Resolution: A method for proving that a CNF formula is unsatisfiable

Derive new clauses from old using:

Resolution rule:

\[ C_1 \lor x, \quad C_2 \lor \neg x \quad \frac{}{C_1 \lor C_2} \]

Goal: derive empty clause \( \Lambda \)

Resolution rule is sound

Derivation of \( \Lambda \) certifies unsatisfiability
Last Time — Resolution

**Resolution**: A method for proving that a CNF formula is **unsatisfiable**

Derive new clauses from old using:

→ **Resolution rule**: \[
\frac{C_1 \lor x, \ C_2 \lor \neg x}{C_1 \lor C_2}
\]

**Goal**: derive empty clause \(\Lambda\)

Resolution rule is **sound**

\[\quad (x_2 \lor x_3) \land (\neg x_1 \lor \neg x_3) \land (\neg x_2) \land (x_1 \lor \neg x_3)\]

\[\quad \neg x_3 \quad (\neg x_2) \quad (x_2 \lor x_3) \quad (x_1 \lor \neg x_3) \quad (\neg x_1 \lor \neg x_3)\]

\[\quad \Lambda\]

\[\quad x_2\]
Last Time — Resolution

**Resolution**: A method for proving that a CNF formula is **unsatisfiable**

Derive new clauses from old using:

→ **Resolution rule**: 

\[
\frac{C_1 \lor x, \quad C_2 \lor \neg x}{C_1 \lor C_2}
\]

**Goal**: derive empty clause \( \Lambda \)

Resolution rule is **sound**

\[ \rightarrow \text{ Derivation of } \Lambda \text{ certifies unsatisfiability} \]

\[(x_2 \lor x_3) \land \neg x_1 \lor \neg x_3 \land \neg x_2 \land (x_1 \lor \neg x_3)\]
Resolution: A method for proving that a CNF formula is unsatisfiable

Derive new clauses from old using:

\[ \rightarrow \text{Resolution rule:} \]

\[ \frac{C_1 \lor x, \quad C_2 \lor \neg x}{C_1 \lor C_2} \]

Goal: derive empty clause \( \Lambda \)

Resolution rule is sound

\[ \Rightarrow \text{Derivation of } \Lambda \text{ certifies unsatisfiability} \]
Derive new clauses from old using:

\[ C_1 \lor x, \quad C_2 \lor \lnot x \]
\[ \frac{}{C_1 \lor C_2} \]

**Resolution** rule:

Goal: derive empty clause \( \Lambda \)

Resolution rule is *sound*

Derivation of \( \Lambda \) certifies unsatisfiability

\[ (x_2 \lor x_3) \land (\lnot x_1 \lor \lnot x_3) \land (\lnot x_2) \land (x_1 \lor \lnot x_3) \]

\[ \text{Size}_R(F): \text{# of clauses} \quad (7) \]

\[ \text{Width}_R(F): \text{max # of literals in any clause} \]
Resolution: A method for proving that a CNF formula is **unsatisfiable**

Derive new clauses from old using:

\[ C_1 \lor x, \quad C_2 \lor \neg x \quad \Rightarrow \quad C_1 \lor C_2 \]

**Goal:** derive empty clause \( \Lambda \)

Resolution rule is **sound**

\[ \Rightarrow \quad \text{Derivation of } \Lambda \text{ certifies unsatisfiability} \]

\[ (x_2 \lor x_3) \land (\neg x_1 \lor \neg x_3) \land (\neg x_2) \land (x_1 \lor \neg x_3) \]

\[ \text{Size}_R(F): \# \text{ of clauses} \]

\[ (\neg x_2) \]

\[ (x_2 \lor x_3) \]

\[ \neg x_3 \]

\[ (x_1 \lor \neg x_3) \quad (\neg x_1 \lor \neg x_3) \]

\[ \text{Width}_R(F): \max \# \text{ of literals in any clause} \]

\[ (7) \]

\[ (2) \]
Last Time

- Introduced the **DPLL** algorithm
  - Lower bounds on the runtime of **DPLL** follow from lower bounds on tree Resolution proofs

- Introduced the **CDCL** algorithm by extending DPLL with
  - Unit Propagation
  - Clause Learning
  - Restarts
Last Time — Unit Prop

\[(x \lor y) \land (z \lor w) \land (h \lor \bar{z} \lor \bar{y}) \land (\bar{i} \lor \bar{z}) \land (i \lor \bar{z} \lor \bar{y})\]

**Unit clause:** a clause containing a **single** literal \(\ell\)

**Unit Propagation:** if \(F\) contains a unit clause
(under the current assignment), set \(\ell = 1\)
Last Time — Unit Prop

\[(x \lor y) \land (z \lor w) \land (h \lor \bar{z} \lor \bar{y}) \land (\bar{i} \lor \bar{z}) \land (i \lor \bar{z} \lor \bar{y})\]

**Unit clause:** a clause containing a **single** literal \( \ell \)

**Unit Propagation:** if \( F \) contains a unit clause (under the current assignment), set \( \ell = 1 \)
Unit clause: a clause containing a single literal $\ell$

Unit Propagation: if $F$ contains a unit clause (under the current assignment), set $\ell = 1$
Last Time — Unit Prop

Unit clause: a clause containing a **single** literal $\ell$

Unit Propagation: if $F$ contains a unit clause (under the current assignment), set $\ell = 1$
Last Time — Unit Prop

Unit clause: a clause containing a **single** literal \( \ell \)

Unit Propagation: if \( F \) contains a unit clause
(under the current assignment), set \( \ell = 1 \)

\[(x \lor y) \land (z \lor w) \land (h \lor \bar{z} \lor \bar{y}) \land (\bar{i} \lor \bar{z}) \land (i \lor \bar{z} \lor \bar{y})\]
Unit clause: a clause containing a **single** literal \( \ell \).

Unit Propagation: if \( F \) contains a unit clause (under the current assignment), set \( \ell = 1 \).
Last Time — Unit Prop

Unit clause: a clause containing a single literal $\ell$

Unit Propagation: if $F$ contains a unit clause (under the current assignment), set $\ell = 1$
Last Time — Clause Learning

\[(x \lor y) \land (z \lor w) \land (h \lor \bar{z} \lor \bar{y}) \land (\bar{i} \lor \bar{z}) \land (i \lor \bar{z} \lor \bar{y})\]

When a conflict, use Resolution to learn a clause
Last Time — Clause Learning

When a conflict, use Resolution to learn a clause

1-UIP Clause
Obtained by resolving the conflict clause along the path until there is only one literal in the clause at the largest decision level

\[(x \lor y) \land (z \lor w) \land (h \lor \bar{z} \lor \bar{y}) \land (\bar{i} \lor \bar{z}) \land (i \lor \bar{z} \lor \bar{y})\]
When a conflict, use Resolution to learn a clause

1-UlIP Clause
Obtained by resolving the conflict clause along the path until there is only one literal in the clause at the largest decision level
Last Time — Clause Learning

When a conflict, use Resolution to learn a clause

\[(x \lor y) \land (z \lor w) \land (h \lor \bar{z} \lor \bar{y}) \land (i \lor \bar{z} \lor \bar{y})\]

1-UIP Clause

Obtained by resolving the conflict clause along the path until there is only one literal in the clause at the largest decision level.
Last Time — Clause Learning

When a conflict, use Resolution to learn a clause

1-UIP Clause

Obtained by resolving the conflict clause along the path until there is only one literal in the clause at the largest decision level

\[(x \lor y) \land (z \lor w) \land (h \lor \bar{z} \lor \bar{y}) \land (\bar{i} \lor \bar{z}) \land (i \lor \bar{z} \lor \bar{y})\]
Last Time — Clause Learning

When a conflict, use Resolution to learn a clause

1-UIP Clause

Obtained by resolving the conflict clause along the path until there is only one literal in the clause at the largest decision level

Backtracking with 1-UIP:
Remove everything up to the second largest decision level in the learned clause
When a conflict, use Resolution to learn a clause

1-UIP Clause
Obtained by resolving the conflict clause along the path until there is only one literal in the clause at the largest decision level

Backtracking with 1-UIP:
Remove everything up to the second largest decision level in the learned clause

\[(\bar{z} \lor \bar{y}) \land (x \lor y) \land (z \lor w) \land (h \lor \bar{z} \lor \bar{y}) \land (\bar{i} \lor \bar{z}) \land (i \lor \bar{z} \lor \bar{y})\]
Analyzing CDCL

Can we show CDCL doesn’t solve SAT in polytime?
Analyzing CDCL

Q. Can we show CDCL doesn’t solve SAT in polytime?

→ We saw that DPLL when run on an unsatisfiable formula gives a tree Resolution proof
Analyzing CDCL

Can we show CDCL doesn’t solve SAT in polytime?

We saw that DPLL when run on an unsatisfiable formula gives a tree Resolution proof.

**Theorem:** Let $F$ be an unsatisfiable CNF formula. If CDCL takes time $s$ to solve $F$, then there is a size-$s$ Resolution proof of $F$. 
Analyzing CDCL

Q. Can we show CDCL doesn’t solve SAT in polytime?

→ We saw that DPLL when run on an unsatisfiable formula gives a tree Resolution proof

**Theorem:** Let $F$ be an unsatisfiable CNF formula. If CDCL takes time $s$ to solve $F$, then there is a size-$s$ Resolution proof of $F$

Q. What do we learn if we run CDCL on an unsatisfiable formula?
Analyzing CDCL

What do we learn if we run CDCL on an unsatisfiable formula?

\((y \lor z) \land (y \lor \bar{z}) \land (x \lor \bar{y} \lor z) \land (x \lor \bar{y} \lor \bar{z}) \land (\bar{x} \lor \bar{y})\)
Analyzing CDCL

Q. What do we learn if we run CDCL on an unsatisfiable formula?

\[(y \lor z) \land (y \lor \bar{z}) \land (x \lor \bar{y} \lor z) \land (x \lor \bar{y} \lor \bar{z}) \land (\bar{x} \lor \bar{y})\]
Analyzing CDCL

What do we learn if we run CDCL on an unsatisfiable formula?

\[(y \lor z) \land (y \lor \bar{z}) \land (x \lor \bar{y} \lor z) \land (x \lor \bar{y} \lor \bar{z}) \land (\bar{x} \lor \bar{y})\]
Analyzing CDCL

Q. What do we learn if we run CDCL on an unsatisfiable formula?

\[(y \lor z) \land (y \lor \neg z) \land (x \lor \neg y \lor z) \land (x \lor \neg y \lor \neg z) \land (\neg x \lor \neg y)\]
Analyzing CDCL

What do we learn if we run CDCL on an unsatisfiable formula?

\[(y \lor z) \land (y \lor \overline{z}) \land (x \lor \overline{y} \lor z) \land (x \lor \overline{y} \lor \overline{z}) \land (\overline{x} \lor \overline{y})\]
Analyzing CDCL

Q. What do we learn if we run CDCL on an unsatisfiable formula?

\[(y \lor z) \land (y \lor \neg z) \land (x \lor \neg y \lor z) \land (x \lor \neg y \lor \neg z) \land (\neg x \lor \neg y)\]
Analyzing CDCL

Q: What do we learn if we run CDCL on an unsatisfiable formula?

\[(y) \land (y \lor z) \land (y \lor \bar{z}) \land (x \lor \bar{y} \lor z) \land (x \lor \bar{y} \lor \bar{z}) \land (\bar{x} \lor \bar{y})\]
Analyzing CDCL

Q. What do we learn if we run CDCL on an unsatisfiable formula?

\[(y) \land (y \lor z) \land (y \lor \bar{z}) \land (x \lor \bar{y} \lor z) \land (x \lor \bar{y} \lor \bar{z}) \land (\bar{x} \lor \bar{y})\]

1-UIP Clause!

Backjump to second highest decision level: 0
Analyzing CDCL

Q. What do we learn if we run CDCL on an unsatisfiable formula?

$$(y) \land (y \lor z) \land (y \lor \bar{z}) \land (x \lor \bar{y} \lor z) \land (x \lor \bar{y} \lor \bar{z}) \land (\bar{x} \lor \bar{y})$$
Analyzing CDCL

Q. What do we learn if we run CDCL on an unsatisfiable formula?

\[(y) \land (y \lor z) \land (y \lor \overline{z}) \land (x \lor \overline{y} \lor z) \land (x \lor \overline{y} \lor \overline{z}) \land (\overline{x} \lor \overline{y})\]
Analyzing CDCL

Q. What do we learn if we run CDCL on an unsatisfiable formula?

$$(y) \land (y \lor z) \land (y \lor \overline{z}) \land (x \lor \overline{y} \lor z) \land (x \lor \overline{y} \lor \overline{z}) \land (\overline{x} \lor \overline{y})$$
Analyzing CDCL

What do we learn if we run CDCL on an unsatisfiable formula?

\[(y) \land (y \lor z) \land (y \lor \overline{z}) \land (x \lor \overline{y} \lor z) \land (x \lor \overline{y} \lor \overline{z}) \land (\overline{x} \lor \overline{y})\]
Analyzing CDCL

$Q$. What do we learn if we run CDCL on an unsatisfiable formula?

$$(y) \land (y \lor z) \land (y \lor \bar{z}) \land (x \lor \bar{y} \lor z) \land (x \lor \bar{y} \lor \bar{z}) \land (\bar{x} \lor \bar{y})$$
Analyzing CDCL

Q. What do we learn if we run CDCL on an unsatisfiable formula?

\[(y) \land (y \lor z) \land (y \lor \overline{z}) \land (x \lor \overline{y} \lor z) \land (x \lor \overline{y} \lor \overline{z}) \land (\overline{x} \lor \overline{y})\]
Analyzing CDCL

What do we learn if we run CDCL on an unsatisfiable formula?

\[(y) \land (y \lor z) \land (y \lor \overline{z}) \land (x \lor \overline{y} \lor z) \land (x \lor \overline{y} \lor \overline{z}) \land (\overline{x} \lor \overline{y})\]
Analyzing CDCL

*Q.* What do we learn if we run CDCL on an unsatisfiable formula?

\[ (\neg y) \land (y) \land (y \lor z) \land (y \lor \neg z) \land (x \lor \neg y \lor z) \land (x \lor \neg y \lor \neg z) \land (\neg x \lor y) \]
Analyzing CDCL

What do we learn if we run CDCL on an unsatisfiable formula?

\[(\bar{y}) \land (y) \land (y \lor z) \land (y \lor \bar{z}) \land (x \lor \bar{y} \lor z) \land (x \lor \bar{y} \lor \bar{z}) \land (\bar{x} \lor \bar{y})\]

1-UIP Clause!

Backjump to second highest decision level: 0
Analyzing CDCL

What do we learn if we run CDCL on an unsatisfiable formula?

\((\overline{y}) \land (y) \land (y \lor z) \land (y \lor \overline{z}) \land (x \lor \overline{y} \lor z) \land (x \lor \overline{y} \lor \overline{z}) \land (x \lor \overline{y})\)
Analyzing CDCL

What do we learn if we run CDCL on an unsatisfiable formula?

\((\bar{y}) \land (y) \land (y \lor z) \land (y \lor \bar{z}) \land (x \lor \bar{y} \lor z) \land (x \lor \bar{y} \lor \bar{z}) \land (\bar{x} \lor \bar{y})\)
Analyzing CDCL

What do we learn if we run CDCL on an unsatisfiable formula?

\[ \Lambda \land (\bar{y}) \land (y) \land (y \lor z) \land (y \lor \bar{z}) \land (x \lor \bar{y} \lor z) \land (x \lor \bar{y} \lor \bar{z}) \land (\bar{x} \lor \bar{y}) \]
Analyzing CDCL

Q. What do we learn if we run CDCL on an unsatisfiable formula?

\[ \Lambda \land (\bar{y}) \land (y) \land (y \lor z) \land (y \lor \bar{z}) \land (x \lor \bar{y} \lor z) \land (x \lor \bar{y} \lor \bar{z}) \land (\bar{x} \lor \bar{y}) \]

Learned empty clause \( \Lambda \).

Halt: Unsatisfiable!
Analyzing CDCL

Q. What do we learn if we run CDCL on an unsatisfiable formula?

\[ \Lambda \land (\bar{y}) \land (y) \land (y \lor z) \land (y \lor \bar{z}) \land (x \lor \bar{y} \lor z) \land (x \lor \bar{y} \lor \bar{z}) \land (\bar{x} \lor \bar{y}) \]

Learned empty clause \( \Lambda \).

Halt: Unsatisfiable!

Takeaway:
CDCL run on an unsatisfiable formula halts when \( \Lambda \) is derived from clause learning
Analyzing CDCL

Q. What do we learn if we run CDCL on an unsatisfiable formula?

\[ \Lambda \land (\overline{y}) \land (y) \land (y \lor z) \land (y \lor \overline{z}) \land (x \lor \overline{y} \lor z) \land (x \lor \overline{y} \lor \overline{z}) \land (\overline{x} \lor \overline{y}) \]

Learned empty clause \( \Lambda \).

Halt: Unsatisfiable!

Takeaway:
CDCL run on an unsatisfiable formula halts when \( \Lambda \) is derived from clause learning
\( \rightarrow \) Clause learning derives new clauses from old ones using Resolution
Analyzing CDCL

**Theorem:** Let $F$ be an unsatisfiable CNF formula. If CDCL takes time $s$ to solve $F$, then there is a size-$s$ Resolution proof of $F$.

**Proof:** Every time CDCL learns a clause, derive that clause in Resolution.
Analyzing CDCL

**Theorem:** Let $F$ be an unsatisfiable CNF formula. If CDCL takes time $s$ to solve $F$, then there is a size-$s$ Resolution proof of $F$.

**Proof:** Every time CDCL learns a clause, derive that clause in Resolution.

→ Because CDCL halts when $\Lambda$ is derived, we have a Resolution proof!
Analyzing CDCL

(y \lor z) \land (y \lor \bar{z}) \land (x \lor \bar{y} \lor z) \land (x \lor \bar{y} \lor \bar{z}) \land (\bar{x} \lor \bar{y})

Resolution

(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})
Analyzing CDCL

\[(y \lor z) \land (y \lor \bar{z}) \land (x \lor \bar{y} \lor z) \land (x \lor \bar{y} \lor \bar{z}) \land (\bar{x} \lor \bar{y})\]

Resolution

\[(y \lor z), (y \lor \bar{z}), (x \lor \bar{y} \lor z), (x \lor \bar{y} \lor \bar{z}), (\bar{x} \lor \bar{y})\]
Analyzing CDCL

Resolution

\[(y \lor z) \land (y \lor \bar{z}) \land (x \lor \bar{y} \lor z) \land (x \lor \bar{y} \lor \bar{z}) \land (\bar{x} \lor \bar{y})\]

\[(y \lor z), (y \lor \bar{z}), (x \lor \bar{y} \lor z), (x \lor \bar{y} \lor \bar{z}), (\bar{x} \lor \bar{y})\]
Analyzing CDCL

\[(y \lor z) \land (y \lor \bar{z}) \land (x \lor \bar{y} \lor z) \land (x \lor \bar{y} \lor \bar{z}) \land (\bar{x} \lor \bar{y})\]

Resolution

\[(y \lor z), (y \lor \bar{z}), (x \lor \bar{y} \lor z), (x \lor \bar{y} \lor \bar{z}), (\bar{x} \lor \bar{y})\]
Analyzing CDCL

Resolution

\[(y \lor z) \land (y \lor \bar{z}) \land (x \lor \bar{y} \lor z) \land (x \lor \bar{y} \lor \bar{z}) \land (\bar{x} \lor \bar{y})\]
Analyzing CDCL

Resolution

(y ∨ z) ∧ (y ∨ ȳ) ∧ (x ∨ ŷ ∨ z) ∧ (x ∨ ŷ ∨ ȳ) ∧ (̄x ∨ ŷ)
Analyzing CDCL

\[(y) \land (y \lor z) \land (y \lor \bar{z}) \land (x \lor \bar{y} \lor z) \land (x \lor \bar{y} \lor \bar{z}) \land (\bar{x} \lor \bar{y})\]
Analyzing CDCL

\[(y) \land (y \lor z) \land (y \lor \bar{z}) \land (x \lor \bar{y} \lor z) \land (x \lor \bar{y} \lor \bar{z}) \land (\bar{x} \lor \bar{y})\]

Resolution

\[(y \lor z), (y \lor \bar{z}), (x \lor \bar{y} \lor z), (x \lor \bar{y} \lor \bar{z}), (\bar{x} \lor \bar{y})\]
Analyzing CDCL

\[(y) \land (y \lor z) \land (y \lor \bar{z}) \land (x \lor \bar{y} \lor z) \land (x \lor \bar{y} \lor \bar{z}) \land (\bar{x} \lor \bar{y})\]
Analyzing CDCL

\[(y) \land (y \lor z) \land (y \lor \bar{z}) \land (x \lor \bar{y} \lor z) \land (x \lor \bar{y} \lor \bar{z}) \land (\bar{x} \lor \bar{y})\]
Analyzing CDCL

Resolution

\[(y) \land (y \lor z) \land (y \lor \bar{z}) \land (x \lor \bar{y} \lor z) \land (x \lor \bar{y} \lor \bar{z}) \land (\bar{x} \lor \bar{y})\]
Analyzing CDCL

\[(y) \land (y \lor z) \land (y \lor \bar{z}) \land (x \lor \bar{y} \lor z) \land (x \lor \bar{y} \lor \bar{z}) \land (\bar{x} \lor \bar{y})\]

Resolution

\[(y \lor z), (y \lor \bar{z}), (x \lor \bar{y} \lor z), (x \lor \bar{y} \lor \bar{z}), (\bar{x} \lor \bar{y})\]
Analyzing CDCL

Resolution

\[(\bar{y}) \land (y) \land (y \lor z) \land (y \lor \bar{z}) \land (x \lor \bar{y} \lor z) \land (x \lor \bar{y} \lor \bar{z}) \land (\bar{x} \lor \bar{y})\]
Analyzing CDCL

\[(\bar{y}) \land (y) \land (y \lor z) \land (y \lor \bar{z}) \land (x \lor \bar{y} \lor z) \land (x \lor \bar{y} \lor \bar{z}) \land (\bar{x} \lor \bar{y})\]
Analyzing CDCL

\[(\bar{y}) \land (y) \land (y \lor z) \land (y \lor \bar{z}) \land (x \lor \bar{y} \lor z) \land (x \lor \bar{y} \lor \bar{z}) \land (\bar{x} \lor \bar{y})\]
Analyzing CDCL

\[(\overline{y}) \land (y) \land (y \lor z) \land (y \lor \overline{z}) \land (x \lor \overline{y} \lor z) \land (x \lor \overline{y} \lor \overline{z}) \land (\overline{x} \lor \overline{y})\]
Analyzing CDCL

\[ \Lambda \land (\bar{y}) \land (y) \land (y \lor z) \land (y \lor \bar{z}) \land (x \lor \bar{y} \lor z) \land (x \lor \bar{y} \lor \bar{z}) \land (\bar{x} \lor \bar{y}) \]

Resolution

(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})
Analyzing CDCL

\[ \Lambda \land (\bar{y}) \land (y) \land (y \lor z) \land (y \lor \bar{z}) \land (x \lor \bar{y} \lor z) \land (x \lor \bar{y} \lor \bar{z}) \land (\bar{x} \lor \bar{y}) \]
Analyzing CDCL

**Theorem:** Let $F$ be an unsatisfiable CNF formula. If CDCL takes time $s$ to solve $F$, then there is a size-$s$ Resolution proof of $F$.

In order to prove bounds on the runtime of CDCL it suffices to analyze Resolution proof size.
Resolution Lower Bounds — Some History

First lower bound proved by Armin Haken in `85

Technique: Bottleneck Counting
Resolution Lower Bounds — Some History

First lower bound proved by Armin Haken in `85

**Technique: Bottleneck Counting**

In every Resolution proof of $F$,

1. Every $x \in \{0, 1\}^n$ falsifies a wide clause in the proof
First lower bound proved by Armin Haken in `85

**Technique: Bottleneck Counting**

In every Resolution proof of $F$, 

1. Every $x \in \{0,1\}^n$ falsifies a wide clause in the proof
2. Every wide clause is falsified by only a small number of $x \in \{0,1\}^n$
Resolution Lower Bounds — Some History

First lower bound proved by Armin Haken in `85

**Technique: Bottleneck Counting**

In every Resolution proof of $F$,

1. Every $x \in \{0,1\}^n$ falsifies a wide clause in the proof
2. Every wide clause is falsified by only a small number of $x \in \{0,1\}^n$

$\implies$ Proof must have many wide clauses! (Size lower bound!)
Resolution Lower Bounds — Some History

Connection between *width* and *size* formalized by Ben-Sasson Wigderson `99
Resolution Lower Bounds — Some History

Connection between width and size formalized by Ben-Sasson Wigderson ‘99

**Theorem:**
For any unsatisfiable CNF formula $F$ on $n$ variables with clauses of width $\leq w$,

$$\text{size}_R(F) \geq \exp \Omega((\text{width}_R(F) - w)^2/n)$$
Resolution Lower Bounds — Some History

Connection between width and size formalized by Ben-Sasson Wigderson `99

**Theorem:**
For any unsatisfiable CNF formula $F$ on $n$ variables with clauses of width $\leq w$,

$$size_R(F) \geq \exp \Omega((width_R(F) - w)^2/n)$$

**Takeaway:** If $w = O(1)$ and $width_R(F) = \omega(\sqrt{n})$ then we get size lower bounds!
Connection between width and size formalized by Ben-Sasson Wigderson `99

**Theorem:**
For any unsatisfiable CNF formula $F$ on $n$ variables with clauses of width $\leq w$,

$$size_R(F) \geq \exp \Omega((\text{width}_R(F) - w)^2 / n)$$

**Takeaway:** If $w = O(1)$ and $\text{width}_R(F) = \omega(\sqrt{n})$ then we get size lower bounds!

We can prove a similar theorem, with a much simpler proof by composition!
Resolution Lower Bounds — Some History

Connection between width and size formalized by Ben-Sasson Wigderson `99

Theorem:
For any unsatisfiable CNF formula $F$ on $n$ variables with clauses of width $\leq w$,

$$size_R(F) \geq \exp \Omega((width_R(F) - w)^2/n)$$

Takeaway: If $w = O(1)$ and $width_R(F) = \omega(\sqrt{n})$ then we get size lower bounds!

We can prove a similar theorem, with a much simpler proof by composition!

Theorem:
For any unsatisfiable CNF formula $F$,

$$size_R(F \circ XOR) \geq 2^{\Omega(Width_R(F))}$$
**Theorem:**
For any unsatisfiable CNF formula $F$,

$$size_R(F \circ XOR) \geq 2^{\Omega(\text{Width}_R(F))}$$

$F(x_1, \ldots, x_n) \circ XOR$ obtained by substituting $x_i \leftarrow y_i \oplus z_i$.
Theorem:
For any unsatisfiable CNF formula $F$,
\[ \text{size}_R(F \circ XOR) \geq 2^\Omega(\text{Width}_R(F)) \]

$F(x_1, \ldots, x_n) \circ XOR$ obtained by substituting $x_i \leftarrow y_i \oplus z_i = (\bar{y_i} \lor \bar{z_i}) \land (y_i \lor z_i)$
Theorem:

For any unsatisfiable CNF formula $F$, 

$$size_R(F \circ XOR) \geq 2^{\Omega(\text{Width}_R(F))}$$

$F(x_1, \ldots, x_n) \circ XOR$ obtained by substituting 

$$x_i \leftarrow y_i \oplus z_i = (\bar{y}_i \lor \bar{z}_i) \land (y_i \lor z_i)$$

e.g. $(x_1 \lor \bar{x}_2) \circ XOR$
**Theorem:**
For any unsatisfiable CNF formula $F$,

$$\text{size}_R(F \circ XOR) \geq 2^{\Omega(\text{Width}_R(F))}$$

$F(x_1, \ldots, x_n) \circ XOR$ obtained by substituting $x_i \leftarrow y_i \oplus z_i = (\bar{y}_i \lor \bar{z}_i) \land (y_i \lor z_i)$

e.g. $(x_1 \lor \bar{x}_2) \circ XOR = (y_1 \oplus z_1) \lor \neg(y_2 \oplus z_2)$
Size to Width

**Theorem:**
For any unsatisfiable CNF formula $F$,

$$size_R(F \circ XOR) \geq 2^{\Omega(\text{Width}_R(F))}$$

$F(x_1, \ldots, x_n) \circ XOR$ obtained by substituting $x_i \leftarrow y_i \oplus z_i = (\bar{y}_i \lor \bar{z}_i) \land (y_i \lor z_i)$

e.g. $(x_1 \lor \bar{x}_2) \circ XOR = (y_1 \oplus z_1) \lor \neg(y_2 \oplus z_2)$

$$= ((\bar{y}_1 \lor \bar{z}_1) \land (y_1 \lor z_1)) \lor ((y_2 \land z_2) \lor (\bar{y}_2 \land z_2))$$
Size to Width

Theorem:
For any unsatisfiable CNF formula $F$,

$$size_R(F \circ XOR) \geq 2^{\Omega(\text{Width}_R(F))}$$

$F(x_1, \ldots, x_n) \circ XOR$ obtained by substituting $x_i \leftarrow y_i \oplus z_i = (\bar{y}_i \lor \bar{z}_i) \land (y_i \lor z_i)$

e.g. $(x_1 \lor \bar{x}_2) \circ XOR = (y_1 \oplus z_1) \lor \neg(y_2 \oplus z_2) = (((\bar{y}_1 \lor \bar{z}_1) \land (y_1 \lor z_1)) \lor ((y_2 \land z_2) \lor (\bar{y}_2 \land z_2))$}
=... expand as CNF
Reminder of Today

Prove a lower bound on Resolution and therefore CDCL runtime!
Reminder of Today

Prove a lower bound on Resolution and therefore CDCL runtime!

Step 1. Prove the theorem:

**Theorem:**

For any unsatisfiable CNF formula $F$,

$$\text{size}_R(F \circ XOR) \geq 2^{\Omega(\text{Width}_R(F))}$$
Reminder of Today

Prove a lower bound on Resolution and therefore CDCL runtime!

Step 1. Prove the theorem:

**Theorem:**
For any unsatisfiable CNF formula $F$,

$$size_R(F \circ XOR) \geq 2^{\Omega(Width_R(F))}$$

Step 2. Prove that some formula $F$ (Pigeonhole formula) requires large width
**Theorem:**
For any unsatisfiable CNF formula $F$,

$$size_R(F \circ XOR) \geq 2^{\Omega(Width_R(F))}$$

**Proof:** $w := Width_R(F)$, let $\Pi$ be any Resolution proof of $F \circ XOR$
Size to Width

**Theorem:**
For any unsatisfiable CNF formula $F$,

$$\text{size}_R(F \circ \text{XOR}) \geq 2^{\Omega(\text{Width}_R(F))}$$

**Proof:** $w := \text{Width}_R(F)$, let $\Pi$ be any Resolution proof of $F \circ \text{XOR}$
Theorem: For any unsatisfiable CNF formula $F$, 

$$\text{size}_R(F \circ XOR) \geq 2^{\Omega(\text{Width}_R(F))}$$

Proof: $w := \text{Width}_R(F)$, let $\Pi$ be any Resolution proof of $F \circ XOR$

Idea: construct a partial assignment $\rho \in \{0, 1, *\}^n$ so that
Size to Width

**Theorem:**
For any unsatisfiable CNF formula $F$, 

$$\text{size}_R(F \circ XOR) \geq 2^{\Omega(\text{Width}_R(F))}$$

**Proof:** $w := \text{Width}_R(F)$, let $\Pi$ be any Resolution proof of $F \circ XOR$

**Idea:** construct a partial assignment $\rho \in \{0,1,*\}^n$ so that

1. $\Pi \upharpoonright \rho$ is a proof of $F$
**Size to Width**

**Theorem:**
For any unsatisfiable CNF formula $F$, 

$$\text{size}_R(F \circ \text{XOR}) \geq 2^{\Omega(\text{Width}_R(F))}$$

**Proof:** $w := \text{Width}_R(F)$, let $\Pi$ be any Resolution proof of $F \circ \text{XOR}$

**Idea:** construct a partial assignment $\rho \in \{0,1,*\}^n$ so that

1. $\Pi \upharpoonright \rho$ is a proof of $F$
2. If $\Pi$ is small $\implies \Pi \upharpoonright \rho$ has width $< w$
Size to Width

Generate $\rho$ randomly: For each $i \in [n]$, flip a coin
Generate $\rho$ randomly: For each $i \in [n]$, flip a coin

- **Heads:** fix $y_i \in \{0,1\}$ with equal probability.
- **Tails:** fix $z_i \in \{0,1\}$ with equal probability.
Size to Width

Generate $\rho$ randomly: For each $i \in [n]$, flip a coin

$\rightarrow$ **Heads:** fix $y_i \in \{0,1\}$ with equal probability.

$\rightarrow$ **Tails:** fix $z_i \in \{0,1\}$ with equal probability.

**Claim:** $\Pi \upharpoonright \rho$ is a proof of $F \upharpoonright \rho = F$
Size to Width

Generate $\rho$ randomly: For each $i \in [n]$, flip a coin

→ **Heads:** fix $y_i \in \{0,1\}$ with equal probability.

→ **Tails:** fix $z_i \in \{0,1\}$ with equal probability.

**Claim:** $\Pi \upharpoonright \rho$ is a proof of $F \upharpoonright \rho = F$

For all $i \in [n]$, $\rho$ fixes exactly one of $y_i$ or $z_i$ in $(y_i \oplus z_i)$; suppose it’s $z_i$
Generate $\rho$ randomly: For each $i \in [n]$, flip a coin

→ **Heads:** fix $y_i \in \{0,1\}$ with equal probability.

→ **Tails:** fix $z_i \in \{0,1\}$ with equal probability.

**Claim:** $\Pi \upharpoonright \rho$ is a proof of $F \upharpoonright \rho = F$

For all $i \in [n]$, $\rho$ fixes exactly one of $y_i$ or $z_i$ in $(y_i \oplus z_i)$; suppose it’s $z_i$

- If $z_i = 1$ then $(y_i \oplus z_i) = \bar{z}_i$

- If $z_i = 0$ then $(y_i \oplus z_i) = z_i$
Size to Width

Generate $\rho$ randomly: For each $i \in [n]$, flip a coin

→ **Heads:** fix $y_i \in \{0,1\}$ with equal probability.

→ **Tails:** fix $z_i \in \{0,1\}$ with equal probability.

**Claim:** $\Pi \upharpoonright \rho$ is a proof of $F \upharpoonright \rho = F$

For all $i \in [n]$, $\rho$ fixes *exactly* one of $y_i$ or $z_i$ in $(y_i \oplus z_i)$; suppose it’s $z_i$

- If $z_i = 1$ then $(y_i \oplus z_i) = \bar{z}_i$
- If $z_i = 0$ then $(y_i \oplus z_i) = z_i$

Thus, $F \upharpoonright \rho = F$ (up to a renaming of the variables, and a flipping of their sign)
Size to Width

Generate $\rho$ randomly: For each $i \in [n]$, flip a coin

$\rightarrow$ **Heads:** fix $y_i \in \{0,1\}$ with equal probability.

$\rightarrow$ **Tails:** fix $z_i \in \{0,1\}$ with equal probability.

**Claim:** $\Pi \upharpoonright \rho$ is a proof of $F \upharpoonright \rho = F$

For all $i \in [n]$, $\rho$ fixes exactly one of $y_i$ or $z_i$ in $(y_i \oplus z_i)$; suppose it’s $z_i$

- If $z_i = 1$ then $(y_i \oplus z_i) = \bar{z}_i$
- If $z_i = 0$ then $(y_i \oplus z_i) = z_i$

Thus, $F \upharpoonright \rho = F$ (up to a renaming of the variables, and a flipping of their sign)

$\implies \Pi \upharpoonright \rho$ is a Resolution proof of $F$
Size to Width

Generate $\rho$ randomly: For each $i \in [n]$, flip a coin

$\rightarrow$ **Heads:** fix $y_i \in \{0,1\}$ with equal probability.

$\rightarrow$ **Tails:** fix $z_i \in \{0,1\}$ with equal probability.

**Claim:** If $|\Pi|$ is small, then there is $\rho$ such that $\Pi \upharpoonright \rho$ has width $< w$
Size to Width

Generate $\rho$ randomly: For each $i \in [n]$, flip a coin

$\rightarrow$ **Heads:** fix $y_i \in \{0,1\}$ with equal probability.

$\rightarrow$ **Tails:** fix $z_i \in \{0,1\}$ with equal probability.

**Claim:** If $|\Pi|$ is small, then there is $\rho$ such that $\Pi \upharpoonright \rho$ has width $< w$

**Want to show:** every wide clause in $\Pi$ is satisfied by $\rho$ with probability $> 0$
Size to Width

Generate $\rho$ randomly: For each $i \in [n]$, flip a coin

$\rightarrow$ **Heads:** fix $y_i \in \{0,1\}$ with equal probability.

$\rightarrow$ **Tails:** fix $z_i \in \{0,1\}$ with equal probability.

**Claim:** If $|\Pi|$ is small, then there is $\rho$ such that $\Pi \upharpoonright \rho$ has width $< w$

**Want to show:** every wide clause in $\Pi$ is satisfied by $\rho$ with probability $> 0$

Let $C$ have width $w$
Generate $\rho$ randomly: For each $i \in [n]$, flip a coin

→ **Heads:** fix $y_i \in \{0,1\}$ with equal probability.

→ **Tails:** fix $z_i \in \{0,1\}$ with equal probability.

**Claim:** If $|\Pi|$ is small, then there is $\rho$ such that $\Pi \upharpoonright \rho$ has width $< w$

**Want to show:** every wide clause in $\Pi$ is satisfied by $\rho$ with probability $> 0$

Let $C$ have width $w$

→ Each literal in $C$ is set to 1 w.p. $1/4 \implies Pr[C(\rho) \neq 1] \leq (3/4)^w$
Size to Width

Generate $\rho$ randomly: For each $i \in [n]$, flip a coin

→ **Heads:** fix $y_i \in \{0, 1\}$ with equal probability.
→ **Tails:** fix $z_i \in \{0, 1\}$ with equal probability.

**Claim:** If $|\Pi|$ is small, then there is $\rho$ such that $\Pi \upharpoonright \rho$ has width $< w$

**Want to show:** every wide clause in $\Pi$ is satisfied by $\rho$ with probability $> 0$

Let $C$ have width $w$

→ Each literal in $C$ is set to 1 w.p. $1/4 \implies Pr[C(\rho) \neq 1] \leq (3/4)^w$

By a union bound over the wide clauses in $\Pi$

$$Pr[\Pi \upharpoonright \rho \text{ has width } \geq w] \leq (3/4)^w |\Pi|$$
**Size to Width**

Generate $\rho$ randomly: For each $i \in [n]$, flip a coin

→ **Heads:** fix $y_i \in \{0,1\}$ with equal probability.

→ **Tails:** fix $z_i \in \{0,1\}$ with equal probability.

**Claim:** If $|\Pi|$ is small, then there is $\rho$ such that $\Pi \upharpoonright \rho$ has width $< w$

**Want to show:** every wide clause in $\Pi$ is satisfied by $\rho$ with probability $> 0$

Let $C$ have width $w$

→ Each literal in $C$ is set to 1 w.p. $1/4 \implies Pr[C(\rho) \neq 1] \leq (3/4)^w$

By a union bound over the wide clauses in $\Pi$

$$Pr[\Pi \upharpoonright \rho \text{ has width } \geq w] \leq (3/4)^w |\Pi|$$

If $|\Pi| \leq (4/3)^w \implies$ exists $\rho$ such that $\Pi \upharpoonright \rho$ has width $< w$
**Size to Width**

Generate \( \rho \) randomly: For each \( i \in [n] \), flip a coin

→ **Heads:** fix \( y_i \in \{0,1\} \) with equal probability.

→ **Tails:** fix \( z_i \in \{0,1\} \) with equal probability.

**Claim:** If \( |\Pi| \) is small, then there is \( \rho \) such that \( \Pi \upharpoonright \rho \) has width \( < w \)

**Want to show:** every wide clause in \( \Pi \) is satisfied by \( \rho \) with probability \( > 0 \)

Let \( C \) have width \( w \)

→ Each literal in \( C \) is set to 1 w.p. \( 1/4 \) \( \implies \Pr[C(\rho) \neq 1] \leq (3/4)^w \)

By a union bound over the wide clauses in \( \Pi \)

\[
\Pr[\Pi \upharpoonright \rho \text{ has width } \geq w] \leq (3/4)^w |\Pi|
\]

If \( |\Pi| \leq (4/3)^w \implies \) exists \( \rho \) such that \( \Pi \upharpoonright \rho \) has width \( < w \) ← **Contradiction!**