\leftarrow$ trivial

When studying a set of Horn clauses

- possible refutations can be restricted to SLD refutations
- search trees can be restricted to SLD search trees for $\square$

SLD resolution

Depth and breadth

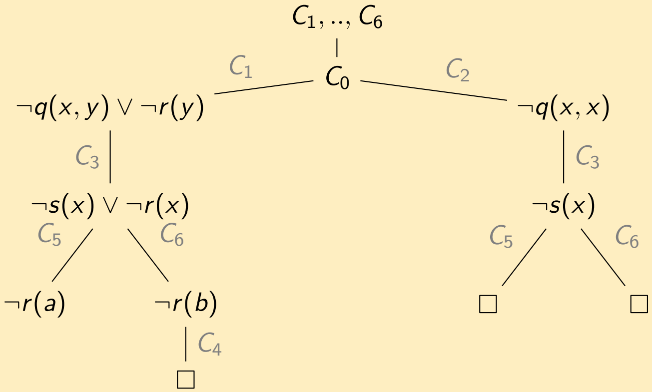
- breadth-first SLD is complete, depth-first is not
- in the depth-first approach, it is crucial how to choose the order for selecting support clauses to be resolved with the current goal clause
  - *computation function*
- depending on the search strategy
  - some refutations are not found
  - some derivations do not terminate

SLD resolution

Example

$C_1 : p(y) \vee \neg q(x,y) \vee \neg r(y)$ $\quad C_2 : p(x) \vee \neg q(x,x)$ $\quad C_3 : q(x,x) \vee \neg s(x)$

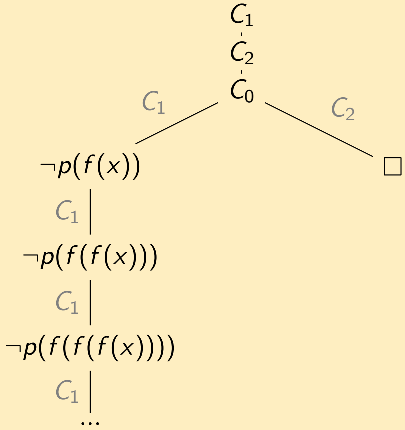
$C_4 : r(b)$ $\quad C_5 : s(a)$ $\quad C_6 : s(b)$ $\quad C_0 : \neg p(x)$



SLD resolution

Example: $C_1 : p(x) \vee \neg p(f(x))$, $C_2 : p(a)$, $C_0 : \neg p(y)$

- a depth search with a computation function which chooses the first support clause does not terminate



SLD resolution

Example: $C_1 : p(x) \vee \neg p(f(x))$, $C_2 : p(a)$, $C_0 : \neg p(y)$

- but a refutation can be obtained by changing the order of the support clauses (C_2 before C_1)