Semantic Resolution Lemmaizing and Contraction

Maria Paola Bonacina*

Jieh Hsiang**

* Dept. of CS - Univ. of Iowa

** Dept. of CSIE - Nat. Taiwan Univ.

/* Presented at the Dagstuhl Seminar 9512 on Deduction, Schloß Dagstuhl, Germany, March 1995 */

Semantic Resolution

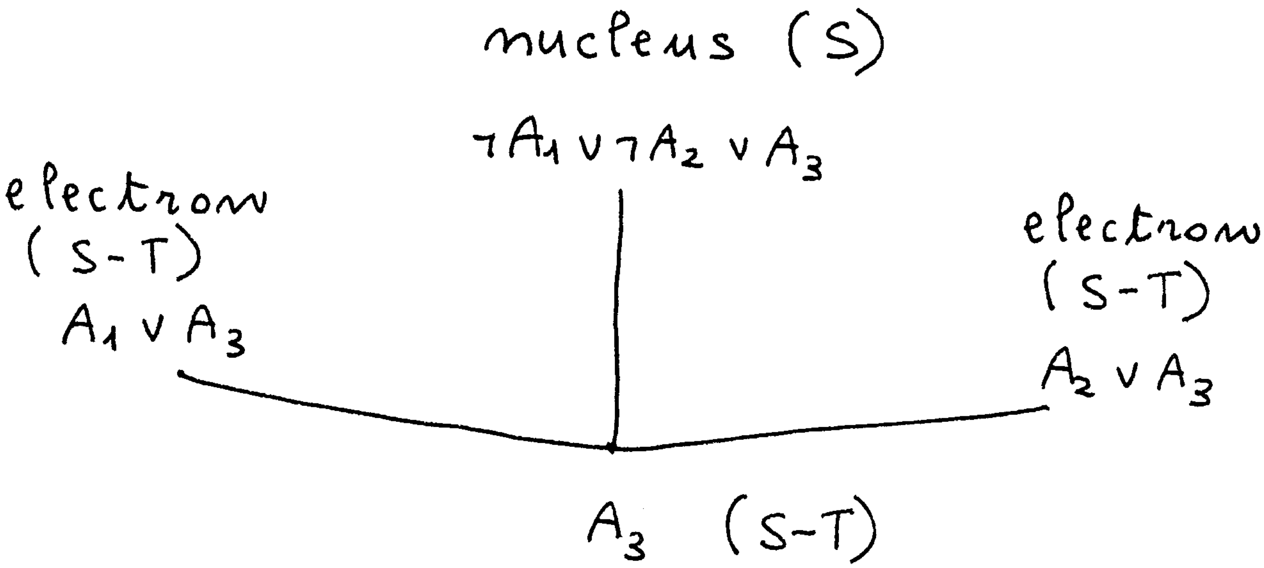
Set of clauses S

Prove S unsatisfiable

Consistent $T \subset S$ $\quad (I \models T)$

Do not expand T:

Example:



Do not generate intermediate resolvents that belong to T.

Semantic Resolution

- Hyperresolution — positive / negative

- Set of Support

  T : axioms

  S−T = SOS goals

$$(T; SOS_0) \vdash (T; SOS_1) \vdash \ldots\ldots$$

Forward / Backward Reasoning

- **Forward Reasoning:**
  generate consequences from the axioms.

- **Backward Reasoning:**
  generate subgoals from goals.

- Combination.

Forward / Backward Reasoning in Semantic Resolution

T : axioms in T, goals in $S-T$

Do not expand T $\Rightarrow$ backward reasoning

T : goals in T

Do not expand T $\Rightarrow$ forward reasoning

Lemmaizing in Semantic Resolution

Generation of lemmas:

retain selected lemmas in T

(relax in a controlled way the essential restriction of semantic resolution).

Semantic resolution does backward (forward) reasoning	$\Rightarrow$	Lemmaizing adds forward (backward) reasoning

Generation of unit lemmas

$(T_0 ; SOS_0) \vdash (T_1 ; SOS_1) \vdash \dots$

$\neg L \lor C$ in SOS

If $\neg L \lor C$ and T derive $C\sigma$
(without using SOS and C)
then $L\sigma$ is a lemma of T.

Example:

$$\neg L(y) \lor \neg S(x) \lor G(y,x)$$

$$L(a) \lor G(z, f(z))$$

$$G(z, f(z)) \lor \neg S(x) \lor G(a,x)$$

$$G(a,x) \lor \neg S(x)$$

$$\neg S(f(a)) \lor \neg S(x) \lor G(a,x)$$

$$S(f(a))$$

$$\neg S(x) \lor G(a,x)$$

$\boxed{L(a) \text{ lemma}}$

Generation of unit lemmas

$(T_0 ; SOS_0) \vdash (T_1 ; SOS_1) \vdash \ldots$

$\neg L \vee C$ in SOS

$L' \vee Q_1 \vee \ldots \vee Q_m$ in T

$(Q_1 \vee \ldots \vee Q_m \vee C)\rho$

$\ldots$

C not involved

all side-clauses in T

$\ldots$

$C\sigma$

$C\sigma$ is linearly derived from $\neg L \vee C$ by using T.

Lemma : $\boxed{L\sigma}$

Generation of unit lemmas

Meta-rule for unit lemmaizing:

if $C\sigma$ is linearly derived from $\neg L \vee C$ by using T, then add

$$\boxed{L\sigma}$$

to T.

Soundness: $T \models L\sigma$

Lemmaizing as Meta-level Reasoning

Lemmaizing is a meta-level inference rule (meta-rule) because it uses knowledge about a fragment of the derivation

- more than one inference step
- shape of the derivation
- ancestry relation

to make an inference.

Generation of non-unit lemmas

Example:

$\neg L \vee C$ in SOS

$L \vee P$ in T

$P \vee C$

$\neg P \vee Q \vee R$ in T

$Q \vee R \vee C$

$\neg Q$ in SOS

$R \vee C$

$\neg R$ in SOS

C

Lemma: $L \vee Q \vee R$

Generation of non-unit lemmas

Example:

$P \vee \neg Q$ in SOS

$\neg P \vee \neg Q$ in T

$\neg Q \vee \neg Q$

$\neg Q$

Lemma: $\neg P \vee \neg Q$

Generation of non-unit lemmas

$(T_0 ; SOS_0) \vdash (T_1 ; SOS_1) \vdash \ldots$

$\neg L \vee C$ in SOS

$\quad L' \vee Q_1 \vee \ldots \vee Q_m$ in T

$(Q_1 \vee \ldots \vee Q_m \vee C)\rho$

$\quad \ldots$

$\vdots$

$\quad \ldots$

$\vdots$

$C\sigma$

side-clauses from either T or SOS

$C\sigma$ is linearly derived from $\neg L \vee C$ by using T and SOS.

Lemma: $\boxed{(L \vee \text{"residue"})\sigma.}$

Generation of non-unit lemmas

Residue of $\neg L$ in T:

disjunction of the subgoals of $\neg L$ that cannot be solved by T (and are solved by SOS) in the linear derivation of $C\sigma$ from $\neg L \vee C$.

Generation of non-unit lemmas

Meta-rule for non-unit lemmaizing:

if $C\sigma$ is linearly derived from

$\neg L \vee C$ by using T and SOS,

then add lemma

$$\boxed{(L \vee \text{"residue"})\,\sigma}$$

to T.

Soundness : $T \models (L \vee \text{"residue"})\sigma$

An inference system with

- Resolution
- Factoring
- Lemmaizing
- Contraction

Lemmaizing and Contraction

(Unit) lemmas are useful for contraction:

- (unit) subsumption
- clausal simplification.

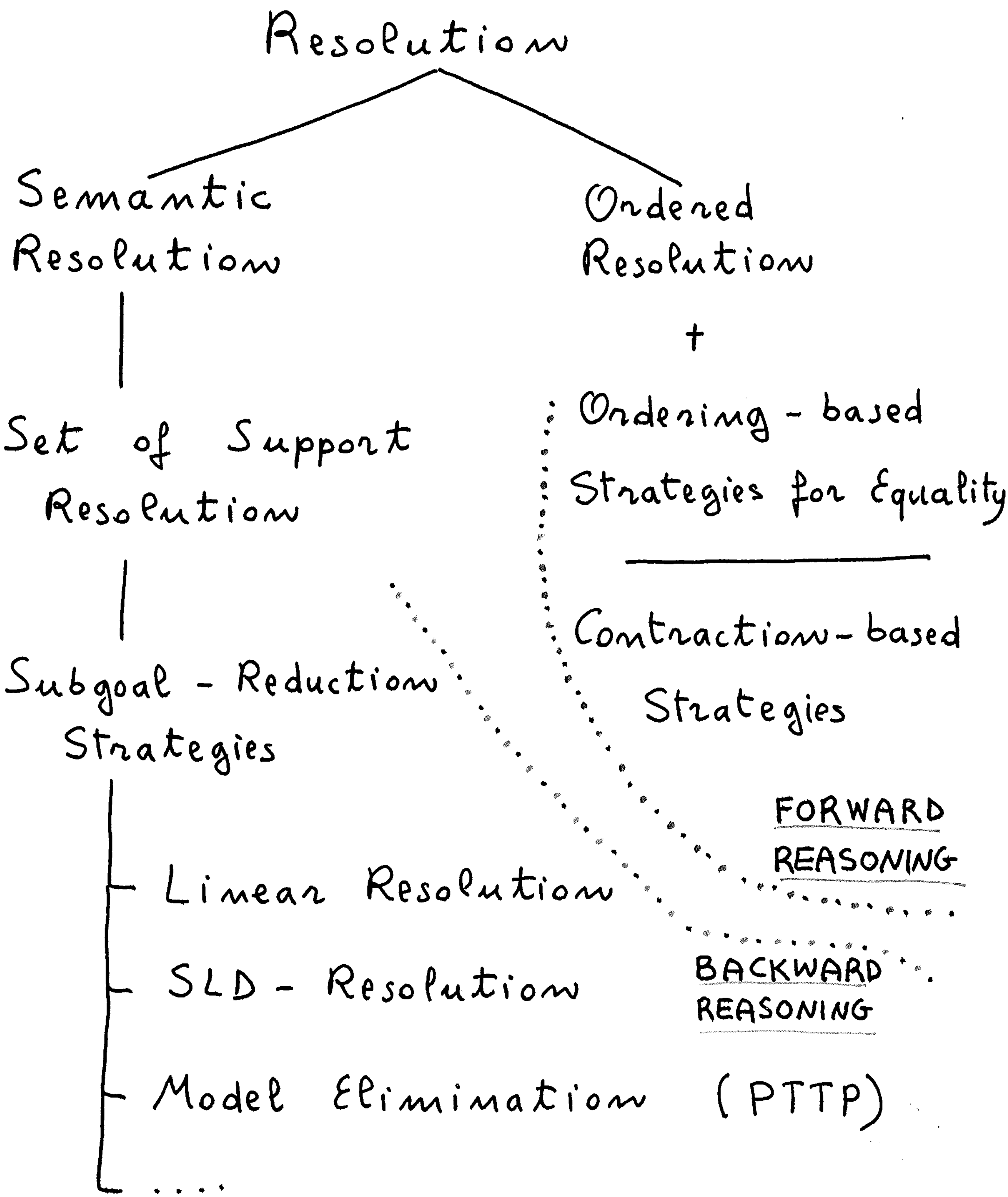
Since lemmaizing is added to an already complete strategy, it can be restricted, e.g. only unit lemmas.

Another contraction rule: Purity Rule for FOL

A literal is *pure* if it does not resolve with other literals (*it "fails"*).

Purity Rule: delete a clause if it contains a pure literal.

Instances of pure literals are pure: *caching of pure literals*.


Resolution
Semantic Resolution
Ordered Resolution
+
Ordering-based Strategies for Equality
Set of Support Resolution
Contraction-based Strategies
Subgoal-Reduction Strategies
FORWARD REASONING
Linear Resolution
SLD-Resolution
BACKWARD REASONING
Model Elimination (PTTP)
....

Model Elimination

[Loveland 1965, 1969, Stickel 1984, 1986...]

ME-extension ($\approx$ input resolution):

$$\neg L \vee C \qquad\qquad L' \vee Q \text{ in } T$$

$$(Q \vee [\neg L] \vee C)\sigma$$

ME-reduction ($\approx$ ancestor resolution):

$$\neg L \vee D \vee [L'] \vee C$$

$$(D \vee [L'] \vee C)\sigma$$

Key idea: represent locally (at the clause level) global knowledge (the ancestry relation).

Lemmaizing in ME

[Loveland 1969, Astrachan-Stickel 1992]

ME-contraction (with Lemmaizing):

$$\begin{array}{c} [\neg L] \vee C \\ | \\ C \end{array}$$

Lemma:

$\boxed{L \vee \text{"complements of needed ancestors"}}$

complements of needed ancestors $\equiv$ T-unsolved subgoals (residue)

Caching in ME (PTTP)

[Astrachan - Stickel 1992]

- Horn logic
- Store solved goals in cache
- Replace lemmaizing by caching
  search by table look-up

$A = A'\sigma$ (goal literals)

- Failure caching:

  A' failed $\Longrightarrow$ A fails

- Success caching:

  A' solved
  all solutions $\Longrightarrow$ all solutions of A are
  $A'\rho_1 \ldots A'\rho_n$ $\qquad$ $A'\rho_i$ instances of A

(Semantic) Resolution

- non-deterministic
- variable search plan
- forward / backward reasoning
- lemmaizing
- contraction: (cut search)

* subsumption
* purity deletion

ME-PTTP

- linear, selected literal
- DFID
- backward reasoning
- lemmaizing
- caching: (cut search)

* success caching
* failure caching

- Resolution [Robinson 1963]
- Hyperresolution [Robinson 1965]
- Set of Support [Wos 1965]
- Semantic Resolution [Slagle 1967]
- Linear Resolution
  [Loveland 1968] [Luckham 1968]
- SLD - Resolution
  [Kowalski - Kuehner 1971]
- Model Elimination
  [Loveland 1965, 1969]
- Simplified Problem Reduction Format
  [Plaisted 1982]
- Prolog Technology Theorem Proving
  [Stickel 1984, 1986]
- PTTP with Lemmaizing and Caching
  [Astrachan - Stickel 1992]

- Resolution [Robinson 1963]
- Ordered Resolution
  [Reiter 1971] [Slagle-Norton 1971]
- Locking [Boyer 1971]

  .....

- Ordered Resolution
  with Simplification Ordering
  [Dershowitz 1982]
  [Hsiang - Rusinowitch 1986, 1991]
  [Bachmair - Ganzinger 1990, ...]
  [Nieuwenhuis - Orejas - Rubio 1990, ...]
  [Dershowitz 1990]

  ....