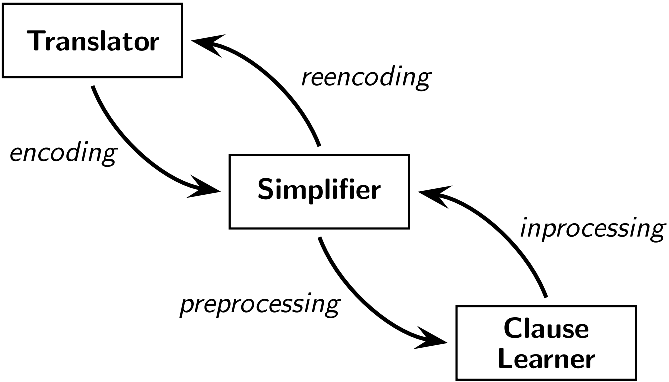
# Prepocessing

**Marijn J.H. Heule**

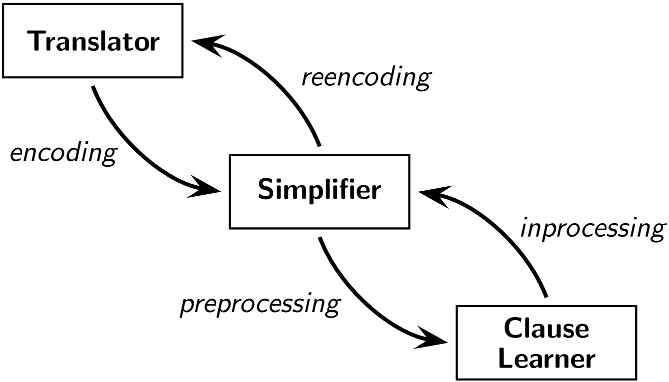
The University of Texas at Austin

July 5, 2013

# Interaction between different solving approaches



# Interaction between different solving approaches




It all comes down to adding and removing redundant clauses

# Redundant clauses

A clause is redundant w.r.t. a formula if adding it to the formula preserves satisfiability.

- For unsatisfiable formulas, all clauses can be added, including the empty clause ( ).

# Redundant clauses

A clause is redundant w.r.t. a formula if adding it to the formula preserves satisfiability.

- For unsatisfiable formulas, all clauses can be added, including the empty clause ( ).

A clause is redundant w.r.t. a formula if removing it from the formula preserves unsatisfiability.

- For satisfiable formulas, all clauses can be removed.

# Redundant clauses

A clause is redundant w.r.t. a formula if adding it to the formula preserves satisfiability.

- For unsatisfiable formulas, all clauses can be added, including the empty clause ( ).

A clause is redundant w.r.t. a formula if removing it from the formula preserves unsatisfiability.

- For satisfiable formulas, all clauses can be removed.

Challenge regarding redundant clauses:

- How to check redundancy in polynomial time?
- Ideally find redundant clauses in linear time

# Outline

1. Subsumption
2. Variable Elimination
3. Bounded Variable Addition
4. Blocked Clause Elimination
5. Hyper Binary Resolution
6. Unhiding Redundancy

# Subsumption

# Tautologies and Subsumption

**Definition (Tautology)**

A clause $C$ is a tautology if its contains two complementary literals $l, \bar{l}$.

**Example**

The clause $(a \vee b \vee \bar{b})$ is a tautology.

**Definition (Subsumption)**

Clause $C$ subsumes clause $D$ if and only if $C \subset D$.

**Example**

The clause $(a \vee b)$ subsumes clause $(a \vee b \vee \bar{c})$.

# Self-Subsuming Resolution

## Self-Subsuming Resolution

$$\frac{C \vee l \qquad D \vee \bar{l}}{D} \quad C \subseteq D \qquad\qquad \frac{(a \vee b \vee l) \qquad (a \vee b \vee c \vee \bar{l})}{(a \vee b \vee c)}$$

resolvent $D$ subsumes second antecedent $D \vee \bar{l}$

# Self-Subsuming Resolution

## Self-Subsuming Resolution

$$\frac{C \vee l \qquad D \vee \bar{l}}{D} \quad C \subseteq D \qquad\qquad \frac{(a \vee b \vee l) \qquad (a \vee b \vee c \vee \bar{l})}{(a \vee b \vee c)}$$

resolvent $D$ subsumes second antecedent $D \vee \bar{l}$

## Example

assume a CNF contains both antecedents

$\ldots(a \vee b \vee l)(a \vee b \vee c \vee \bar{l})\cdots$

if $D$ is added then $D \vee \bar{l}$ can be removed $\Downarrow$

which in essence *removes* $\bar{l}$ from $D \vee \bar{l}$ $\ldots(a \vee b \vee l)(a \vee b \vee c \quad )\ldots$

initially in the SATeLite preprocessor [EenBiere'07]

now common in many SAT solvers

# Self-Subsuming Example

## Self-Subsuming Resolution

$$\frac{C \vee l \qquad D \vee \bar{l}}{D} \quad C \subseteq D \qquad\qquad \frac{(a \vee b \vee l) \qquad (a \vee b \vee c \vee \bar{l})}{(a \vee b \vee c)}$$

resolvent $D$ subsumes second antecedent $D \vee \bar{l}$

## Example: Remove literals using self-subsumption

$$\begin{array}{l}(a \vee b \vee c) \wedge (\bar{a} \vee b \vee c) \wedge \\ (\bar{a} \vee b \vee \bar{c}) \wedge (a \vee \bar{b} \vee c) \wedge \\ (\bar{a} \vee \bar{b} \vee d) \wedge (\bar{a} \vee \bar{b} \vee \bar{d}) \wedge \\ (a \vee \bar{c} \vee d) \wedge (a \vee \bar{c} \vee \bar{d})\end{array}$$

# Self-Subsuming Example

## Self-Subsuming Resolution

$$\frac{C \vee l \qquad D \vee \bar{l}}{D} \quad C \subseteq D \qquad\qquad \frac{(a \vee b \vee l) \qquad (a \vee b \vee c \vee \bar{l})}{(a \vee b \vee c)}$$

resolvent $D$ subsumes second antecedent $D \vee \bar{l}$

## Example: Remove literals using self-subsumption

$$\begin{array}{lcl}
(\quad b \vee c) & \wedge & (\bar{a} \vee b \vee c) \wedge \\
(\bar{a} \vee b \vee \bar{c}) & \wedge & (a \vee \bar{b} \vee c) \wedge \\
(\bar{a} \vee \bar{b} \vee d) & \wedge & (\bar{a} \vee \bar{b} \vee \bar{d}) \wedge \\
(a \vee \bar{c} \vee d) & \wedge & (a \vee \bar{c} \vee \bar{d})
\end{array}$$

# Self-Subsuming Example

## Self-Subsuming Resolution

$$\frac{C \vee l \qquad D \vee \bar{l}}{D} \quad C \subseteq D \qquad\qquad \frac{(a \vee b \vee l) \qquad (a \vee b \vee c \vee \bar{l})}{(a \vee b \vee c)}$$

resolvent $D$ subsumes second antecedent $D \vee \bar{l}$

## Example: Remove literals using self-subsumption

$$\begin{array}{l} (\quad b \vee c) \wedge (\bar{a} \vee b \vee c) \wedge \\ (\bar{a} \vee b \quad) \wedge (a \vee \bar{b} \vee c) \wedge \\ (\bar{a} \vee \bar{b} \vee d) \wedge (\bar{a} \vee \bar{b} \vee \bar{d}) \wedge \\ (a \vee \bar{c} \vee d) \wedge (a \vee \bar{c} \vee \bar{d}) \end{array}$$

# Self-Subsuming Example

## Self-Subsuming Resolution

$$\frac{C \vee l \qquad D \vee \bar{l}}{D} \quad C \subseteq D \qquad\qquad \frac{(a \vee b \vee l) \qquad (a \vee b \vee c \vee \bar{l})}{(a \vee b \vee c)}$$

resolvent $D$ subsumes second antecedent $D \vee \bar{l}$

## Example: Remove literals using self-subsumption

$$\begin{array}{llll}( \quad b \vee c) & \wedge & (\bar{a} \vee b \vee c) & \wedge \\ (\bar{a} \vee b \quad) & \wedge & (a \vee \quad c) & \wedge \\ (\bar{a} \vee \bar{b} \vee d) & \wedge & (\bar{a} \vee \bar{b} \vee \bar{d}) & \wedge \\ (a \vee \bar{c} \vee d) & \wedge & (a \vee \bar{c} \vee \bar{d}) & \end{array}$$

# Self-Subsuming Example

## Self-Subsuming Resolution

$$\frac{C \vee l \qquad D \vee \bar{l}}{D} \quad C \subseteq D \qquad\qquad \frac{(a \vee b \vee l) \qquad (a \vee b \vee c \vee \bar{l})}{(a \vee b \vee c)}$$

resolvent $D$ subsumes second antecedent $D \vee \bar{l}$

## Example: Remove literals using self-subsumption

$$\begin{array}{llll}
(\quad b \vee c) & \wedge & (\bar{a} \vee b \vee c) & \wedge \\
(\bar{a} \vee b \quad) & \wedge & (a \vee \quad c) & \wedge \\
(\bar{a} \vee \bar{b} \quad) & \wedge & (\bar{a} \vee \bar{b} \vee \bar{d}) & \wedge \\
(a \vee \bar{c} \vee d) & \wedge & (a \vee \bar{c} \vee \bar{d}) &
\end{array}$$

# Self-Subsuming Example

## Self-Subsuming Resolution

$$\frac{C \vee l \qquad D \vee \bar{l}}{D} \quad C \subseteq D \qquad\qquad \frac{(a \vee b \vee l) \qquad (a \vee b \vee c \vee \bar{l})}{(a \vee b \vee c)}$$

resolvent $D$ subsumes second antecedent $D \vee \bar{l}$

## Example: Remove literals using self-subsumption

$$\begin{array}{llll}
(\quad b \vee c) & \wedge & (\bar{a} \vee b \vee c) & \wedge \\
(\bar{a} \vee b \quad) & \wedge & (a \vee \quad c) & \wedge \\
(\bar{a} \vee \bar{b} \quad) & \wedge & (\bar{a} \vee \bar{b} \vee \bar{d}) & \wedge \\
(a \vee \bar{c} \quad) & \wedge & (a \vee \bar{c} \vee \bar{d}) &
\end{array}$$

# Self-Subsuming Example

## Self-Subsuming Resolution

$$\frac{C \vee l \qquad D \vee \bar{l}}{D} \quad C \subseteq D \qquad\qquad \frac{(a \vee b \vee l) \qquad (a \vee b \vee c \vee \bar{l})}{(a \vee b \vee c)}$$

resolvent $D$ subsumes second antecedent $D \vee \bar{l}$

## Example: Remove literals using self-subsumption

$$\begin{array}{l}
(\quad b \vee c) \wedge (\bar{a} \vee b \vee c) \wedge \\
(\bar{a} \quad) \wedge (a \vee \quad c) \wedge \\
(\bar{a} \vee \bar{b} \quad) \wedge (\bar{a} \vee \bar{b} \vee \bar{d}) \wedge \\
(a \vee \bar{c} \quad) \wedge (a \vee \bar{c} \vee \bar{d})
\end{array}$$

# Implementing Subsumtion

## Definition (Subsumption)

Clause $C$ subsumes clause $D$ if and only if $C \subset D$.

## Example

The clause $(a \vee b)$ subsumes clause $(a \vee b \vee \bar{c})$.

## Forward Subsumption

**for** each clause $C$ in formula $F$ **do**
  **if** $C$ is subsumed by a clause $D$ in $F \setminus C$ **then**
    remove $C$ from $F$

# Implementing Subsumtion

## Definition (Subsumption)

Clause $C$ subsumes clause $D$ if and only if $C \subset D$.

## Example

The clause $(a \lor b)$ subsumes clause $(a \lor b \lor \bar{c})$.

## Forward Subsumption

**for** each clause $C$ in formula $F$ **do**
  **if** $C$ is subsumed by a clause $D$ in $F \setminus C$ **then**
    remove $C$ from $F$

## Backward Subsumption

**for** each clause $C$ in formula $F$ **do**
  remove all clauses $D$ in $F$ that are subsumed by $C$

# Implementing Subsumtion

**Definition (Subsumption)**

Clause $C$ subsumes clause $D$ if and only if $C \subset D$.

**Example**

The clause $(a \vee b)$ subsumes clause $(a \vee b \vee \bar{c})$.

**Forward Subsumption**

**for** each clause $C$ in formula $F$ **do**
  **if** $C$ is subsumed by a clause $D$ in $F \setminus C$ **then**
    remove $C$ from $F$

**Backward Subsumption**

**for** each clause $C$ in formula $F$ **do**
  pick a literal $l$ in $C$
  remove all clauses $D$ in $F_l$ that are subsumed by $C$

# Variable Elimination

# Variable Elimination [DavisPutnam'60]

## Definition (Resolution)

Given two clauses $C = (x \vee a_1 \vee \cdots \vee a_i)$ and $D = (\bar{x} \vee b_1 \vee \cdots \vee b_j)$, the *resolvent* of $C$ and $D$ on variable $x$ (denoted by $C \otimes_x D$) is $(a_1 \vee \cdots \vee a_i \vee b_1 \vee \cdots \vee b_j)$

Resolution on sets of clauses $F_x$ and $F_{\bar{x}}$ (denoted by $F_x \otimes_x F_{\bar{x}}$) generates all (non-tautological) resolvents of $C \in F_x$ and $D \in F_{\bar{x}}$.

# Variable Elimination [DavisPutnam'60]

## Definition (Resolution)

Given two clauses $C = (x \vee a_1 \vee \dots \vee a_i)$ and $D = (\bar{x} \vee b_1 \vee \dots \vee b_j)$, the *resolvent* of $C$ and $D$ on variable $x$ (denoted by $C \otimes_x D$) is $(a_1 \vee \dots \vee a_i \vee b_1 \vee \dots \vee b_j)$

Resolution on sets of clauses $F_x$ and $F_{\bar{x}}$ (denoted by $F_x \otimes_x F_{\bar{x}}$) generates all (non-tautological) resolvents of $C \in F_x$ and $D \in F_{\bar{x}}$.

## Definition (Variable elimination (VE))

Given a CNF formula $F$, *variable elimination* (or DP resolution) removes a variable $x$ by replacing $F_x$ and $F_{\bar{x}}$ by $F_x \otimes_x F_{\bar{x}}$

# Variable Elimination [DavisPutnam'60]

## Definition (Resolution)

Given two clauses $C = (x \vee a_1 \vee \cdots \vee a_i)$ and $D = (\bar{x} \vee b_1 \vee \cdots \vee b_j)$, the *resolvent* of $C$ and $D$ on variable $x$ (denoted by $C \otimes_x D$) is $(a_1 \vee \cdots \vee a_i \vee b_1 \vee \cdots \vee b_j)$

Resolution on sets of clauses $F_x$ and $F_{\bar{x}}$ (denoted by $F_x \otimes_x F_{\bar{x}}$) generates all (non-tautological) resolvents of $C \in F_x$ and $D \in F_{\bar{x}}$.

## Definition (Variable elimination (VE))

Given a CNF formula $F$, *variable elimination* (or DP resolution) removes a variable $x$ by replacing $F_x$ and $F_{\bar{x}}$ by $F_x \otimes_x F_{\bar{x}}$

## Proof procedure [DavisPutnam60]

VE is a complete proof procedure. Applying VE until fixpoint results in the empty formula (satisfiable) or empty clause (unsatisfiable)

# Example VE by clause distribution [DavisPutnam'60]

**Definition (Variable elimination (VE))**

Given a CNF formula $F$, *variable elimination* (or DP resolution) removes a variable $x$ by replacing $F_x$ and $F_{\bar{x}}$ by $F_x \otimes_x F_{\bar{x}}$

# Example VE by clause distribution [DavisPutnam'60]

## Definition (Variable elimination (VE))

Given a CNF formula $F$, *variable elimination* (or DP resolution) removes a variable $x$ by replacing $F_x$ and $F_{\bar{x}}$ by $F_x \otimes_x F_{\bar{x}}$

## Example of clause distribution

| | $F_x$: $(x \vee c)$ | $F_x$: $(x \vee \bar{d})$ | $F_x$: $(x \vee \bar{a} \vee \bar{b})$ |
|---|---|---|---|
| $F_{\bar{x}}$: $(\bar{x} \vee a)$ | $(a \vee c)$ | $(a \vee d)$ | $(a \vee \bar{a} \vee \bar{b})$ |
| $F_{\bar{x}}$: $(\bar{x} \vee b)$ | $(b \vee c)$ | $(b \vee d)$ | $(b \vee \bar{a} \vee \bar{b})$ |
| $F_{\bar{x}}$: $(\bar{x} \vee \bar{e} \vee f)$ | $(c \vee \bar{e} \vee f)$ | $(d \vee \bar{e} \vee f)$ | $(\bar{a} \vee \bar{b} \vee \bar{e} \vee f)$ |

# Example VE by clause distribution [DavisPutnam'60]

## Definition (Variable elimination (VE))

Given a CNF formula $F$, *variable elimination* (or DP resolution) removes a variable $x$ by replacing $F_x$ and $F_{\bar{x}}$ by $F_x \otimes_x F_{\bar{x}}$

## Example of clause distribution

| | | $F_x$ | |
|---|---|---|---|
| | $(x \vee c)$ | $(x \vee \bar{d})$ | $(x \vee \bar{a} \vee \bar{b})$ |
| $F_{\bar{x}}$: $(\bar{x} \vee a)$ | $(a \vee c)$ | $(a \vee d)$ | ~~$(a \vee \bar{a} \vee \bar{b})$~~ |
| $F_{\bar{x}}$: $(\bar{x} \vee b)$ | $(b \vee c)$ | $(b \vee d)$ | ~~$(b \vee \bar{a} \vee \bar{b})$~~ |
| $F_{\bar{x}}$: $(\bar{x} \vee \bar{e} \vee f)$ | $(c \vee \bar{e} \vee f)$ | $(d \vee \bar{e} \vee f)$ | $(\bar{a} \vee \bar{b} \vee \bar{e} \vee f)$ |

# Example VE by clause distribution [DavisPutnam'60]

## Definition (Variable elimination (VE))

Given a CNF formula $F$, *variable elimination* (or DP resolution) removes a variable $x$ by replacing $F_x$ and $F_{\bar{x}}$ by $F_x \otimes_x F_{\bar{x}}$

## Example of clause distribution

| | $F_x$: $(x \vee c)$ | $(x \vee \bar{d})$ | $(x \vee \bar{a} \vee \bar{b})$ |
|---|---|---|---|
| $F_{\bar{x}}$: $(\bar{x} \vee a)$ | $(a \vee c)$ | $(a \vee d)$ | ~~$(a \vee \bar{a} \vee \bar{b})$~~ |
| $(\bar{x} \vee b)$ | $(b \vee c)$ | $(b \vee d)$ | ~~$(b \vee \bar{a} \vee \bar{b})$~~ |
| $(\bar{x} \vee \bar{e} \vee f)$ | $(c \vee \bar{e} \vee f)$ | $(d \vee \bar{e} \vee f)$ | $(\bar{a} \vee \bar{b} \vee \bar{e} \vee f)$ |

example: $|F_x \otimes F_{\bar{x}}| > |F_x| + |F_{\bar{x}}|$; in general: exponential growth of clauses

# VE by substitution [EenBiere07]

## General idea

Detect gates (or definitions) $x = \mathrm{GATE}(a_1, \ldots, a_n)$ in the formula and use them to reduce the number of added clauses

# VE by substitution [EenBiere07]

## General idea

Detect gates (or definitions) $x = \mathrm{GATE}(a_1, \ldots, a_n)$ in the formula and use them to reduce the number of added clauses

## Possible gates

| gate | $G_x$ | $G_{\bar{x}}$ |
|---|---|---|
| $\mathrm{AND}(a_1, \ldots, a_n)$ | $(x \vee \bar{a}_1 \vee \cdots \vee \bar{a}_n)$ | $(\bar{x} \vee a_1), \ldots, (\bar{x} \vee a_n)$ |
| $\mathrm{OR}(a_1, \ldots, a_n)$ | $(x \vee \bar{a}_1), \ldots, (x \vee \bar{a}_n)$ | $(\bar{x} \vee a_1 \vee \cdots \vee a_n)$ |
| $\mathrm{ITE}(c, t, f)$ | $(x \vee \bar{c} \vee \bar{t}), (x \vee c \vee \bar{f})$ | $(\bar{x} \vee \bar{c} \vee t), (\bar{x} \vee c \vee f)$ |

# VE by substitution [EenBiere07]

## General idea

Detect gates (or definitions) $x = \mathrm{GATE}(a_1, \ldots, a_n)$ in the formula and use them to reduce the number of added clauses

## Possible gates

| gate | $G_x$ | $G_{\bar{x}}$ |
|---|---|---|
| $\mathrm{AND}(a_1, \ldots, a_n)$ | $(x \vee \bar{a}_1 \vee \cdots \vee \bar{a}_n)$ | $(\bar{x} \vee a_1), \ldots, (\bar{x} \vee a_n)$ |
| $\mathrm{OR}(a_1, \ldots, a_n)$ | $(x \vee \bar{a}_1), \ldots, (x \vee \bar{a}_n)$ | $(\bar{x} \vee a_1 \vee \cdots \vee a_n)$ |
| $\mathrm{ITE}(c, t, f)$ | $(x \vee \bar{c} \vee \bar{t}), (x \vee c \vee \bar{f})$ | $(\bar{x} \vee \bar{c} \vee t), (\bar{x} \vee c \vee f)$ |

## Variable elimination by substitution [EenBiere07]

Let $R_x = F_x \setminus G_x$; $R_{\bar{x}} = F_{\bar{x}} \setminus G_{\bar{x}}$.

Replace $F_x \wedge F_{\bar{x}}$ by $G_x \otimes_x R_{\bar{x}} \wedge G_{\bar{x}} \otimes_x R_x$. Always less than $F_x \otimes_x F_{\bar{x}}$ !

# VE by substitution [EenBiere'07]

**Example of gate extraction: $x = \mathrm{AND}(a, b)$**

$$F_x = (x \vee c) \wedge (x \vee \bar{d}) \wedge (x \vee \bar{a} \vee \bar{b})$$
$$F_{\bar{x}} = (\bar{x} \vee a) \wedge (\bar{x} \vee b) \wedge (\bar{x} \vee \bar{e} \vee f)$$

# VE by substitution [EenBiere'07]

## Example of gate extraction: $x = \mathrm{AND}(a, b)$

$$F_x = (x \vee c) \wedge (x \vee \bar{d}) \wedge (x \vee \bar{a} \vee \bar{b})$$
$$F_{\bar{x}} = (\bar{x} \vee a) \wedge (\bar{x} \vee b) \wedge (\bar{x} \vee \bar{e} \vee f)$$

## Example of substitution

| | $R_x$: $(x \vee c)$ | $R_x$: $(x \vee \bar{d})$ | $G_x$: $(x \vee \bar{a} \vee \bar{b})$ |
|---|---|---|---|
| $G_{\bar{x}}$: $(\bar{x} \vee a)$ | $(a \vee c)$ | $(a \vee d)$ | |
| $G_{\bar{x}}$: $(\bar{x} \vee b)$ | $(b \vee c)$ | $(b \vee d)$ | |
| $R_{\bar{x}}$: $(\bar{x} \vee \bar{e} \vee f)$ | | | $(\bar{a} \vee \bar{b} \vee \bar{e} \vee f)$ |

# VE by substitution [EenBiere'07]

## Example of gate extraction: $x = \mathrm{AND}(a, b)$

$$F_x = (x \vee c) \wedge (x \vee \bar{d}) \wedge (x \vee \bar{a} \vee \bar{b})$$
$$F_{\bar{x}} = (\bar{x} \vee a) \wedge (\bar{x} \vee b) \wedge (\bar{x} \vee \bar{e} \vee f)$$

## Example of substitution

| | | $R_x$ | | $G_x$ |
|---|---|---|---|---|
| | | $(x \vee c)$ | $(x \vee \bar{d})$ | $(x \vee \bar{a} \vee \bar{b})$ |
| $G_{\bar{x}}$ | $(\bar{x} \vee a)$ | $(a \vee c)$ | $(a \vee d)$ | |
| | $(\bar{x} \vee b)$ | $(b \vee c)$ | $(b \vee d)$ | |
| $R_{\bar{x}}$ | $(\bar{x} \vee \bar{e} \vee f)$ | | | $(\bar{a} \vee \bar{b} \vee \bar{e} \vee f)$ |

using substitution: $|F_x \otimes F_{\bar{x}}| < |F_x| + |F_{\bar{x}}|$

# Bounded Variable Addition

# Bounded Variable Addition

## Main Idea

Given a CNF formula $F$, can we construct a logically equivalent $F'$ by introducing a new variable $x \notin VAR(F)$ such that $|F'| < |F|$?

# Bounded Variable Addition

## Main Idea

Given a CNF formula $F$, can we construct a logically equivalent $F'$ by introducing a new variable $x \notin VAR(F)$ such that $|F'| < |F|$?

## Reverse of Variable Elimination

For example, replace the clauses

$$\begin{array}{ccc} (a \vee c) & (a \vee d) & \\ (b \vee c) & (b \vee d) & \\ (c \vee \bar{e} \vee f) & (d \vee \bar{e} \vee f) & (\bar{a} \vee \bar{b} \vee \bar{e} \vee f) \end{array}$$

by

$$\begin{array}{ccc} (\bar{x} \vee a) & (\bar{x} \vee b) & (\bar{x} \vee \bar{e} \vee f) \\ (x \vee c) & (x \vee d) & (x \vee \bar{a} \vee \bar{b}) \end{array}$$

# Bounded Variable Addition

## Main Idea

Given a CNF formula $F$, can we construct a logically equivalent $F'$ by introducing a new variable $x \notin VAR(F)$ such that $|F'| < |F|$?

## Reverse of Variable Elimination

For example, replace the clauses

$$\begin{array}{ccc} (a \lor c) & (a \lor d) & \\ (b \lor c) & (b \lor d) & \\ (c \lor \bar{e} \lor f) & (d \lor \bar{e} \lor f) & (\bar{a} \lor \bar{b} \lor \bar{e} \lor f) \end{array}$$

by

$$\begin{array}{ccc} (\bar{x} \lor a) & (\bar{x} \lor b) & (\bar{x} \lor \bar{e} \lor f) \\ (x \lor c) & (x \lor d) & (x \lor \bar{a} \lor \bar{b}) \end{array}$$

Challenge: how to find suitable patterns for replacement?

# Factoring Out Subclauses

## Example

Replace

$$(a \vee b \vee c \vee d) \quad (a \vee b \vee c \vee e) \quad (a \vee b \vee c \vee f)$$

by

$$(x \vee d) \quad (x \vee e) \quad (x \vee f) \quad (\bar{x} \vee a \vee b \vee c)$$

*adds* 1 variable and one clause

*reduces* number of literals by 2

Not compatible with BVE, which would eliminate $x$ immediately!

... *so this does not work* ...

# Bounded Variable Addition

## Smallest Example

Replace

$$(a \lor d) \quad (a \lor e)$$
$$(b \lor d) \quad (b \lor e)$$
$$(c \lor d) \quad (c \lor e)$$

by

$$(\bar{x} \lor a) \quad (\bar{x} \lor b) \quad (\bar{x} \lor c)$$
$$(x \lor d) \quad (x \lor e)$$

*adds* 1 variable

*removes* 1 clause

# Bounded Variable Addition

## Possible Patterns

$$\begin{matrix} (X_1 \vee L_1) & \dots & (X_1 \vee L_k) \\ \vdots & & \vdots \\ (X_n \vee L_1) & \dots & (X_n \vee L_k) \end{matrix} \quad \equiv \quad \bigwedge_{i=1}^{n} \bigwedge_{j=1}^{k} (X_i \vee L_j)$$

*replaced by* $\bigwedge_{i=1}^{n} (y \vee X_i) \wedge \bigwedge_{j=1}^{k} (\bar{y} \vee L_j)$

- Every $k$ clauses share sets of literals $L_j$
- There are $n$ sets of literals $X_i$ that appear in clauses with $L_j$

# Bounded Variable Addition

## Possible Patterns

$$\begin{matrix} (X_1 \vee L_1) & \dots & (X_1 \vee L_k) \\ \vdots & & \vdots \\ (X_n \vee L_1) & \dots & (X_n \vee L_k) \end{matrix} \quad \equiv \quad \bigwedge_{i=1}^{n} \bigwedge_{j=1}^{k} (X_i \vee L_j)$$

*replaced by* $\bigwedge_{i=1}^{n} (y \vee X_i) \wedge \bigwedge_{j=1}^{k} (\bar{y} \vee L_j)$

- Every $k$ clauses share sets of literals $L_j$
- There are $n$ sets of literals $X_i$ that appear in clauses with $L_j$
- Reduction: $nk - n - k$ clauses are removed by replacement

# Bounded Variable Addition

## Possible Patterns

$$\begin{matrix} (X_1 \vee L_1) & \dots & (X_1 \vee L_k) \\ \vdots & & \vdots \\ (X_n \vee L_1) & \dots & (X_n \vee L_k) \end{matrix} \quad \equiv \quad \bigwedge_{i=1}^{n} \bigwedge_{j=1}^{k} (X_i \vee L_j)$$

*replaced by* $\bigwedge_{i=1}^{n} (y \vee X_i) \wedge \bigwedge_{j=1}^{k} (\bar{y} \vee L_j)$

- Every $k$ clauses share sets of literals $L_j$
- There are $n$ sets of literals $X_i$ that appear in clauses with $L_j$
- Reduction: $nk - n - k$ clauses are removed by replacement

# Bounded Variable Addition on AtMostOneZero (1)

## Example encoding of AtMostOneZero $(x_1, x_2, \ldots, x_n)$

$$
\begin{aligned}
&(x_1 \lor x_2) \land (x_9 \lor x_{10}) \land (x_8 \lor x_{10}) \land (x_7 \lor x_{10}) \land (x_6 \lor x_{10}) \land \\
&(x_1 \lor x_3) \land (x_2 \lor x_3) \land (x_8 \lor x_9) \land (x_7 \lor x_9) \land (x_6 \lor x_9) \land \\
&(x_1 \lor x_4) \land (x_2 \lor x_4) \land (x_3 \lor x_4) \land (x_7 \lor x_8) \land (x_6 \lor x_8) \land \\
&(x_1 \lor x_5) \land (x_2 \lor x_5) \land (x_3 \lor x_5) \land (x_4 \lor x_5) \land (x_6 \lor x_7) \land \\
&(x_1 \lor x_6) \land (x_2 \lor x_6) \land (x_3 \lor x_6) \land (x_4 \lor x_6) \land (x_5 \lor x_6) \land \\
&(x_1 \lor x_7) \land (x_2 \lor x_7) \land (x_3 \lor x_7) \land (x_4 \lor x_7) \land (x_5 \lor x_7) \land \\
&(x_1 \lor x_8) \land (x_2 \lor x_8) \land (x_3 \lor x_8) \land (x_4 \lor x_8) \land (x_5 \lor x_8) \land \\
&(x_1 \lor x_9) \land (x_2 \lor x_9) \land (x_3 \lor x_9) \land (x_4 \lor x_9) \land (x_5 \lor x_9) \land \\
&(x_1 \lor x_{10}) \land (x_2 \lor x_{10}) \land (x_3 \lor x_{10}) \land (x_4 \lor x_{10}) \land (x_5 \lor x_{10})
\end{aligned}
$$

# Bounded Variable Addition on AtMostOneZero (1)

## Example encoding of AtMostOneZero $(x_1, x_2, \ldots, x_n)$

$$(x_1 \vee x_2) \wedge (x_9 \vee x_{10}) \wedge (x_8 \vee x_{10}) \wedge (x_7 \vee x_{10}) \wedge (x_6 \vee x_{10}) \wedge$$
$$(x_1 \vee x_3) \wedge (x_2 \vee x_3) \wedge (x_8 \vee x_9) \wedge (x_7 \vee x_9) \wedge (x_6 \vee x_9) \wedge$$
$$(x_1 \vee x_4) \wedge (x_2 \vee x_4) \wedge (x_3 \vee x_4) \wedge (x_7 \vee x_8) \wedge (x_6 \vee x_8) \wedge$$
$$(x_1 \vee x_5) \wedge (x_2 \vee x_5) \wedge (x_3 \vee x_5) \wedge (x_4 \vee x_5) \wedge (x_6 \vee x_7) \wedge$$
$$(x_1 \vee x_6) \wedge (x_2 \vee x_6) \wedge (x_3 \vee x_6) \wedge (x_4 \vee x_6) \wedge (x_5 \vee x_6) \wedge$$
$$(x_1 \vee x_7) \wedge (x_2 \vee x_7) \wedge (x_3 \vee x_7) \wedge (x_4 \vee x_7) \wedge (x_5 \vee x_7) \wedge$$
$$(x_1 \vee x_8) \wedge (x_2 \vee x_8) \wedge (x_3 \vee x_8) \wedge (x_4 \vee x_8) \wedge (x_5 \vee x_8) \wedge$$
$$(x_1 \vee x_9) \wedge (x_2 \vee x_9) \wedge (x_3 \vee x_9) \wedge (x_4 \vee x_9) \wedge (x_5 \vee x_9) \wedge$$
$$(x_1 \vee x_{10}) \wedge (x_2 \vee x_{10}) \wedge (x_3 \vee x_{10}) \wedge (x_4 \vee x_{10}) \wedge (x_5 \vee x_{10})$$

Replace $(x_i \vee x_j)$ with $i \in \{1..5\}, j \in \{6..10\}$ by $(x_i \vee y)$, $(x_j \vee \bar{y})$

# Bounded Variable Addition on AtMostOneZero (2)

## Example encoding of AtMostOneZero $(x_1, x_2, \ldots, x_n)$

$$
\begin{array}{l}
(x_1 \vee x_2) \wedge (x_9 \vee x_{10}) \wedge (x_8 \vee x_{10}) \wedge (x_7 \vee x_{10}) \wedge (x_6 \vee x_{10}) \wedge \\
(x_1 \vee x_3) \wedge (x_2 \vee x_3) \wedge (x_8 \vee x_9) \wedge (x_7 \vee x_9) \wedge (x_6 \vee x_9) \wedge \\
(x_1 \vee x_4) \wedge (x_2 \vee x_4) \wedge (x_3 \vee x_4) \wedge (x_7 \vee x_8) \wedge (x_6 \vee x_8) \wedge \\
(x_1 \vee x_5) \wedge (x_2 \vee x_5) \wedge (x_3 \vee x_5) \wedge (x_4 \vee x_5) \wedge (x_6 \vee x_7) \wedge \\
(x_1 \vee y) \wedge (x_2 \vee y) \wedge (x_3 \vee y) \wedge (x_4 \vee y) \wedge (x_5 \vee y) \wedge \\
(x_6 \vee \bar{y}) \wedge (x_7 \vee \bar{y}) \wedge (x_8 \vee \bar{y}) \wedge (x_9 \vee \bar{y}) \wedge (x_{10} \vee \bar{y})
\end{array}
$$

# Bounded Variable Addition on AtMostOneZero (2)

## Example encoding of AtMostOneZero $(x_1, x_2, \ldots, x_n)$

$$(x_1 \vee x_2) \wedge (x_9 \vee x_{10}) \wedge (x_8 \vee x_{10}) \wedge (x_7 \vee x_{10}) \wedge (x_6 \vee x_{10}) \wedge$$
$$(x_1 \vee x_3) \wedge (x_2 \vee x_3) \wedge (x_8 \vee x_9) \wedge (x_7 \vee x_9) \wedge (x_6 \vee x_9) \wedge$$
$$(x_1 \vee x_4) \wedge (x_2 \vee x_4) \wedge (x_3 \vee x_4) \wedge (x_7 \vee x_8) \wedge (x_6 \vee x_8) \wedge$$
$$(x_1 \vee x_5) \wedge (x_2 \vee x_5) \wedge (x_3 \vee x_5) \wedge (x_4 \vee x_5) \wedge (x_6 \vee x_7) \wedge$$
$$(x_1 \vee y) \wedge (x_2 \vee y) \wedge (x_3 \vee y) \wedge (x_4 \vee y) \wedge (x_5 \vee y) \wedge$$
$$(x_6 \vee \bar{y}) \wedge (x_7 \vee \bar{y}) \wedge (x_8 \vee \bar{y}) \wedge (x_9 \vee \bar{y}) \wedge (x_{10} \vee \bar{y})$$

## Replace matched pattern

$$(x_1 \vee z) \wedge (x_2 \vee z) \wedge (x_3 \vee z) \wedge$$
$$(x_4 \vee \bar{z}) \wedge (x_5 \vee \bar{z}) \wedge (y \vee \bar{z})$$

# Bounded Variable Addition on AtMostOneZero (3)

## Example encoding of AtMostOneZero $(x_1, x_2, \ldots, x_n)$

$$(x_1 \vee x_2) \wedge (x_9 \vee x_{10}) \wedge (x_8 \vee x_{10}) \wedge (x_7 \vee x_{10}) \wedge (x_6 \vee x_{10}) \wedge$$
$$(x_1 \vee x_3) \wedge (x_2 \vee x_3) \wedge (x_8 \vee x_9) \wedge (x_7 \vee x_9) \wedge (x_6 \vee x_9) \wedge$$
$$(x_1 \vee z) \wedge (x_2 \vee z) \wedge (x_3 \vee z) \wedge (x_7 \vee x_8) \wedge (x_6 \vee x_8) \wedge$$
$$(x_4 \vee \bar{z}) \wedge (x_5 \vee \bar{z}) \wedge (y \vee \bar{z}) \wedge (x_4 \vee x_5) \wedge (x_6 \vee x_7) \wedge$$
$$(x_4 \vee y) \wedge (x_5 \vee y) \wedge (x_6 \vee \bar{y}) \wedge (x_7 \vee \bar{y}) \wedge (x_8 \vee \bar{y})$$
$$(x_9 \vee \bar{y}) \wedge (x_{10} \vee \bar{y})$$

# Bounded Variable Addition on AtMostOneZero (3)

## Example encoding of AtMostOneZero $(x_1, x_2, \ldots, x_n)$

$$(x_1 \vee x_2) \wedge (x_9 \vee x_{10}) \wedge (x_8 \vee x_{10}) \wedge (x_7 \vee x_{10}) \wedge (x_6 \vee x_{10}) \wedge$$
$$(x_1 \vee x_3) \wedge (x_2 \vee x_3) \wedge (x_8 \vee x_9) \wedge (x_7 \vee x_9) \wedge (x_6 \vee x_9) \wedge$$
$$(x_1 \vee z) \wedge (x_2 \vee z) \wedge (x_3 \vee z) \wedge (x_7 \vee x_8) \wedge (x_6 \vee x_8) \wedge$$
$$(x_4 \vee \bar{z}) \wedge (x_5 \vee \bar{z}) \wedge (y \vee \bar{z}) \wedge (x_4 \vee x_5) \wedge (x_6 \vee x_7) \wedge$$
$$(x_4 \vee y) \wedge (x_5 \vee y) \wedge (x_6 \vee \bar{y}) \wedge (x_7 \vee \bar{y}) \wedge (x_8 \vee \bar{y})$$
$$(x_9 \vee \bar{y}) \wedge (x_{10} \vee \bar{y})$$

## Replace matched pattern

$$(x_6 \vee w) \wedge (x_7 \vee w) \wedge (x_8 \vee w) \wedge$$
$$(x_9 \vee \bar{w}) \wedge (x_{10} \vee \bar{w}) \wedge (\bar{y} \vee \bar{w})$$

# Blocked Clause Elimination

# Blocked Clauses [Kullmann'99]

**Definition (Blocking literal)**

A literal $l$ in a clause $C$ of a CNF $F$ blocks $C$ w.r.t. $F$ if for every clause $D \in F$ with $\bar{l} \in D$, the resolvent $(C \setminus \{l\}) \cup (D \setminus \{\bar{l}\})$ obtained from resolving $C$ and $D$ on $l$ is a tautology.

With respect to a fixed CNF and its clauses we have:

**Definition (Blocked clause)**

A clause is blocked if it contains a literal that blocks it.

# Blocked Clauses [Kullmann'99]

## Definition (Blocking literal)

A literal $l$ in a clause $C$ of a CNF $F$ blocks $C$ w.r.t. $F$ if for every clause $D \in F$ with $\bar{l} \in D$, the resolvent $(C \setminus \{l\}) \cup (D \setminus \{\bar{l}\})$ obtained from resolving $C$ and $D$ on $l$ is a tautology.

With respect to a fixed CNF and its clauses we have:

## Definition (Blocked clause)

A clause is blocked if it contains a literal that blocks it.

## Example

Consider the formula $(a \vee b) \wedge (a \vee \bar{b} \vee \bar{c}) \wedge (\bar{a} \vee c)$.
First clause is not blocked.
Second clause is blocked by both $a$ and $\bar{c}$. Third clause is blocked by $c$

# Blocked Clauses [Kullmann'99]

**Definition (Blocking literal)**

A literal $l$ in a clause $C$ of a CNF $F$ blocks $C$ w.r.t. $F$ if for every clause $D \in F$ with $\bar{l} \in D$, the resolvent $(C \setminus \{l\}) \cup (D \setminus \{\bar{l}\})$ obtained from resolving $C$ and $D$ on $l$ is a tautology.

With respect to a fixed CNF and its clauses we have:

**Definition (Blocked clause)**

A clause is blocked if it contains a literal that blocks it.

**Example**

Consider the formula $(a \vee b) \wedge (a \vee \bar{b} \vee \bar{c}) \wedge (\bar{a} \vee c)$.
First clause is not blocked.
Second clause is blocked by both $a$ and $\bar{c}$. Third clause is blocked by $c$

**Proposition**

Removal of an arbitrary blocked clause preserves satisfiability.

# Blocked Clause Elimination (BCE)

## Definition (BCE)

While there is a blocked clause $C$ in a CNF $F$, remove $C$ from $F$.

## Example

Consider $(a \vee b) \wedge (a \vee \bar{b} \vee \bar{c}) \wedge (\bar{a} \vee c)$.
After removing either $(a \vee \bar{b} \vee \bar{c})$ or $(\bar{a} \vee c)$,
the clause $(a \vee b)$ becomes blocked (*no clause with either $\bar{b}$ or $\bar{a}$*).
An extreme case in which BCE removes all clauses of a formula!

# Blocked Clause Elimination (BCE)

## Definition (BCE)

While there is a blocked clause $C$ in a CNF $F$, remove $C$ from $F$.

## Example

Consider $(a \vee b) \wedge (a \vee \bar{b} \vee \bar{c}) \wedge (\bar{a} \vee c)$.
After removing either $(a \vee \bar{b} \vee \bar{c})$ or $(\bar{a} \vee c)$,
the clause $(a \vee b)$ becomes blocked (*no clause with either* $\bar{b}$ *or* $\bar{a}$).
An extreme case in which BCE removes all clauses of a formula!

## Proposition

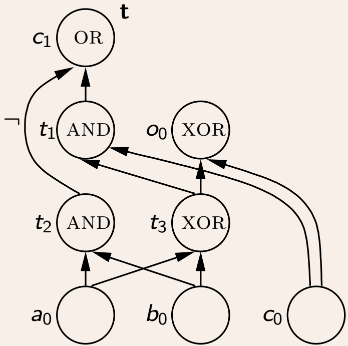
BCE is confluent, i.e., has a unique fixpoint

- Blocked clauses stay blocked w.r.t. removal

# BCE very effective on circuits [JärvisaloBiereHeule'10]

BCE converts the Tseitin encoding to Plaisted Greenbaum
BCE simulates Pure literal elimination, Cone of influence and much more

## Example of circuit simplification by BCE on Tseitin encoding



# BCE very effective on circuits [JärvisaloBiereHeule'10]

BCE converts the Tseitin encoding to Plaisted Greenbaum
BCE simulates Pure literal elimination, Cone of influence and much more

## Example of circuit simplification by BCE on Tseitin encoding

$(c_1)$

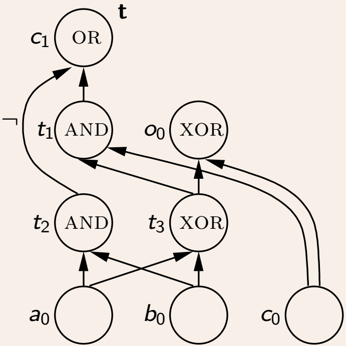
$(\bar{c}_1 \vee t_1 \vee \bar{t}_2)$
$(c_1 \vee \bar{t}_1)$
$(c_1 \vee t_2)$

$(\bar{o}_0 \vee t_3 \vee c_0)$
$(\bar{o}_0 \vee \bar{t}_3 \vee \bar{c}_0)$
$(o_0 \vee t_3 \vee \bar{c}_0)$
$(o_0 \vee \bar{t}_3 \vee c_0)$

$(t_1 \vee \bar{t}_3 \vee \bar{c}_0)$
$(\bar{t}_1 \vee t_3)$
$(\bar{t}_1 \vee c_0)$

$(t_2 \vee \bar{a}_0 \vee \bar{b}_0)$
$(\bar{t}_2 \vee a_0)$
$(\bar{t}_2 \vee b_0)$

$(\bar{t}_3 \vee a_0 \vee b_0)$
$(\bar{t}_3 \vee \bar{a}_0 \vee \bar{b}_0)$
$(t_3 \vee a_0 \vee \bar{b}_0)$
$(t_3 \vee \bar{a}_0 \vee b_0)$



# BCE very effective on circuits [JärvisaloBiereHeule'10]

BCE converts the Tseitin encoding to Plaisted Greenbaum
BCE simulates Pure literal elimination, Cone of influence and much more

## Example of circuit simplification by BCE on Tseitin encoding

$(c_1)$

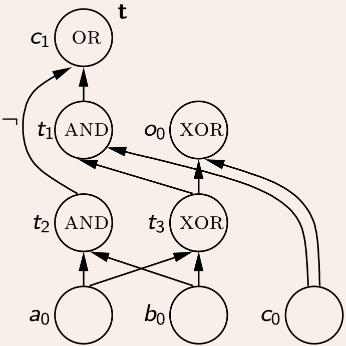
$(\bar{c}_1 \vee t_1 \vee \bar{t}_2)$
~~$(c_1 \vee \bar{t}_1)$~~
~~$(c_1 \vee t_2)$~~

$(\bar{o}_0 \vee t_3 \vee c_0)$
$(\bar{o}_0 \vee \bar{t}_3 \vee \bar{c}_0)$
$(o_0 \vee t_3 \vee \bar{c}_0)$
$(o_0 \vee \bar{t}_3 \vee c_0)$

$(t_1 \vee \bar{t}_3 \vee \bar{c}_0)$
$(\bar{t}_1 \vee t_3)$
$(\bar{t}_1 \vee c_0)$

$(t_2 \vee \bar{a}_0 \vee \bar{b}_0)$
$(\bar{t}_2 \vee a_0)$
$(\bar{t}_2 \vee b_0)$

$(\bar{t}_3 \vee a_0 \vee b_0)$
$(\bar{t}_3 \vee \bar{a}_0 \vee \bar{b}_0)$
$(t_3 \vee a_0 \vee \bar{b}_0)$
$(t_3 \vee \bar{a}_0 \vee b_0)$



# BCE very effective on circuits [JärvisaloBiereHeule'10]

BCE converts the Tseitin encoding to Plaisted Greenbaum
BCE simulates Pure literal elimination, Cone of influence and much more

## Example of circuit simplification by BCE on Tseitin encoding

$(c_1)$

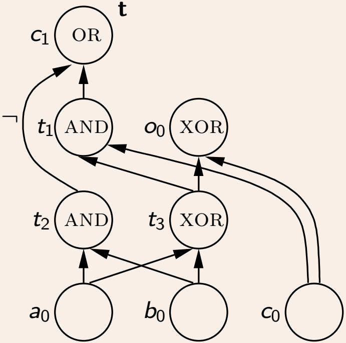
$(\bar{c}_1 \vee t_1 \vee \bar{t}_2)$
~~$(c_1 \vee \bar{t}_1)$~~
~~$(c_1 \vee t_2)$~~

~~$(\bar{o}_0 \vee t_3 \vee c_0)$~~
~~$(\bar{o}_0 \vee \bar{t}_3 \vee \bar{c}_0)$~~
~~$(o_0 \vee t_3 \vee \bar{c}_0)$~~
~~$(o_0 \vee \bar{t}_3 \vee c_0)$~~

$(t_1 \vee \bar{t}_3 \vee \bar{c}_0)$
$(\bar{t}_1 \vee t_3)$
$(\bar{t}_1 \vee c_0)$

$(t_2 \vee \bar{a}_0 \vee \bar{b}_0)$
$(\bar{t}_2 \vee a_0)$
$(\bar{t}_2 \vee b_0)$

$(\bar{t}_3 \vee a_0 \vee b_0)$
$(\bar{t}_3 \vee \bar{a}_0 \vee \bar{b}_0)$
$(t_3 \vee a_0 \vee \bar{b}_0)$
$(t_3 \vee \bar{a}_0 \vee b_0)$



# BCE very effective on circuits [JärvisaloBiereHeule'10]

BCE converts the Tseitin encoding to Plaisted Greenbaum
BCE simulates Pure literal elimination, Cone of influence and much more

## Example of circuit simplification by BCE on Tseitin encoding

$(c_1)$

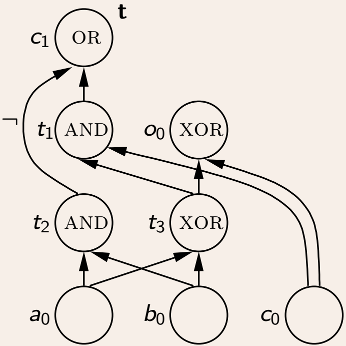
$(\bar{c}_1 \lor t_1 \lor \bar{t}_2)$
~~$(c_1 \lor \bar{t}_1)$~~
~~$(c_1 \lor t_2)$~~

~~$(\bar{o}_0 \lor t_3 \lor c_0)$~~
~~$(\bar{o}_0 \lor \bar{t}_3 \lor \bar{c}_0)$~~
~~$(o_0 \lor t_3 \lor \bar{c}_0)$~~
~~$(o_0 \lor \bar{t}_3 \lor c_0)$~~

~~$(t_1 \lor \bar{t}_3 \lor \bar{c}_0)$~~
$(\bar{t}_1 \lor t_3)$
$(\bar{t}_1 \lor c_0)$

$(t_2 \lor \bar{a}_0 \lor \bar{b}_0)$
$(\bar{t}_2 \lor a_0)$
$(\bar{t}_2 \lor b_0)$

$(\bar{t}_3 \lor a_0 \lor b_0)$
$(\bar{t}_3 \lor \bar{a}_0 \lor \bar{b}_0)$
$(t_3 \lor a_0 \lor \bar{b}_0)$
$(t_3 \lor \bar{a}_0 \lor b_0)$



# BCE very effective on circuits [JärvisaloBiereHeule'10]

BCE converts the Tseitin encoding to Plaisted Greenbaum

BCE simulates Pure literal elimination, Cone of influence and much more

## Example of circuit simplification by BCE on Tseitin encoding

$(c_1)$

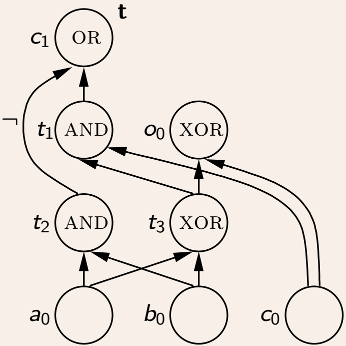
$(\bar{c}_1 \vee t_1 \vee \bar{t}_2)$
~~$(c_1 \vee \bar{t}_1)$~~
~~$(c_1 \vee t_2)$~~

~~$(\bar{o}_0 \vee t_3 \vee c_0)$~~
~~$(\bar{o}_0 \vee \bar{t}_3 \vee \bar{c}_0)$~~
~~$(o_0 \vee t_3 \vee \bar{c}_0)$~~
~~$(o_0 \vee \bar{t}_3 \vee c_0)$~~

~~$(t_1 \vee \bar{t}_3 \vee \bar{c}_0)$~~
$(\bar{t}_1 \vee t_3)$
$(\bar{t}_1 \vee c_0)$

$(t_2 \vee \bar{a}_0 \vee \bar{b}_0)$
~~$(\bar{t}_2 \vee a_0)$~~
~~$(\bar{t}_2 \vee b_0)$~~

$(\bar{t}_3 \vee a_0 \vee b_0)$
$(\bar{t}_3 \vee \bar{a}_0 \vee \bar{b}_0)$
$(t_3 \vee a_0 \vee \bar{b}_0)$
$(t_3 \vee \bar{a}_0 \vee b_0)$



# BCE very effective on circuits [JärvisaloBiereHeule'10]

BCE converts the Tseitin encoding to Plaisted Greenbaum

BCE simulates Pure literal elimination, Cone of influence and much more

## Example of circuit simplification by BCE on Tseitin encoding

$(c_1)$

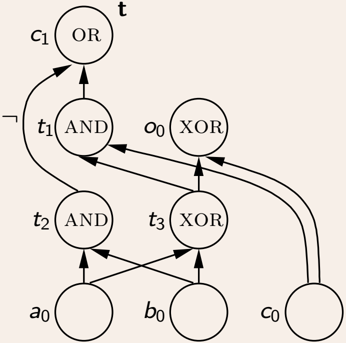
$(\bar{c}_1 \lor t_1 \lor \bar{t}_2)$
~~$(c_1 \lor \bar{t}_1)$~~
~~$(c_1 \lor t_2)$~~

~~$(\bar{o}_0 \lor t_3 \lor c_0)$~~
~~$(\bar{o}_0 \lor \bar{t}_3 \lor \bar{c}_0)$~~
~~$(o_0 \lor t_3 \lor \bar{c}_0)$~~
~~$(o_0 \lor \bar{t}_3 \lor c_0)$~~

~~$(t_1 \lor \bar{t}_3 \lor \bar{c}_0)$~~
$(\bar{t}_1 \lor t_3)$
$(\bar{t}_1 \lor c_0)$

$(t_2 \lor \bar{a}_0 \lor \bar{b}_0)$
~~$(\bar{t}_2 \lor a_0)$~~
~~$(\bar{t}_2 \lor b_0)$~~

$(\bar{t}_3 \lor a_0 \lor b_0)$
$(\bar{t}_3 \lor \bar{a}_0 \lor \bar{b}_0)$
~~$(t_3 \lor a_0 \lor \bar{b}_0)$~~
~~$(t_3 \lor \bar{a}_0 \lor b_0)$~~



# BCE very effective on circuits [JärvisaloBiereHeule'10]

BCE converts the Tseitin encoding to Plaisted Greenbaum

BCE simulates Pure literal elimination, Cone of influence and much more

## Example of circuit simplification by BCE on Tseitin encoding

$(c_1)$

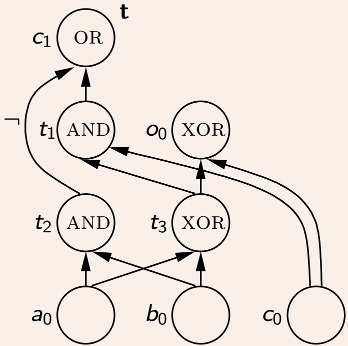
$(\bar{c}_1 \vee t_1 \vee \bar{t}_2)$
~~$(c_1 \vee \bar{t}_1)$~~
~~$(c_1 \vee t_2)$~~

~~$(\bar{o}_0 \vee t_3 \vee c_0)$~~
~~$(\bar{o}_0 \vee \bar{t}_3 \vee \bar{c}_0)$~~
~~$(o_0 \vee t_3 \vee \bar{c}_0)$~~
~~$(o_0 \vee \bar{t}_3 \vee c_0)$~~

~~$(t_1 \vee \bar{t}_3 \vee \bar{c}_0)$~~
$(\bar{t}_1 \vee t_3)$
$(\bar{t}_1 \vee c_0)$

~~$(t_2 \vee \bar{a}_0 \vee \bar{b}_0)$~~
~~$(\bar{t}_2 \vee a_0)$~~
~~$(\bar{t}_2 \vee b_0)$~~

$(\bar{t}_3 \vee a_0 \vee b_0)$
$(\bar{t}_3 \vee \bar{a}_0 \vee \bar{b}_0)$
~~$(t_3 \vee a_0 \vee \bar{b}_0)$~~
~~$(t_3 \vee \bar{a}_0 \vee b_0)$~~



# BCE very effective on circuits [JärvisaloBiereHeule'10]

BCE converts the Tseitin encoding to Plaisted Greenbaum

BCE simulates Pure literal elimination, Cone of influence and much more

## Example of circuit simplification by BCE on Tseitin encoding

$(c_1)$

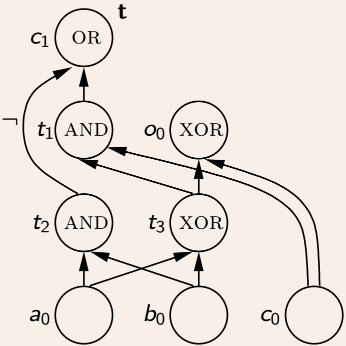
~~$(\bar{c}_1 \vee t_1 \vee \bar{t}_2)$~~
~~$(c_1 \vee \bar{t}_1)$~~
~~$(c_1 \vee t_2)$~~

~~$(\bar{o}_0 \vee t_3 \vee c_0)$~~
~~$(\bar{o}_0 \vee \bar{t}_3 \vee \bar{c}_0)$~~
~~$(o_0 \vee t_3 \vee \bar{c}_0)$~~
~~$(o_0 \vee \bar{t}_3 \vee c_0)$~~

~~$(t_1 \vee \bar{t}_3 \vee \bar{c}_0)$~~
$(\bar{t}_1 \vee t_3)$
$(\bar{t}_1 \vee c_0)$

~~$(t_2 \vee \bar{a}_0 \vee \bar{b}_0)$~~
~~$(\bar{t}_2 \vee a_0)$~~
~~$(\bar{t}_2 \vee b_0)$~~

$(\bar{t}_3 \vee a_0 \vee b_0)$
$(\bar{t}_3 \vee \bar{a}_0 \vee \bar{b}_0)$
~~$(t_3 \vee a_0 \vee \bar{b}_0)$~~
~~$(t_3 \vee \bar{a}_0 \vee b_0)$~~



# BCE very effective on circuits [JärvisaloBiereHeule'10]

BCE converts the Tseitin encoding to Plaisted Greenbaum
BCE simulates Pure literal elimination, Cone of influence and much more

## Example of circuit simplification by BCE on Tseitin encoding

~~$(c_1)$~~

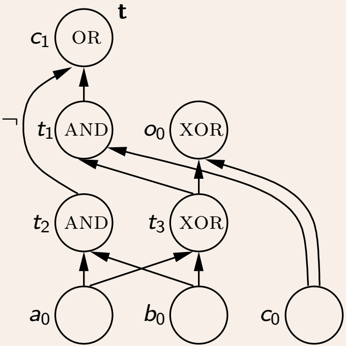
~~$(\bar{c}_1 \vee t_1 \vee \bar{t}_2)$~~
~~$(c_1 \vee \bar{t}_1)$~~
~~$(c_1 \vee t_2)$~~

~~$(\bar{o}_0 \vee t_3 \vee c_0)$~~
~~$(\bar{o}_0 \vee \bar{t}_3 \vee \bar{c}_0)$~~
~~$(o_0 \vee t_3 \vee \bar{c}_0)$~~
~~$(o_0 \vee \bar{t}_3 \vee c_0)$~~

~~$(t_1 \vee \bar{t}_3 \vee \bar{c}_0)$~~
$(\bar{t}_1 \vee t_3)$
$(\bar{t}_1 \vee c_0)$

~~$(t_2 \vee \bar{a}_0 \vee \bar{b}_0)$~~
~~$(\bar{t}_2 \vee a_0)$~~
~~$(\bar{t}_2 \vee b_0)$~~

$(\bar{t}_3 \vee a_0 \vee b_0)$
$(\bar{t}_3 \vee \bar{a}_0 \vee \bar{b}_0)$
~~$(t_3 \vee a_0 \vee \bar{b}_0)$~~
~~$(t_3 \vee \bar{a}_0 \vee b_0)$~~



# BCE very effective on circuits [JärvisaloBiereHeule'10]

BCE converts the Tseitin encoding to Plaisted Greenbaum
BCE simulates Pure literal elimination, Cone of influence and much more

## Example of circuit simplification by BCE on Tseitin encoding

~~$(c_1)$~~

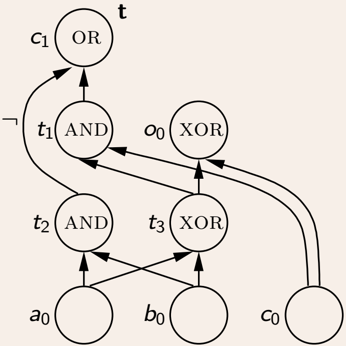
~~$(\bar{c}_1 \vee t_1 \vee \bar{t}_2)$~~
~~$(c_1 \vee \bar{t}_1)$~~
~~$(c_1 \vee t_2)$~~

~~$(\bar{o}_0 \vee t_3 \vee c_0)$~~
~~$(\bar{o}_0 \vee \bar{t}_3 \vee \bar{c}_0)$~~
~~$(o_0 \vee t_3 \vee \bar{c}_0)$~~
~~$(o_0 \vee \bar{t}_3 \vee c_0)$~~

~~$(t_1 \vee \bar{t}_3 \vee \bar{c}_0)$~~
~~$(\bar{t}_1 \vee t_3)$~~
~~$(\bar{t}_1 \vee c_0)$~~

~~$(t_2 \vee \bar{a}_0 \vee \bar{b}_0)$~~
~~$(\bar{t}_2 \vee a_0)$~~
~~$(\bar{t}_2 \vee b_0)$~~

$(\bar{t}_3 \vee a_0 \vee b_0)$
$(\bar{t}_3 \vee \bar{a}_0 \vee \bar{b}_0)$
~~$(t_3 \vee a_0 \vee \bar{b}_0)$~~
~~$(t_3 \vee \bar{a}_0 \vee b_0)$~~



# BCE very effective on circuits [JärvisaloBiereHeule'10]

BCE converts the Tseitin encoding to Plaisted Greenbaum
BCE simulates Pure literal elimination, Cone of influence and much more

## Example of circuit simplification by BCE on Tseitin encoding

~~$(c_1)$~~

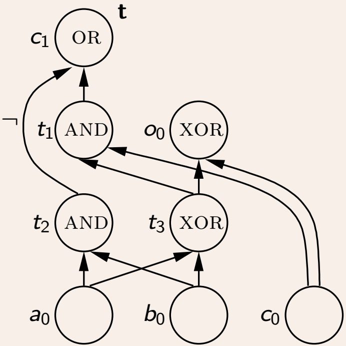
~~$(\bar{c}_1 \vee t_1 \vee \bar{t}_2)$~~
~~$(c_1 \vee \bar{t}_1)$~~
~~$(c_1 \vee t_2)$~~

~~$(\bar{o}_0 \vee t_3 \vee c_0)$~~
~~$(\bar{o}_0 \vee \bar{t}_3 \vee \bar{c}_0)$~~
~~$(o_0 \vee t_3 \vee \bar{c}_0)$~~
~~$(o_0 \vee \bar{t}_3 \vee c_0)$~~

~~$(t_1 \vee \bar{t}_3 \vee \bar{c}_0)$~~
~~$(\bar{t}_1 \vee t_3)$~~
~~$(\bar{t}_1 \vee c_0)$~~

~~$(t_2 \vee \bar{a}_0 \vee \bar{b}_0)$~~
~~$(\bar{t}_2 \vee a_0)$~~
~~$(\bar{t}_2 \vee b_0)$~~

~~$(\bar{t}_3 \vee a_0 \vee b_0)$~~
~~$(\bar{t}_3 \vee \bar{a}_0 \vee \bar{b}_0)$~~
~~$(t_3 \vee a_0 \vee \bar{b}_0)$~~
~~$(t_3 \vee \bar{a}_0 \vee b_0)$~~

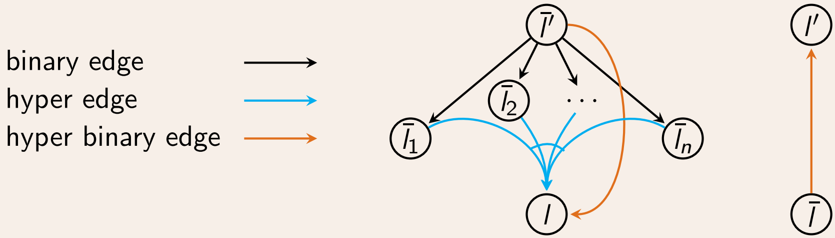


# Hyper Binary Resolution

# Hyper Binary Resolution [Bacchus-AAAI02]

## Definition (Hyper Binary Resolution Rule)

$$\frac{(l \vee l_1 \vee l_2 \vee \cdots \vee l_n) \quad (\bar{l}_1 \vee l') \quad (\bar{l}_2 \vee l') \quad \ldots \quad (\bar{l}_n \vee l')}{(l \vee l')}$$




Hyper Binary Resolution Rule:

- combines multiple resolution steps into one
- uses one n-ary clauses and multiple binary clauses
- special case hyper unary resolution where $l = l'$

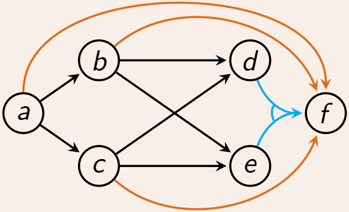
# Hyper Binary Resolution (HBR)

## Definition (Hyper Binary Resolution)

Apply the hyper binary resolution rule until fixpoint

## Example

Consider $(\bar{a} \vee b) \wedge (\bar{a} \vee c) \wedge (\bar{b} \vee d) \wedge (\bar{b} \vee e) \wedge (\bar{c} \vee d) \wedge (\bar{c} \vee e) \wedge (\bar{d} \vee \bar{e} \vee f)$.




hyper binary resolvents:
$(\bar{a} \vee f), (\bar{b} \vee f), (\bar{c} \vee f)$

HBR is confluent, i.e., has a unique fixpoint

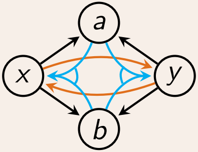
# Structural Hashing of AND-gates via HBR

| gate $g$ | $g \Rightarrow f(g_1, \ldots g_n)$ "positive" | $g \Leftarrow f(g_1, \ldots g_n)$ "negative" |
|---|---|---|
| $g := \text{OR}(g_1, \ldots, g_n)$ | $(\bar{g} \vee g_1 \vee \cdots \vee g_n)$ | $(g \vee \bar{g_1}),\ldots,(g \vee \bar{g_n})$ |
| $g := \text{AND}(g_1, \ldots, g_n)$ | $(\bar{g} \vee g_1),\ldots,(\bar{g} \vee g_n)$ | $(g \vee \bar{g_1} \vee \cdots \vee \bar{g_n})$ |
| $g := \text{XOR}(g_1, g_2)$ | $(\bar{g} \vee \bar{g_1} \vee \bar{g_2})$, $(\bar{g} \vee g_1 \vee g_2)$ | $(g \vee \bar{g_1} \vee g_2)$, $(g \vee g_1 \vee \bar{g_2})$ |
| $g := \text{ITE}(g_1, g_2, g_3)$ | $(\bar{g} \vee \bar{g_1} \vee g_2)$, $(\bar{g} \vee g_1 \vee g_3)$ | $(g \vee \bar{g_1} \vee \bar{g_2})$, $(g \vee g_1 \vee \bar{g_3})$ |

## Definition (Structural Hashing of AND-gates)

Given a Boolean circuit with two equivalent gates, merge the gates.

## Example




$$x = \text{AND(a,b)} : (\bar{x} \vee a) \wedge (\bar{x} \vee b) \wedge (x \vee \bar{a} \vee \bar{b})$$
$$y = \text{AND(a,b)} : (\bar{y} \vee a) \wedge (\bar{y} \vee b) \wedge (y \vee \bar{a} \vee \bar{b})$$

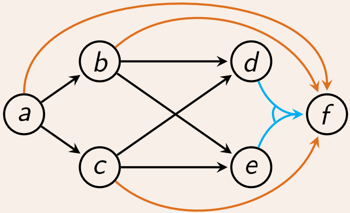
the two HBRs $(\bar{x} \vee y)$ and $(x \vee \bar{y})$ express that $x = y$

# Non-transitive Hyper Binary Resolution (NHBR)

A problem with classic HBR is that it adds many transitive binary clauses

## Example

Consider $(\bar{a} \vee b) \wedge (\bar{a} \vee c) \wedge (\bar{b} \vee d) \wedge (\bar{b} \vee e) \wedge (\bar{c} \vee d) \wedge (\bar{c} \vee e) \wedge (\bar{d} \vee \bar{e} \vee f)$.




adding $(\bar{b} \vee f)$ or $(\bar{c} \vee f)$ makes $(\bar{a} \vee f)$ transitive

## Solution [HeuleJärvisaloBiere 2013]

Add only non-transitive hyper binary resolvents

Can be implemented using an alternative unit propagation style

# Space Complexity of NHBR: Quadratic

## Question regarding complexity [Biere 2009]

- Are there formulas where the transitively reduced hyper binary resolution closure is quadratic in size w.r.t. to the size of the original?
- where size = #clauses or size = #literals or size = #variables

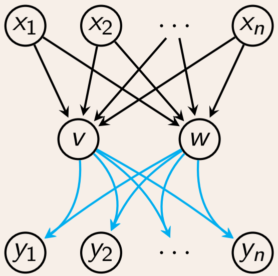
# Space Complexity of NHBR: Quadratic

## Question regarding complexity [Biere 2009]

- Are there formulas where the transitively reduced hyper binary resolution closure is quadratic in size w.r.t. to the size of the original?
- where size = #clauses or size = #literals or size = #variables

## Yes!

Consider the formula $F_n = \bigwedge_{1 \leq i \leq n} \left( (\bar{x}_i \vee v) \wedge (\bar{x}_i \vee w) \wedge (\bar{v} \vee \bar{w} \vee y_i) \right)$




#variables: $2n + 2$
#clauses: $3n$
#literals: $7n$

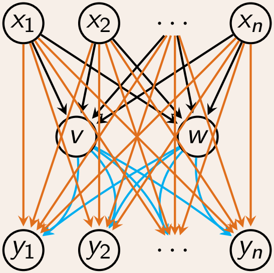
# Space Complexity of NHBR: Quadratic

## Question regarding complexity [Biere 2009]

- Are there formulas where the transitively reduced hyper binary resolution closure is quadratic in size w.r.t. to the size of the original?
- where size = #clauses or size = #literals or size = #variables

## Yes!

Consider the formula $F_n = \bigwedge_{1 \leq i \leq n} ((\bar{x}_i \vee v) \wedge (\bar{x}_i \vee w) \wedge (\bar{v} \vee \bar{w} \vee y_i))$




#variables: $2n + 2$
#clauses: $3n$
#literals: $7n$

$n^2$ hyper binary resolvents:
$(\bar{x}_i \vee y_j)$ for $1 \leq i, j \leq n$

# Unhiding Redundancy

# Redundancy

Redundant clauses:

- Removal of $C \in F$ preserves unsatisfiability of $F$
- Assign all $l \in C$ to false and check for a conflict in $F \setminus \{C\}$

# Redundancy

Redundant clauses:

- Removal of $C \in F$ preserves unsatisfiability of $F$
- Assign all $l \in C$ to false and check for a conflict in $F \setminus \{C\}$

Redundant literals:

- Removal of $l \in C$ preserves satisfiability of $F$
- Assign all $l' \in C \setminus \{l\}$ to false and check for a conflict in $F$

# Redundancy

## Redundant clauses:

- Removal of $C \in F$ preserves unsatisfiability of $F$
- Assign all $l \in C$ to false and check for a conflict in $F \setminus \{C\}$

## Redundant literals:

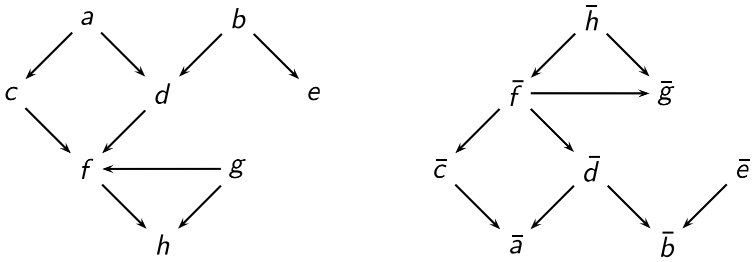
- Removal of $l \in C$ preserves satisfiability of $F$
- Assign all $l' \in C \setminus \{l\}$ to false and check for a conflict in $F$

## Redundancy elimination during pre- and in-processing

- Distillation [JinSomenzi2005]
- ReVivAl [PietteHamadiSaïs2008]
- Unhiding [HeuleJärvisaloBiere2011]

# Unhide: Binary implication graph (BIG)

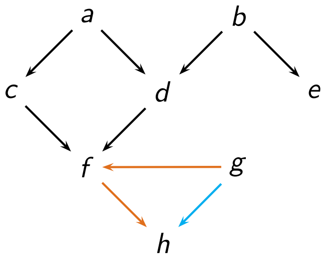
unhide: use the binary clauses to detect redundant clauses and literals

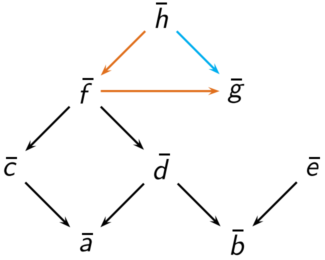



$(\bar{a} \vee c) \wedge (\bar{a} \vee d) \wedge (\bar{b} \vee d) \wedge (\bar{b} \vee e) \wedge$
$(\bar{c} \vee f) \wedge (\bar{d} \vee f) \wedge (\bar{g} \vee f) \wedge (\bar{f} \vee h) \wedge$
$(\bar{g} \vee h) \wedge \underbrace{(\bar{a} \vee \bar{e} \vee h) \wedge (\bar{b} \vee \bar{c} \vee h) \wedge (a \vee b \vee c \vee d \vee e \vee f \vee g \vee h)}_{\text{non binary clauses}}$

# Unhide: Transitive reduction (TRD)

transitive reduction: remove shortcuts in the binary implication graph




$(\bar{a} \vee c) \wedge (\bar{a} \vee d) \wedge (\bar{b} \vee d) \wedge (\bar{b} \vee e) \wedge$

$(\bar{c} \vee f) \wedge (\bar{d} \vee f) \wedge (\bar{g} \vee f) \wedge (\bar{f} \vee h) \wedge$

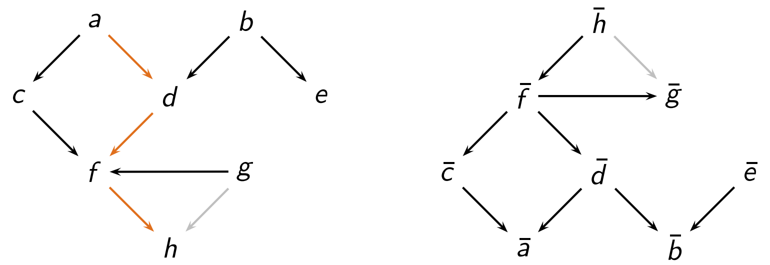
~~$(\bar{g} \vee h)$~~ $\wedge (\bar{a} \vee \bar{e} \vee h) \wedge (\bar{b} \vee \bar{c} \vee h) \wedge (a \vee b \vee c \vee d \vee e \vee f \vee g \vee h)$

TRD

$g \rightarrow f \rightarrow h$

# Unhide: Hidden tautology elimination (HTE) (1)

HTE removes clauses that are subsumed by an implication in BIG




$(\bar{a} \vee c) \wedge (\bar{a} \vee d) \wedge (\bar{b} \vee d) \wedge (\bar{b} \vee e) \wedge$

$(\bar{c} \vee f) \wedge (\bar{d} \vee f) \wedge (\bar{g} \vee f) \wedge (\bar{f} \vee h) \wedge$

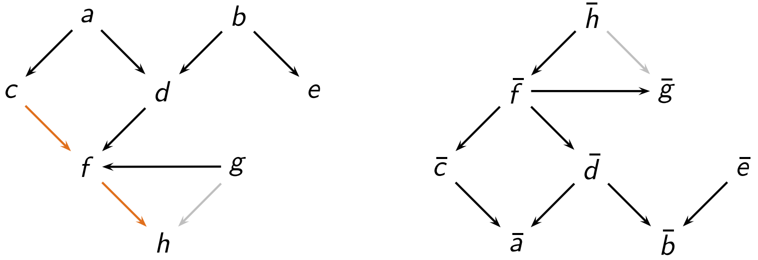
~~$(\bar{a} \vee \bar{e} \vee h)$~~ $\wedge (\bar{b} \vee \bar{c} \vee h) \wedge (a \vee b \vee c \vee d \vee e \vee f \vee g \vee h)$

HTE

$a \rightarrow d \rightarrow f \rightarrow h$

# Unhide: Hidden tautology elimination (HTE) (2)

HTE removes clauses that are subsumed by an implication in BIG



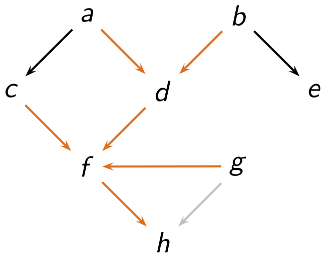

$$(\bar{a} \vee c) \wedge (\bar{a} \vee d) \wedge (\bar{b} \vee d) \wedge (\bar{b} \vee e) \wedge$$
$$(\bar{c} \vee f) \wedge (\bar{d} \vee f) \wedge (\bar{g} \vee f) \wedge (\bar{f} \vee h) \wedge$$
$$\sout{(\bar{b} \vee \bar{c} \vee h)} \wedge (a \vee b \vee c \vee d \vee e \vee f \vee g \vee h)$$

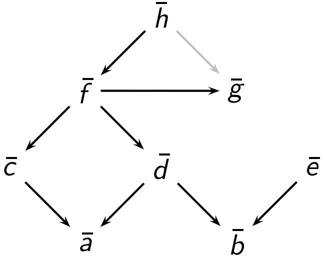
HTE

$c \rightarrow f \rightarrow h$

# Unhide: Hidden literal elimination (HLE)

HLE removes literal using the implication in BIG




$$(\bar{a} \vee c) \wedge (\bar{a} \vee d) \wedge (\bar{b} \vee d) \wedge (\bar{b} \vee e) \wedge$$
$$(\bar{c} \vee f) \wedge (\bar{d} \vee f) \wedge (\bar{g} \vee f) \wedge (\bar{f} \vee h) \wedge$$
$$(\sout{a \vee b \vee c \vee d} \vee e \vee \sout{f \vee g} \vee h)$$

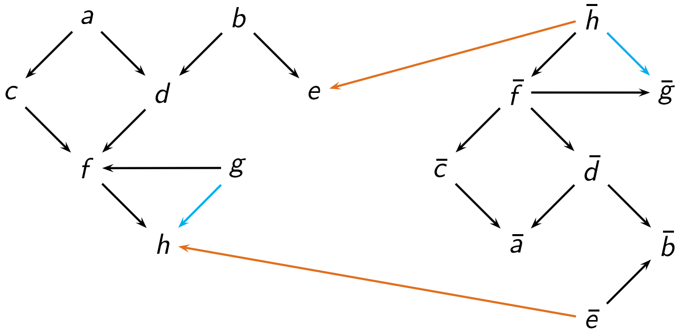
HLE

all but $e$ imply $h$

also $b$ implies $e$

# Unhide: TRD + HTE + HLE

unhide: redundancy elimination removes and adds arcs from BIG(F)




$$(\bar{a} \vee c) \wedge (\bar{a} \vee d) \wedge (\bar{b} \vee d) \wedge (\bar{b} \vee e) \wedge$$
$$(\bar{c} \vee f) \wedge (\bar{d} \vee f) \wedge (\bar{g} \vee f) \wedge (\bar{f} \vee h) \wedge (e \vee h)$$

# Conclusions

Many pre- or in-processing techniques in SAT solvers:

- (Self-)Subsumption
- Variable Elimination
- Blocked Clause Elimination
- Hyper Binary Resolution
- Bounded Variable Addition
- Equivalent Literal Substitution
- Failed Literal Elimination
- Autarky Reasoning
- ...

# Prepocessing

**Marijn J.H. Heule**

The University of Texas at Austin

July 5, 2013