Theoretical limits of 1UIP Learning

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()
```

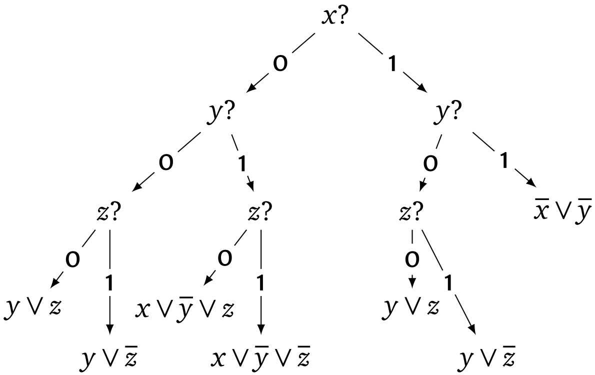
State: partial assignment

DPLL

$y \lor z \quad y \lor \overline{z} \quad x \lor \overline{y} \lor z \quad x \lor \overline{y} \lor \overline{z} \quad \overline{x} \lor \overline{y}$

Algorithm 1: DPLL

while *not solved* **do**
  if *conflict* **then** backtrack()
  else if *unit* **then** propagate()
  else branch()



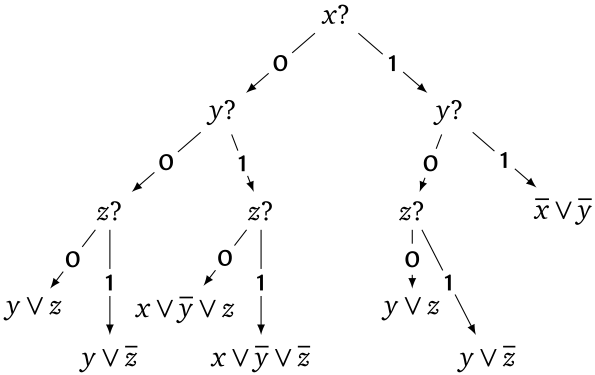
State: partial assignment

Resolution

- Search tree $\rightsquigarrow$ resolution proof

$$\frac{C \lor v \qquad D \lor \overline{v}}{C \lor D}$$

$y \lor z \quad y \lor \overline{z} \quad x \lor \overline{y} \lor z \quad x \lor \overline{y} \lor \overline{z} \quad \overline{x} \lor \overline{y}$



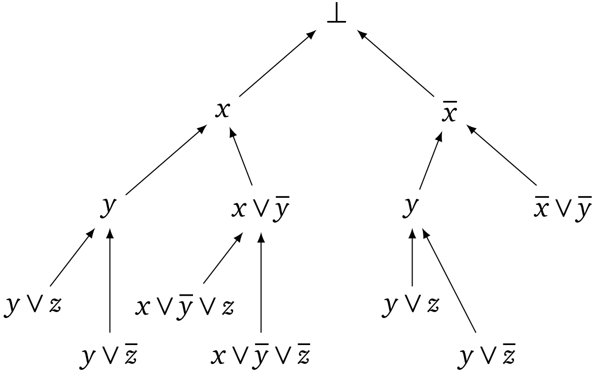
Resolution

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

- Search tree $\rightsquigarrow$ resolution proof

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

- Resolution lower bounds $\Longrightarrow$ DPLL lower bounds



DPLL

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

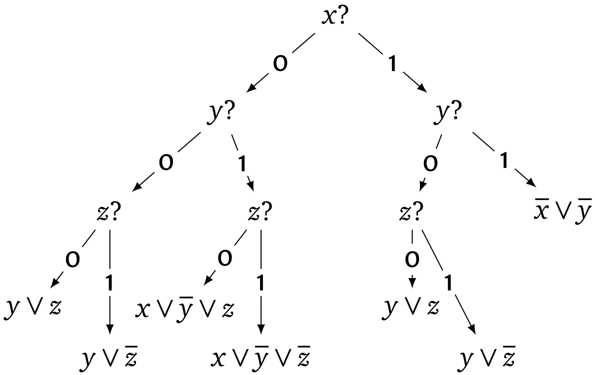
Algorithm 1: DPLL

while *not solved* **do**

  if *conflict* **then** backtrack()

  else if *unit* **then** propagate()

  else branch()



State: partial assignment

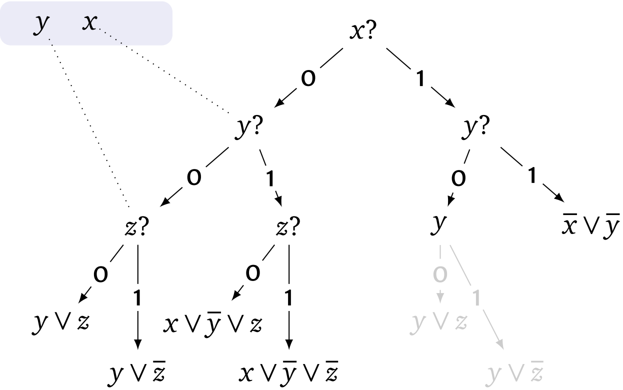
CDCL

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

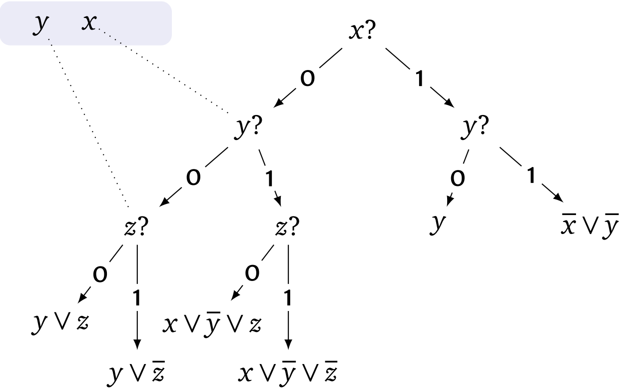
$y \lor z$ $y \lor \bar{z}$ $x \lor \bar{y} \lor z$ $x \lor \bar{y} \lor \bar{z}$ $\bar{x} \lor \bar{y}$



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



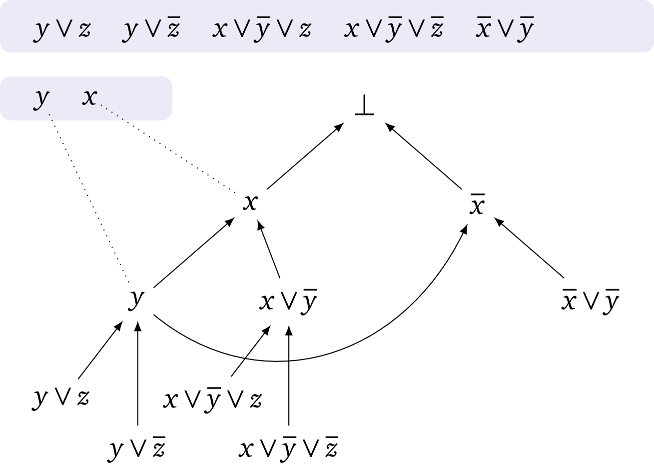
$$\frac{C \vee v \qquad D \vee \bar{v}}{C \vee D}$$

Resolution

- Search tree $\rightsquigarrow$ resolution proof

$$\frac{C \lor v \qquad D \lor \overline{v}}{C \lor D}$$

- 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 resolution proofs

Theorem [Atserias, Fichte, Thurley '09]

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

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

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
```

CDCL equivalent to Resolution: Assumptions

```
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
    restart()
```

- Optimal variable choices
- Clauses not thrown away
- Frequent restarts
- Standard learning

CDCL equivalent to Resolution: Assumptions

```
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
    restart()
```

- Optimal variable choices
- Clauses not thrown away
- Frequent restarts
- Standard learning

CDCL equivalent to Resolution: Assumptions

```
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
    restart()
```

- Optimal variable choices
- Clauses not thrown away
- Frequent restarts
- Standard learning

CDCL equivalent to Resolution: Assumptions

```
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
    restart()
```

- Optimal variable choices
- Clauses not thrown away
- Frequent restarts
- Standard learning

CDCL equivalent to Resolution: Assumptions

```
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
    restart()
```

- Optimal variable choices
- Clauses not thrown away
- Frequent restarts
- Standard learning

Assumptions: Branching

Need optimal variable choices.

- No deterministic algorithm simulates resolution unless FPT hierarchy collapses.
  [Alekhnovich, Razborov '01]
- No deterministic algorithm simulates resolution unless P = NP.
  [Atserias, Müller '19]

Assumptions: Branching

Need optimal variable choices.

- No deterministic algorithm simulates resolution unless FPT hierarchy collapses.
  [Alekhnovich, Razborov '01]
- No deterministic algorithm simulates resolution unless $P = NP$.
  [Atserias, Müller '19]

- CDCL with any static order exponentially worse than resolution.
  [Mull, Pang, Razborov '19]
- CDCL with VSIDS and similar heuristics exponentially worse than resolution.
  [V '20]

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 $\mathrm{O}(n^4\,L)$ steps. [Pipatsrisawat, Darwiche '09]
  [Atserias, Fichte, Thurley '09]
- $\mathrm{O}(n^3\,L)$ [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

Need linear overhead.

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

  [Fleming, Ganesh, Kolokolova, Li, V]

Simulation Overhead

Need linear overhead.

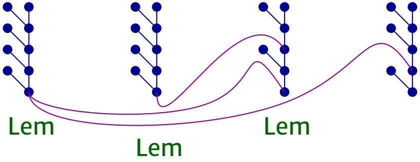
- Exist formulas with $O(n)$ resolution proofs that require $\Omega(n^2)$ steps in CDCL.

  [Fleming, Ganesh, Kolokolova, Li, V]

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

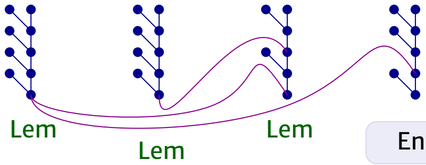
Formalizing CDCL

- 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.



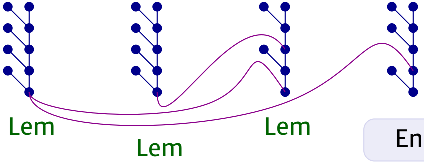
Enable more propagations

- 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 be merges

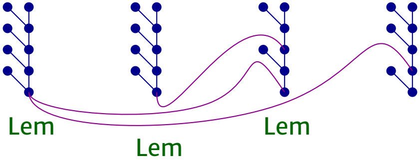
Premises share a literal: $\dfrac{x \lor y \lor z \qquad x \lor \overline{z}}{x \lor y}$

Merge Resolution

Building on [Andrews '68]

Definition

- Sequence of input resolution derivations
- Lemmas (reusable clauses) are 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.

Future: CDCL vs Resolution

Overhead

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

Future: CDCL vs Resolution

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

Future: SAT vs Proof Complexity

Beyond Resolution

- Preprocessing
- Parity and pseudoBoolean constraints
- Symmetry breaking

Future: SAT vs Proof Complexity

Beyond Resolution

- Preprocessing
- Parity and pseudoBoolean constraints
- Symmetry breaking

Beyond Proofs?

- Satisfiable formulas
- When is CDCL efficient?

Take Home

- CDCL needs linear overhead to simulate resolution.

Open Problems

- Improve or explain remaining overhead.
- Assumptions needed for simulation.
- Proof systems beyond resolution.
- ...

Thanks!