On Semantic Resolution with Lemmaizing and Contraction

Maria Paola Bonacina
Dept. of Computer Science
The University of Iowa

joint work with

Jieh Hsiang
Dept. of Computer Science
National Taiwan University

Pacific Rim International Conference on Artificial Intelligence (PRICAI), Cairns, Australia, August 1996

Motivation

Combining features of forward reasoning and backward reasoning

Lemmaizing from backward reasoning to forward reasoning

Forward reasoning

Generate consequences from axioms + negated theorem

Keep database of clauses

Use indexing, search plans

Problem: too many redundant clauses

Answer: Contraction
Subsumption
Simplification
...

Contraction-based strategies

e.g. OTTER, RRL, REVEAL
...

Backward reasoning

Reduce goal to subgoals

Work on stack of goals

Search linear refutation by depth-first search with iterative deepening

Problem: too many repeated goals

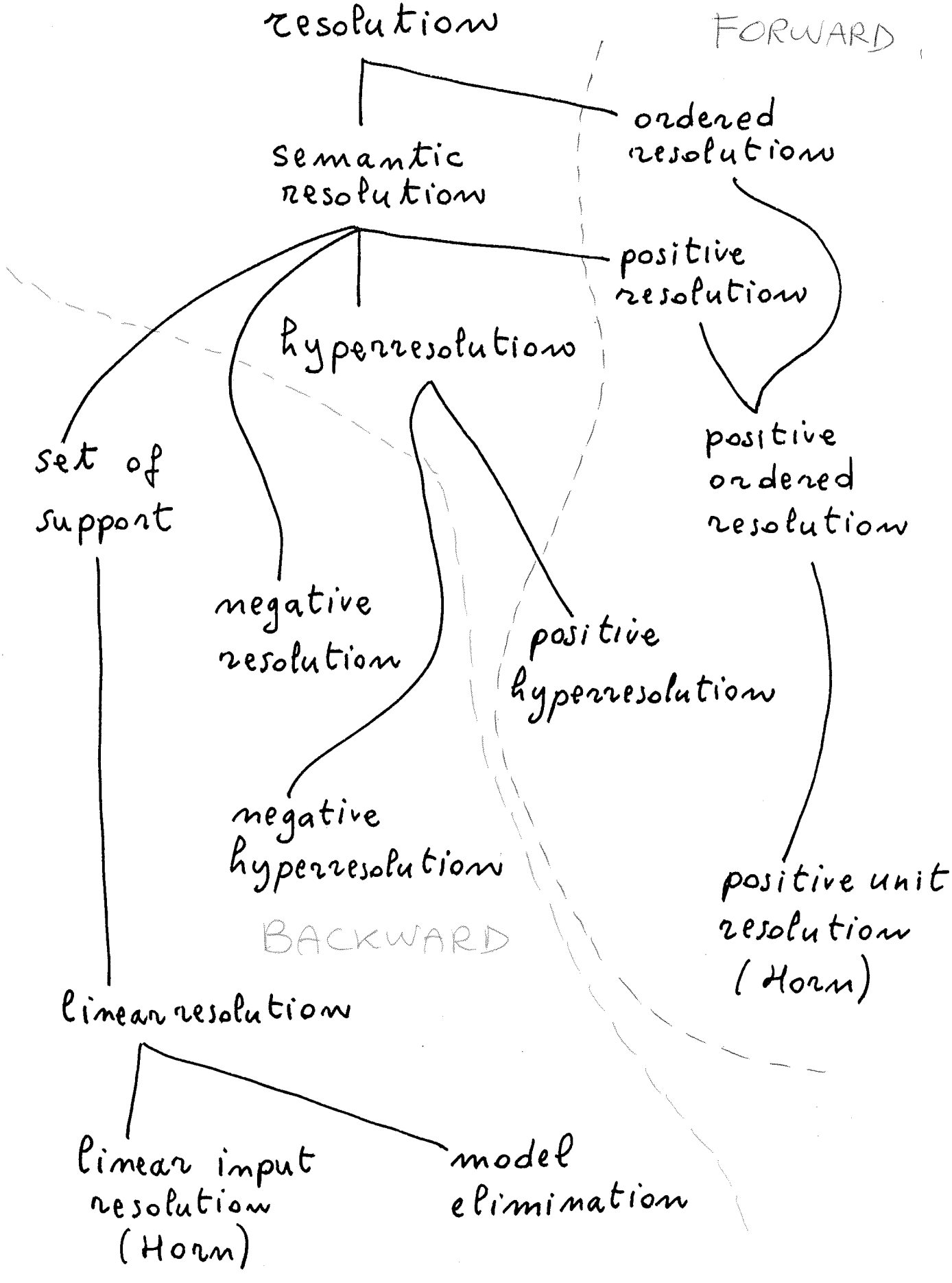
Answer: Lemmaizing / Caching

Subgoal-reduction strategies

e.g. PTTP, METEOR } model elimination

...

Prolog


resolution
FORWARD
ordered resolution
semantic resolution
positive resolution
hyperresolution
set of support
positive ordered resolution
negative resolution
positive hyperresolution
negative hyperresolution
BACKWARD
positive unit resolution (Horn)
linear resolution
linear input resolution (Horn)
model elimination

Synopsis

Lemmaizing

can be done in semantic resolution (ME: special case)

combines forward and backward reasoning

is meta-level reasoning

can coexist with contraction

Contraction

in semantic strategies

purity deletion

Semantic Resolution

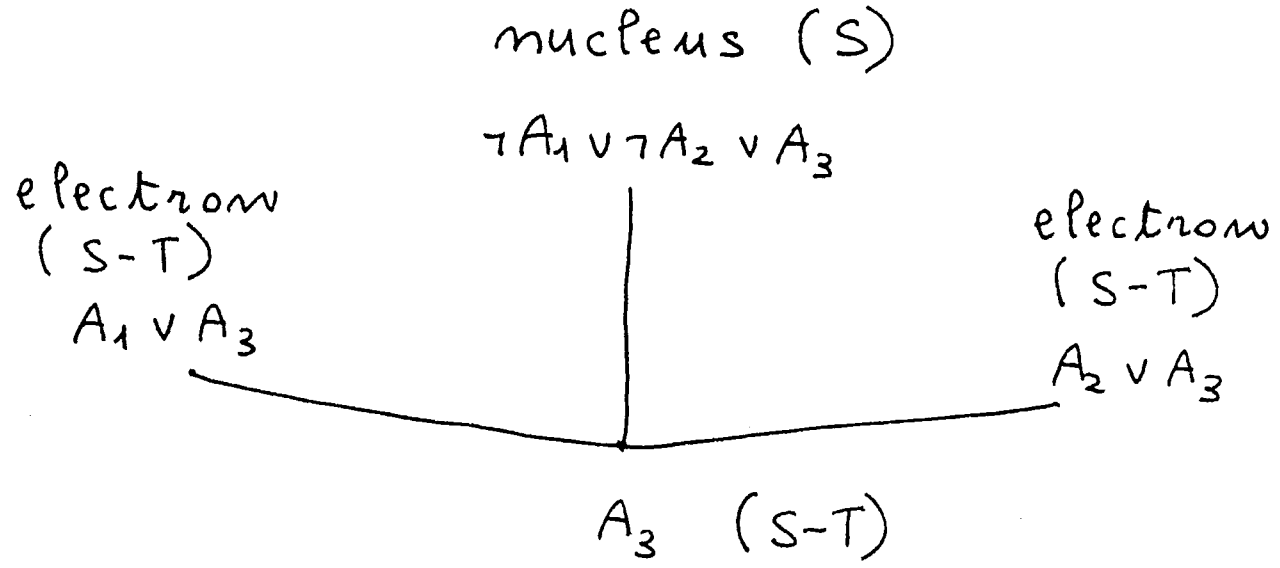
Set of clauses S

Prove S unsatisfiable

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

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 in semantic resolution

Axioms in T

Goals in SOS

Don't expand T $\Rightarrow$ backward reasoning

Goals in T

Axioms in SOS

Don't expand T $\Rightarrow$ forward reasoning

Generation of unit lemmas

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

$\neg L \vee C$ in SOS

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

Example:

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

$$\quad \llcorner \; L(a) \vee G(z, f(z))$$

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

$$\quad \llcorner \; G(a,x) \vee \neg S(x)$$

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

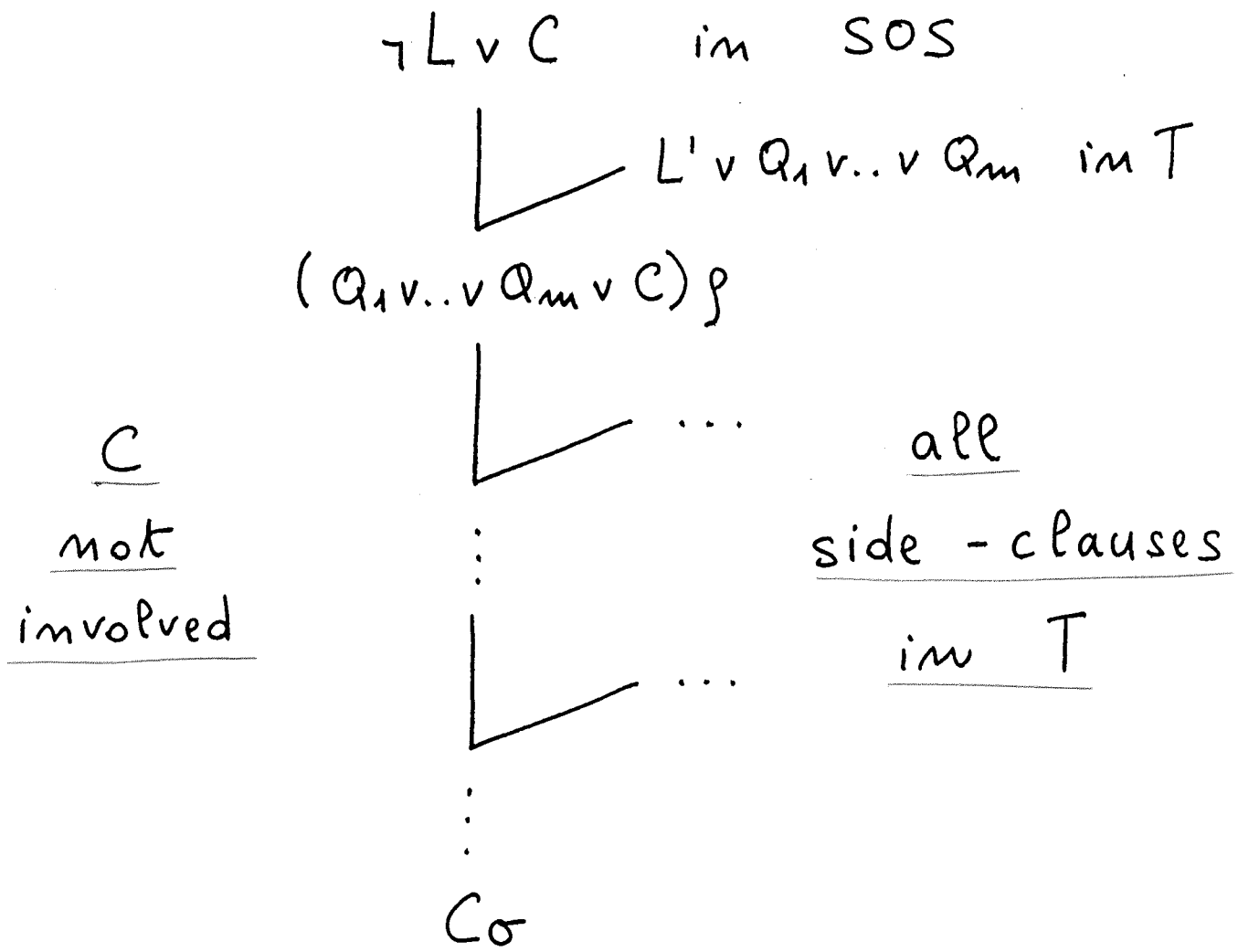
$$\quad \llcorner \; S(f(a))$$

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

$L(a)$ lemma

Generation of unit lemmas

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



Cσ is linearly derived from ¬L ∨ C by using T.

Lemma : Lσ

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

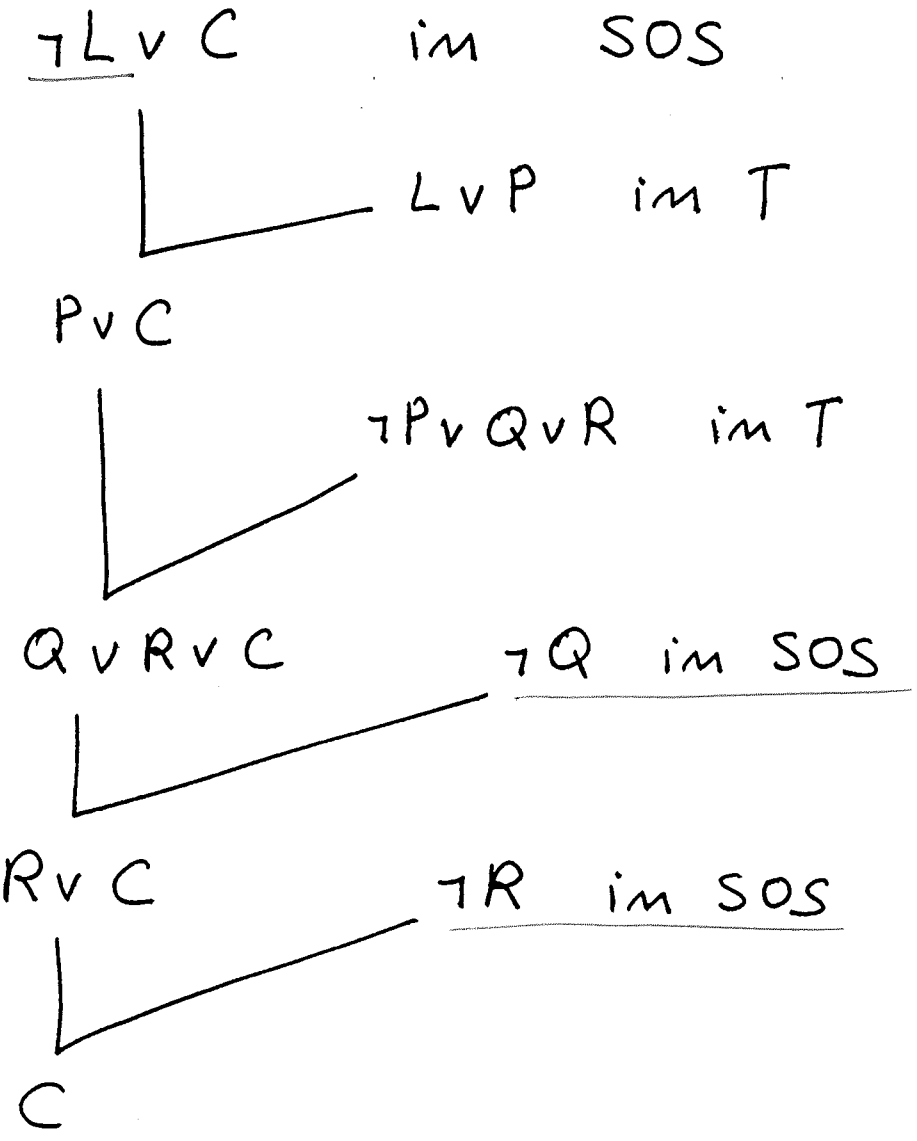
$$\boxed{L\sigma}$$

to T.

Soundness: $T \models L\sigma$

Generation of non-unit lemmas

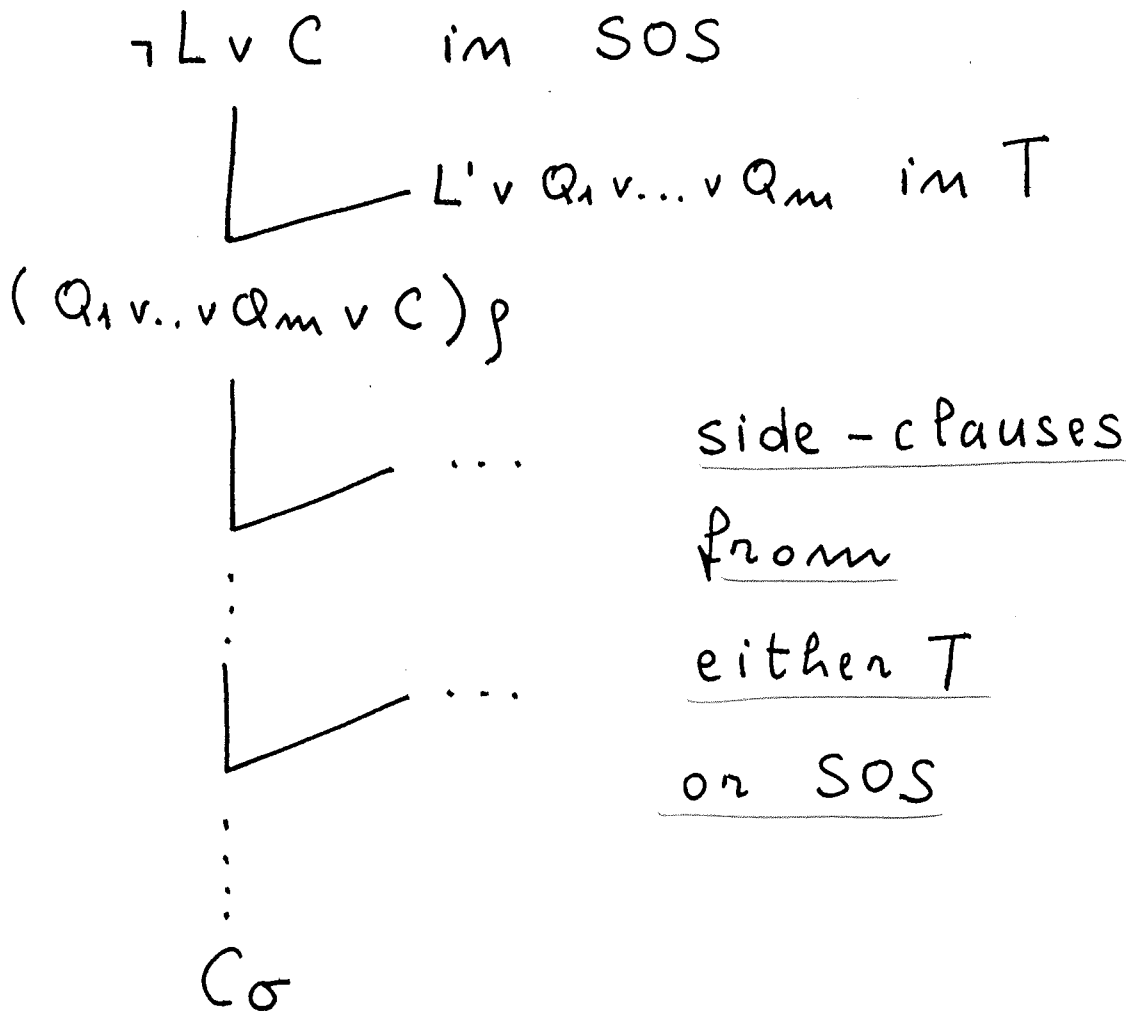
Example:



Lemma: $L \vee Q \vee R$

Generation of non-unit lemmas

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



Cσ is linearly derived from ¬L v C by using T and SOS.

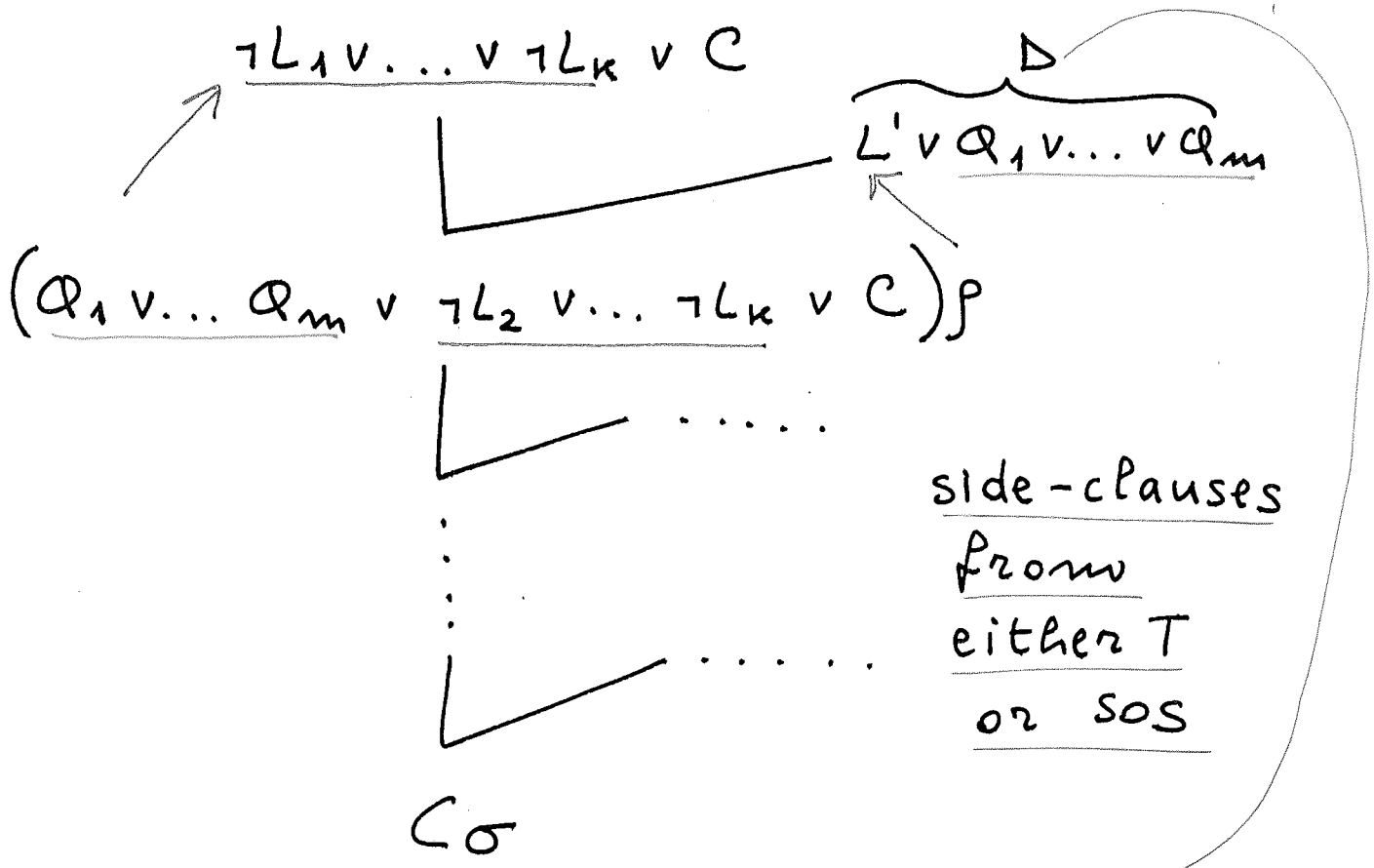
Lemma: (L v "residue")σ.

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 $\neg L \vee C$.

Residue



$$R_T(\neg L_1) = \begin{cases} \neg L_1 & \text{if } D \in SOS \\ \text{false} & \text{if } D \in T \wedge m = 0 \\ R_T(Q_1) \vee .. R_T(Q_m) & \text{if } D \in T \wedge m \geq 1 \\ \neg L_1 & \text{if factoring w. } C \\ R_T(\neg L_j) & \text{if factoring w. } \neg L_j \end{cases}$$

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

$$(L \vee \underbrace{\text{"residue"}}_{R_T(\neg L)})\sigma$$

to T.

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

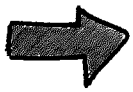
Lemmaizing in Semantic Resolution

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



Lemmaizing adds forward (backward) reasoning

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.

An inference system with

- Resolution
- Factoring
- Lemmaizing
- Contraction

The resolution rules

Resolution with lemma initiation:

$$\frac{\text{(T)}\quad \neg L' \vee D \qquad \overset{\text{SOS}}{L \vee C}}{(D_L \vee [\text{false}]_L \vee C)\sigma} \quad L\sigma = L'\sigma$$

Plain resolution:

$$\frac{\text{SOS}\quad L' \vee D \qquad \overset{\text{SOS}}{\neg L \vee C}}{(D \vee C)\sigma} \quad L'\sigma = L\sigma$$

Residue extension:

$$\frac{\text{SOS}\quad \neg L' \vee Q \qquad \overset{\text{SOS}}{P_L \vee D_L \vee C \vee [F]_L}}{(Q_L \vee D_L \vee C \vee [F \vee P]_L)\sigma} \quad P\sigma = L'\sigma$$

Subgoal elimination:

$$\frac{\text{(T)}\quad \neg L' \vee Q \qquad \overset{\text{SOS}}{P_L \vee D_L \vee C \vee [F]_L}}{(Q_L \vee D_L \vee C \vee [F]_L)\sigma} \quad P\sigma = L'\sigma$$

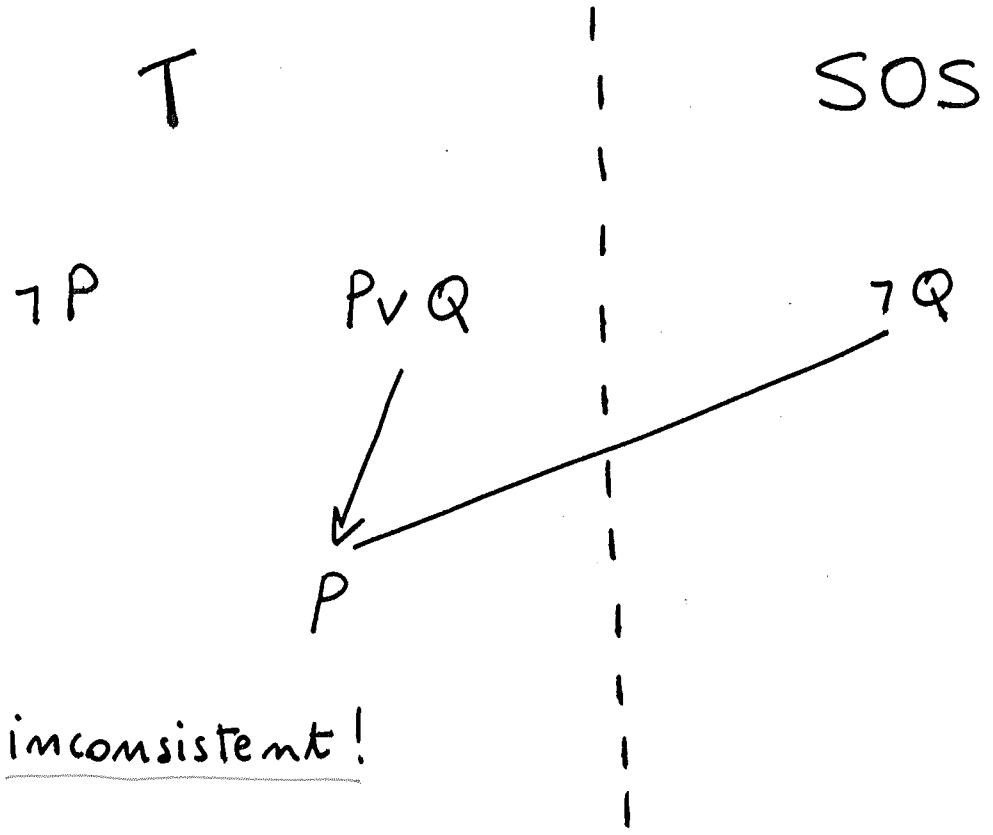
The lemma generation rule

$$\frac{(T\ ;\ SOS \cup \{[F]_L \vee C\})}{(T \cup \{\neg L \vee F\}\ ;\ SOS \cup \{C\})}$$

if C does not contain L-subgoals

F is $R_T(L)$

Contraction in semantic strategies



inconsistent!

Contraction in semantic strategies

(with lemmaizing)

Contraction of

SOS : stay in SOS
(may update residue)

T by T : stay in T

T by SOS : stay in T (true in J)
go to SOS (false in J)

Theorem: P_1 resolution system

P_1' semantic restriction

P_2 sound contraction

P_1' complete

$P_1 \cup P_2$ complete $\Rightarrow$ $P_1' \cup P_2$ complete

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 Deletion for FOL

A literal is pure if

- does not resolve with others
- resolves only with clauses containing pure literal

Instance of pure literal is pure

Purity Deletion: $\dfrac{S \cup \{C\}}{S}$ $A \in C$, A pure

Theorem: $A(\bar{t})$ pure in S

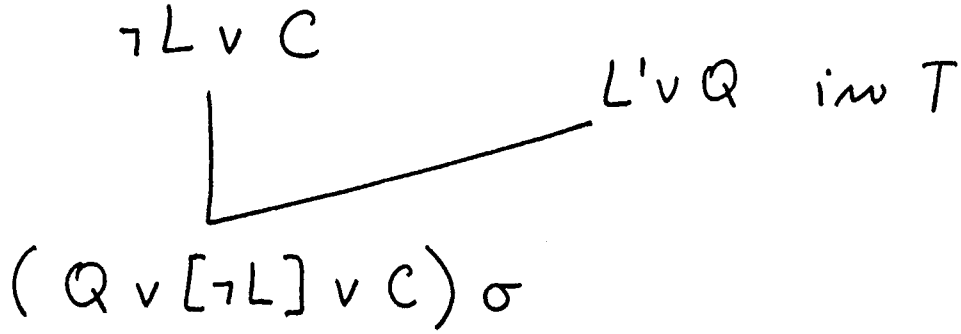
S unsatisfiable $\Rightarrow$

$S - \{C \mid A(\bar{t}) \in C\}$ unsatisfiable

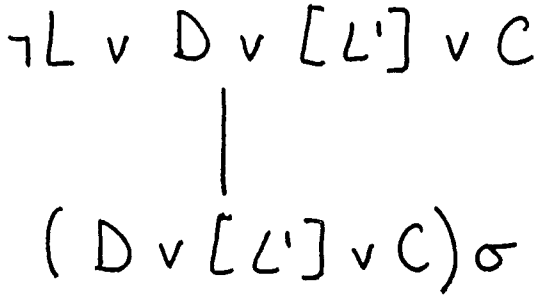
Model Elimination

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

ME-extension (≈ input resolution):



ME-reduction (≈ ancestor resolution):



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):

$$\frac{[\neg L] \vee C}{C}$$

Lemma:

L v "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 $\Rightarrow$ A fails

- Success caching:

  A' solved all solutions $\Rightarrow$ all solutions of A are $A'\rho_1 \dots A'\rho_n$ $A'\rho_i$ instances of A

Summary

Meta-rules for lemmaizing in semantic strategies

Lemmas are unsupported inferences

Inference rules for lemmaizing in resolution with set of support

Schemes for contraction in semantic strategies

Purity deletion in FOL

subsumption ~ success caching

purity deletion ~ failure caching

Unit lemmas enhance contraction

Cache subsumption

Future work

Implementation

Experiments

Criteria to control lemmaizing

e.g. unit
ground
weight or term-length
no nested function

Strategy analysis

analyze how lemmaizing may
reduce the search complexity
of strategies