Limits of CDCL Learning via Merge Resolution

Marc Vinyals

Waipapa Taumata Rau – University of Auckland

joint work with Noah Fleming, Vijay Ganesh, Antonina Kolokolova, and Ian Li

DPLL

$$y \vee z \quad y \vee \bar{z} \quad x \vee \bar{y} \vee z \quad x \vee \bar{y} \vee \bar{z} \quad \bar{x} \vee \bar{y}$$

Algorithm 1: DPLL
while *not solved* **do**
  if *conflict* **then** backtrack()
  else if *unit* **then** propagate()
  else branch()

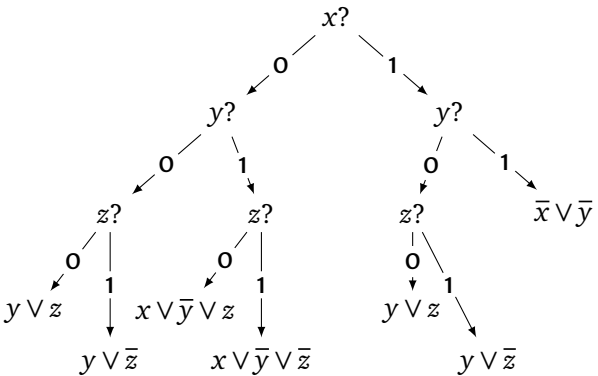
State: partial assignment

DPLL

$$y \vee z \quad y \vee \bar{z} \quad x \vee \bar{y} \vee z \quad x \vee \bar{y} \vee \bar{z} \quad \bar{x} \vee \bar{y}$$

Algorithm 1: DPLL
while *not solved* **do**
  if *conflict* **then** backtrack()
  else if *unit* **then** propagate()
  else branch()

State: partial assignment

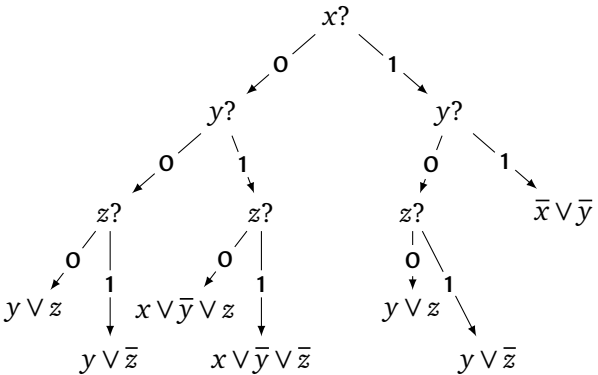


Resolution

$$y \vee z \quad y \vee \bar{z} \quad x \vee \bar{y} \vee z \quad x \vee \bar{y} \vee \bar{z} \quad \bar{x} \vee \bar{y}$$

- Search tree $\rightsquigarrow$ resolution proof

$$\frac{C \vee v \qquad D \vee \bar{v}}{C \vee D}$$



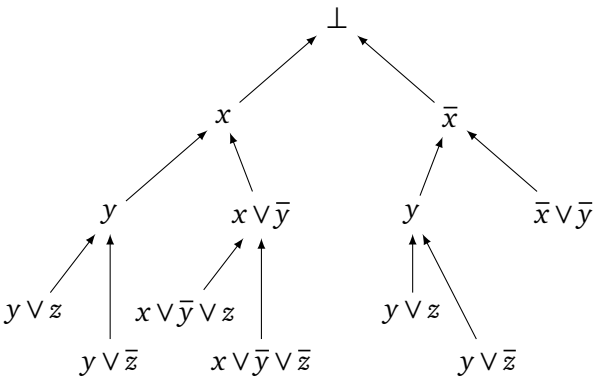
Resolution

$$y \vee z \quad y \vee \bar{z} \quad x \vee \bar{y} \vee z \quad x \vee \bar{y} \vee \bar{z} \quad \bar{x} \vee \bar{y}$$

- Search tree $\rightsquigarrow$ resolution proof

$$\frac{C \vee v \qquad D \vee \bar{v}}{C \vee D}$$

- Resolution lower bounds $\Longrightarrow$ DPLL lower bounds

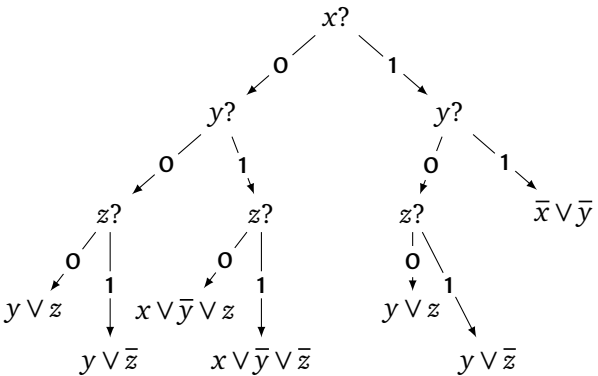


DPLL

$y \vee z \quad y \vee \bar{z} \quad x \vee \bar{y} \vee z \quad x \vee \bar{y} \vee \bar{z} \quad \bar{x} \vee \bar{y}$

Algorithm 1: DPLL
while *not solved* **do**
  if *conflict* **then** backtrack()
  else if *unit* **then** propagate()
  else branch()

State: partial assignment

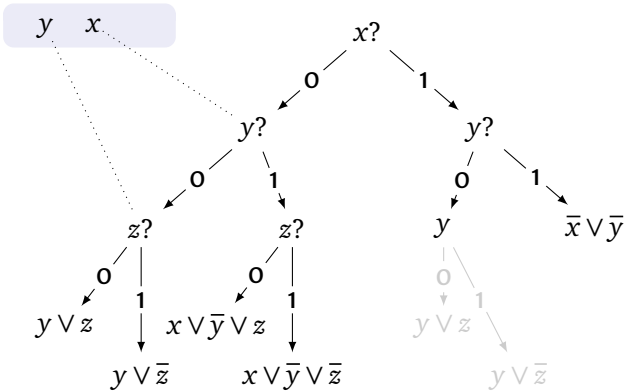


CDCL

$y \vee z \quad y \vee \bar{z} \quad x \vee \bar{y} \vee z \quad x \vee \bar{y} \vee \bar{z} \quad \bar{x} \vee \bar{y}$

Algorithm 2: CDCL
while *not solved* **do**
  if *conflict* **then** learn()
  else if *unit* **then** propagate()
  else
    maybe forget()
    maybe restart()
    branch()

State: partial assignment
& learned clauses

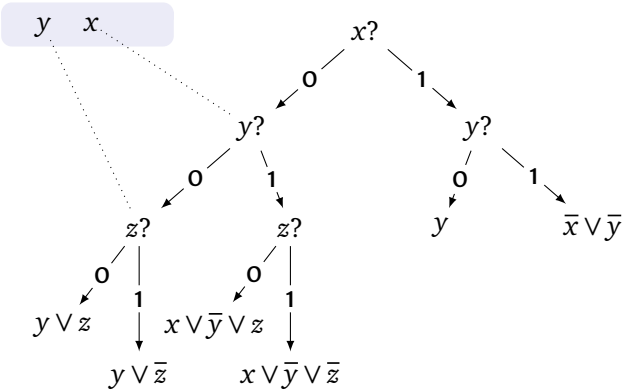


Resolution

$$y \vee z \quad y \vee \bar{z} \quad x \vee \bar{y} \vee z \quad x \vee \bar{y} \vee \bar{z} \quad \bar{x} \vee \bar{y}$$

- Search tree $\rightsquigarrow$ resolution proof

$$\frac{C \vee v \qquad D \vee \bar{v}}{C \vee D}$$



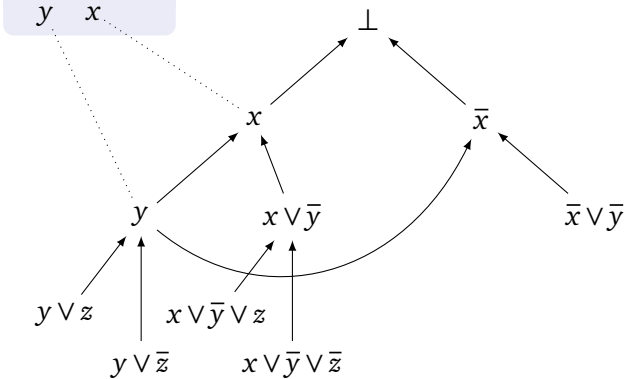
Resolution

$$y \vee z \quad y \vee \bar{z} \quad x \vee \bar{y} \vee z \quad x \vee \bar{y} \vee \bar{z} \quad \bar{x} \vee \bar{y}$$

- Search tree $\rightsquigarrow$ resolution proof

$$\frac{C \vee v \qquad D \vee \bar{v}}{C \vee D}$$

- Resolution lower bounds $\Longrightarrow$ CDCL lower bounds



CDCL vs Resolution

- CDCL proofs are in (general) resolution form
- DPLL proofs are in weaker "tree-like" form
  - There are formulas with polynomial resolution proofs but all tree-like proofs are exponential
- Is CDCL as powerful as general resolution?

CDCL vs Resolution

- CDCL proofs are in (general) resolution form
- DPLL proofs are in weaker "tree-like" form
  - There are formulas with polynomial resolution proofs but all tree-like proofs are exponential
- Is CDCL as powerful as general resolution?

- Partial results in 2000s

[Beame, Kautz, Sabharwal '04]
[Van Gelder '05]
[Hertel, Bacchus, Pitassi, Van Gelder '08]
[Buss, Hoffmann, Johannsen '08]

CDCL vs Resolution

- CDCL proofs are in (general) resolution form
- DPLL proofs are in weaker "tree-like" form
  - There are formulas with polynomial resolution proofs but all tree-like proofs are exponential
- Is CDCL as powerful as general resolution?

- Partial results in 2000s

  [Beame, Kautz, Sabharwal '04]
  [Van Gelder '05]
  [Hertel, Bacchus, Pitassi, Van Gelder '08]
  [Buss, Hoffmann, Johannsen '08]

- Yes (under natural model)

  [Pipatsrisawat, Darwiche '09]
  [Atserias, Fichte, Thurley '09]
  [Beyersdorff, Böhm '21]

CDCL equivalent to Resolution: Statement

Theorem [Pipatsrisawat, Darwiche '09]

With **non-deterministic** variable decisions,
CDCL can efficiently ~~find~~ reproduce resolution proofs

Theorem [Atserias, Fichte, Thurley '09]

With **random** variable decisions,
CDCL can efficiently find **bounded-width** resolution proofs

Simulation Overhead

- Given formula F and resolution proof of length L, CDCL can reproduce proof in $O(n^4\ L)$ steps.

[Pipatsrisawat, Darwiche '09]
[Atserias, Fichte, Thurley '09]

Simulation Overhead

- Given formula F and resolution proof of length L, CDCL can reproduce proof in $O(n^4\ L)$ steps. [Pipatsrisawat, Darwiche '09] [Atserias, Fichte, Thurley '09]
- $O(n^3\ L)$ [Beyersdorff, Böhm '21]

Simulation Overhead

- Given formula F and resolution proof of length L, CDCL can reproduce proof in $O(n^4\ L)$ steps. [Pipatsrisawat, Darwiche '09] [Atserias, Fichte, Thurley '09]
- $O(n^3\ L)$ [Beyersdorff, Böhm '21]

- Theory: Polynomial ☺
- Practice: But my solver runs in linear time ☹

Simulation Overhead

- Given formula F and resolution proof of length L, CDCL can reproduce proof in $O(n^4\ L)$ steps. [Pipatsrisawat, Darwiche '09] [Atserias, Fichte, Thurley '09]
- $O(n^3\ L)$ [Beyersdorff, Böhm '21]

- Theory: Polynomial ☺
- Practice: But my solver runs in linear time ☹

- Can we simulate resolution with less overhead?
- If not, why?

Simulation Overhead

Theorem [Fleming, Ganesh, Kolokolova, Li, V]

CDCL needs linear overhead to reproduce resolutions proofs

- Exist formulas with $O(n)$ resolution proofs that require $\Omega(n^2)$ steps in CDCL.

Simulation Overhead

Theorem [Fleming, Ganesh, Kolokolova, Li, V]

CDCL needs linear overhead to reproduce resolutions proofs

- Exist formulas with $O(n)$ resolution proofs that require $\Omega(n^2)$ steps in CDCL.

- Clauses learned by CDCL have syntactical restrictions
- Define restricted resolution
- Prove separation between restricted and general resolution

CDCL equivalent to Resolution: Simulation

- Derivation $\pi = C_1, \ldots, C_t$.
- Goal: learn every clause $C_i \in \pi$.

CDCL equivalent to Resolution: Simulation

- Derivation $\pi = C_1, \ldots, C_t$.
- Goal: learn every clause $C_i \in \pi$.

Algorithm 3: Simulation

for $C_i \in \pi$ **do**
  while C_i *not learned* **do**
    if *conflict* **then**
      learn()
      restart()
    else if *unit* **then** propagate()
    else assign a literal in C_i to false

CDCL equivalent to Resolution: Simulation

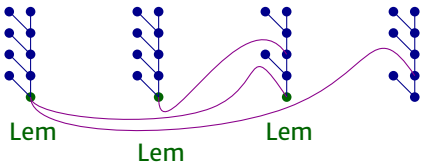
- Derivation $\pi = C_1, \ldots, C_t$.
- Goal: ~~learn~~ absorb every clause $C_i \in \pi$.
- C **absorbed** if learning C does not enable more unit propagations.

Algorithm 3: Simulation

```
for C_i ∈ π do
    while C_i not absorbed do
        if conflict then
            learn()
            restart()
        else if unit then propagate()
        else assign a literal in C_i to false
```

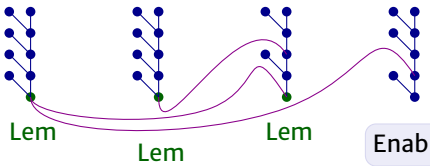
Formalizing CDCL

- Every resolution proof can be decomposed into a sequence of input resolution derivations.
- The final clause of each derivation is called a lemma, and can be used in future derivations.



Formalizing CDCL

- Every resolution proof can be decomposed into a sequence of input resolution derivations.
- The final clause of each derivation is called a lemma, and can be used in future derivations.

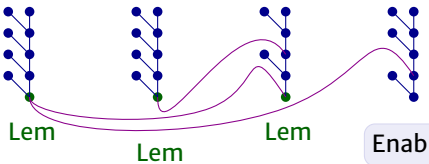


- Natural restriction: all lemmas must be 1-empowering

Formalizing CDCL

- Every resolution proof can be decomposed into a sequence of input resolution derivations.
- The final clause of each derivation is called a lemma, and can be used in future derivations.

A 1-empowering clause contains a merge in its derivation



Enable more propagations

- Natural restriction: all lemmas must be 1-empowering
- Finer restriction: all lemmas must follow merges

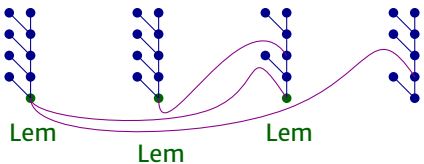
Premises share a literal: $\dfrac{x \vee y \vee z \qquad x \vee \bar{z}}{x \vee y}$

Merge Resolution

Building on [Andrews '68]

Definition

- Sequence of input resolution derivations
- Lemmas (reusable clauses) follow merges

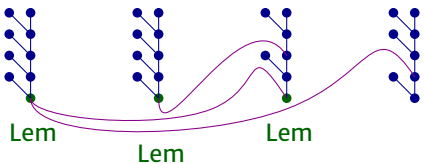


Merge Resolution

Building on [Andrews '68]

Definition

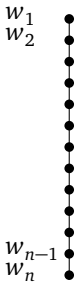
- Sequence of input resolution derivations
- Lemmas (reusable clauses) follow merges



Properties

- CDCL produces merge resolution proofs.
- Merge resolution simulates resolution with $O(n)$ overhead.
- Exist formulas with $O(n)$ resolution proofs that require $\Omega(n^2)$ merge resolution proofs.

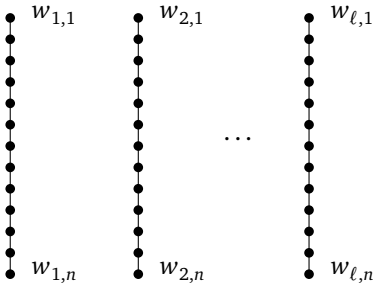
Tricky Formulas for Merge Resolution



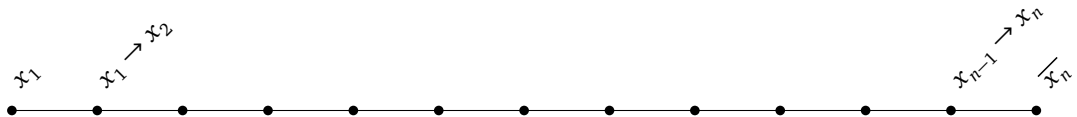
$$(w_1 = w_2) \equiv (w_1 \vee \overline{w_2}) \wedge (\overline{w_1} \vee w_2)$$

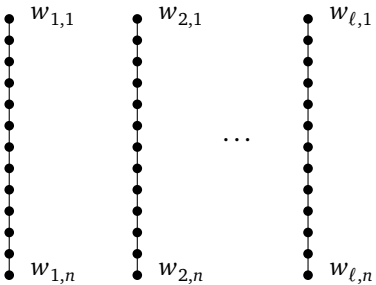
$$(w_{n-1} = w_n) \equiv (w_{n-1} \vee \overline{w_n}) \wedge (\overline{w_{n-1}} \vee w_n)$$

Tricky Formulas for Merge Resolution

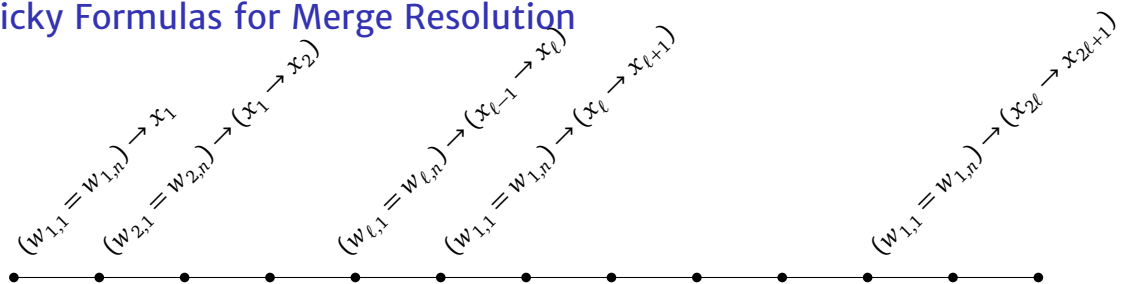


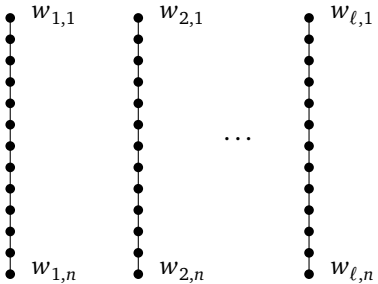
Tricky Formulas for Merge Resolution



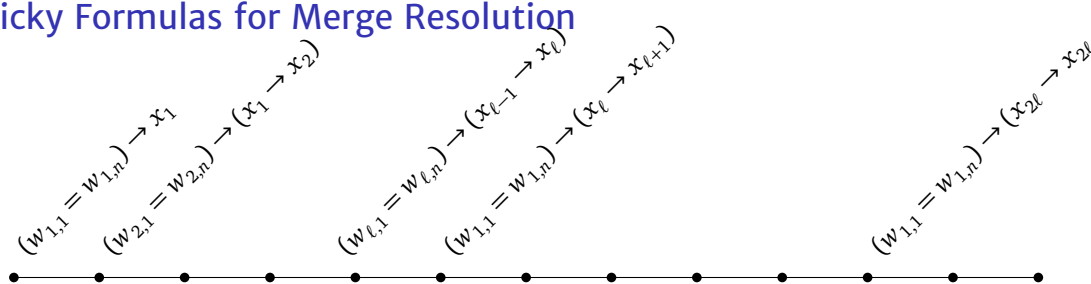


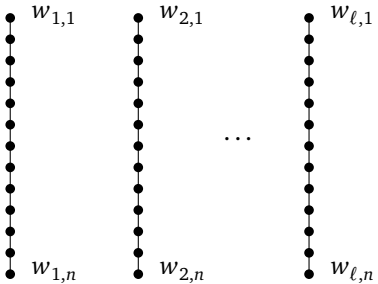
Tricky Formulas for Merge Resolution





Tricky Formulas for Merge Resolution

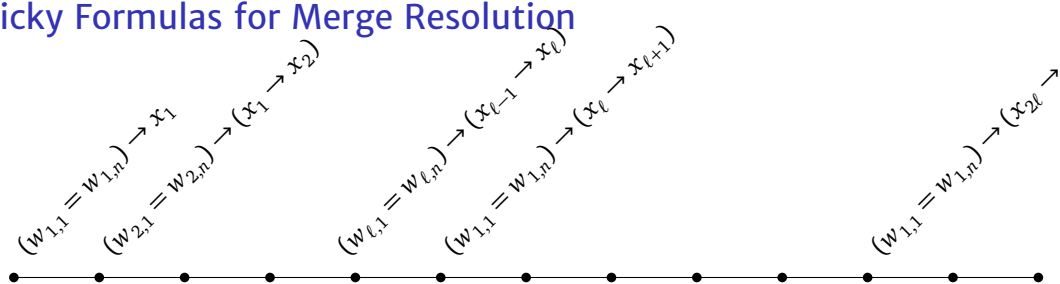


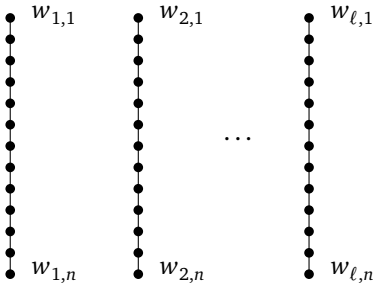


Formula Description

- $w_{i,j} = w_{i,j+1}$ for $i \in [\ell], j \in [n-1]$
- $(w_{\hat{\imath},j} = w_{\hat{\imath},j+1}) \to (x_i \to x_{i+1})$ for $i \in [n]$
  - ℓ is $\approx \log n$
  - $\hat{\imath}$ means $i \pmod{\ell}$
  - Expanded in CNF
  - + Boundary constraints

Tricky Formulas for Merge Resolution

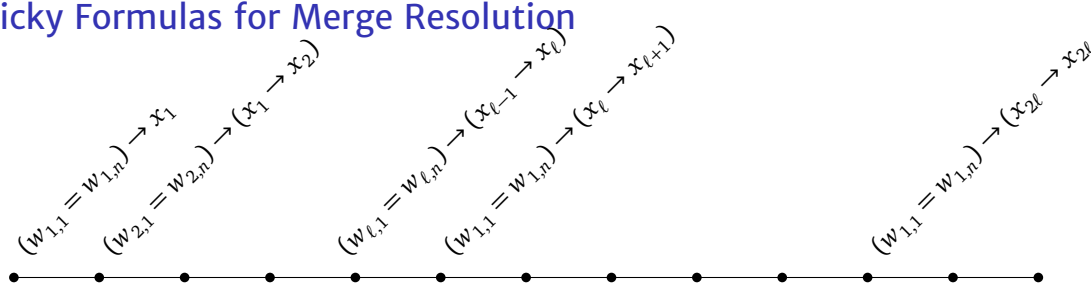


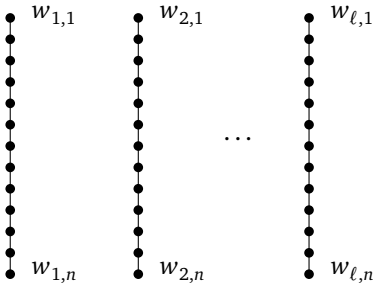


Short Resolution Proof

1. Derive $w_{i,1} = w_{i,n}$ for all $i \in [\ell]$
2. Simplify chain of implications
3. Derive contradiction

Tricky Formulas for Merge Resolution





Short Resolution Proof

1. Derive $w_{i,1} = w_{i,n}$ for all $i \in [\ell]$
2. Simplify chain of implications
3. Derive contradiction

Impossibility (Intuition)

- CDCL cannot remember $w_{i,1} = w_{i,n}$
- Must rederive $w_{i,1} = w_{i,n}$ for all $i \in [n]$

Open Problems

Overhead

- One n explained, n^2 remaining.
- Are merge resolution proofs easier to simulate by CDCL?
- Can we improve learning to avoid overhead?

Open Problems

Overhead

- One n explained, n^2 remaining.
- Are merge resolution proofs easier to simulate by CDCL?
- Can we improve learning to avoid overhead?

Assumptions

- Branching
- Memory
- Restarts

Take Home

- CDCL needs linear overhead to simulate resolution.

Open Problems

- Improve or explain remaining overhead.
- Improve learning.

Thanks!