Preprocessing Techniques

Marijn J.H. Heule

Carnegie Mellon University

http://www.cs.cmu.edu/~mheule/15816-f23/

Automated Reasoning and Satisfiability
September 18, 2023

Motivation

Subsumption

Variable Elimination

Bounded Variable Addition

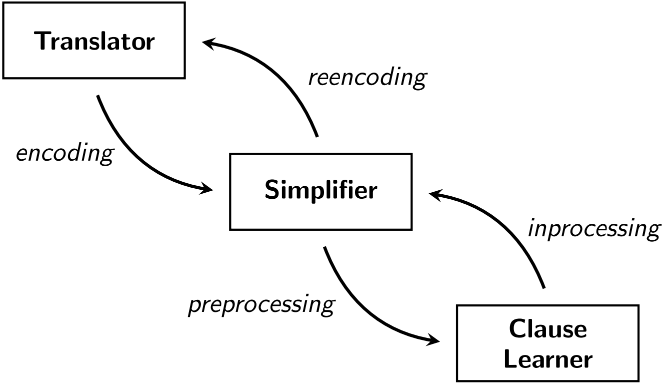
Blocked Clause Elimination

Hyper Binary Resolution

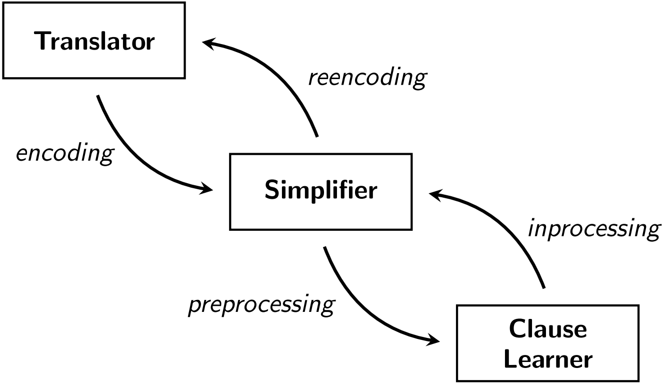
Unhiding Redundancy

Concluding Remarks

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 with respect to a formula if adding it to the formula preserves satisfiability.

- For unsatisfiable formulas, all clauses can be added, including the empty clause $\bot$.

Redundant clauses

A clause is redundant with respect to a formula if adding it to the formula preserves satisfiability.

- For unsatisfiable formulas, all clauses can be added, including the empty clause $\perp$.

A clause is redundant with respect to a formula if removing it from the formula preserves unsatisfiability.

- For satisfiable formulas, all clauses can be removed.

Redundant clauses

A clause is redundant with respect to a formula if adding it to the formula preserves satisfiability.

- For unsatisfiable formulas, all clauses can be added, including the empty clause $\perp$.

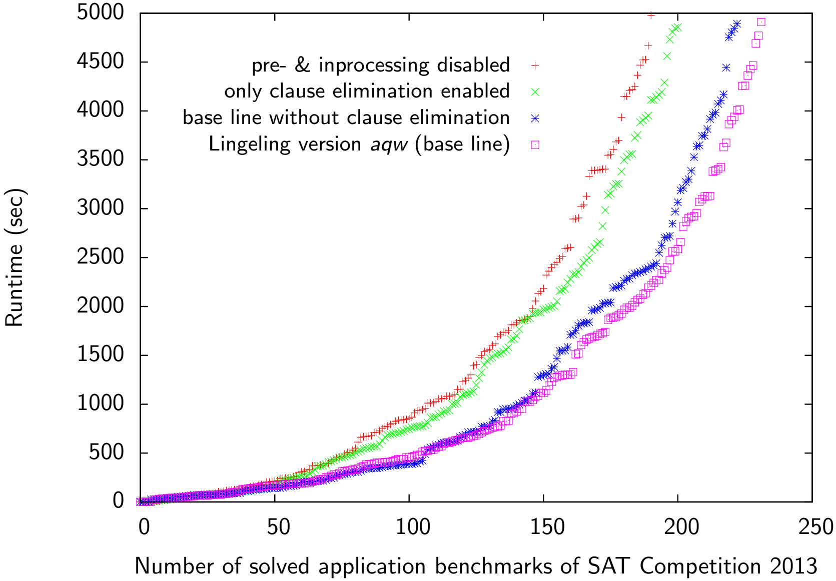
A clause is redundant with respect to a formula if removing it from the formula preserves unsatisfiability.

- 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

Preprocessing and Inprocessing in Practice



Tautologies and Subsumption

Definition (Tautology)

A clause C is a tautology if its contains two complementary literals x and $\overline{x}$.

Example

The clause $(a \lor b \lor \overline{b})$ *is a tautology.*

Definition (Subsumption)

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

Example

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

Self-Subsuming Resolution

Self-Subsuming Resolution

$$\frac{C \lor x \qquad D \lor \overline{x}}{D}\; C \subseteq D \qquad \frac{(a \lor b \lor x) \qquad (a \lor b \lor c \lor \overline{x})}{(a \lor b \lor c)}$$

resolvent D subsumes second antecedent $D \lor \overline{x}$

Self-Subsuming Resolution

Self-Subsuming Resolution

$$\frac{C \vee x \qquad D \vee \overline{x}}{D}\; C \subseteq D \qquad \frac{(a \vee b \vee x) \qquad (a \vee b \vee c \vee \overline{x})}{(a \vee b \vee c)}$$

resolvent D subsumes second antecedent $D \vee \overline{x}$

Example

Assume a CNF contains both antecedents
$\dots (a \vee b \vee x)(a \vee b \vee c \vee \overline{x}) \dots$

If D *is added, then* $D \vee \overline{x}$ *can be removed*

which in essence removes $\overline{x}$ *from* $D \vee \overline{x}$
$\dots (a \vee b \vee x)(a \vee b \vee c \quad) \dots$

Initially in the SATeLite preprocessor, [EenBiere'07]
now common in most solvers (i.e., as pre- and inprocessing)

Self-Subsuming Example

Self-Subsuming Resolution

$$\frac{C \vee x \qquad D \vee \overline{x}}{D}\; C \subseteq D \qquad \frac{(a \vee b \vee x) \qquad (a \vee b \vee c \vee \overline{x})}{(a \vee b \vee c)}$$

resolvent D subsumes second antecedent $D \vee \overline{x}$

Example: Remove literals using self-subsumption

$$\begin{array}{lll}
(a \vee b \vee c) & \wedge & (\overline{a} \vee b \vee c) \wedge \\
(\overline{a} \vee b \vee \overline{c}) & \wedge & (a \vee \overline{b} \vee c) \wedge \\
(\overline{a} \vee \overline{b} \vee d) & \wedge & (\overline{a} \vee \overline{b} \vee \overline{d}) \wedge \\
(a \vee \overline{c} \vee d) & \wedge & (a \vee \overline{c} \vee \overline{d})
\end{array}$$

Self-Subsuming Example

Self-Subsuming Resolution

$$\frac{C \vee x \qquad D \vee \overline{x}}{D} \; C \subseteq D \qquad \frac{(a \vee b \vee x) \qquad (a \vee b \vee c \vee \overline{x})}{(a \vee b \vee c)}$$

resolvent D subsumes second antecedent $D \vee \overline{x}$

Example: Remove literals using self-subsumption

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

Self-Subsuming Example

Self-Subsuming Resolution

$$\frac{C \lor x \qquad D \lor \overline{x}}{D} \; C \subseteq D \qquad \frac{(a \lor b \lor x) \qquad (a \lor b \lor c \lor \overline{x})}{(a \lor b \lor c)}$$

resolvent D subsumes second antecedent $D \lor \overline{x}$

Example: Remove literals using self-subsumption

$$\begin{array}{lcl}
(\phantom{\overline{a} \lor{}} b \lor c) & \land & (\overline{a} \lor b \lor c) \land \\
(\overline{a} \lor b \phantom{{} \lor c}) & \land & (a \lor \overline{b} \lor c) \land \\
(\overline{a} \lor \overline{b} \lor d) & \land & (\overline{a} \lor \overline{b} \lor \overline{d}) \land \\
(a \lor \overline{c} \lor d) & \land & (a \lor \overline{c} \lor \overline{d})
\end{array}$$

Self-Subsuming Example

Self-Subsuming Resolution

$$\frac{C \vee x \qquad D \vee \overline{x}}{D}\ C \subseteq D \qquad \frac{(a \vee b \vee x) \qquad (a \vee b \vee c \vee \overline{x})}{(a \vee b \vee c)}$$

resolvent D subsumes second antecedent $D \vee \overline{x}$

Example: Remove literals using self-subsumption

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

Self-Subsuming Example

Self-Subsuming Resolution

$$\frac{C \lor x \qquad D \lor \overline{x}}{D}\; C \subseteq D \qquad \frac{(a \lor b \lor x) \qquad (a \lor b \lor c \lor \overline{x})}{(a \lor b \lor c)}$$

resolvent D subsumes second antecedent $D \lor \overline{x}$

Example: Remove literals using self-subsumption

$$\begin{array}{lcl} (\phantom{\overline{a} \lor{}} b \lor c) & \land & (\overline{a} \lor b \lor c) \land \\ (\overline{a} \lor b \phantom{{}\lor c}) & \land & (a \lor \phantom{b \lor{}} c) \land \\ (\overline{a} \lor \overline{b} \phantom{{}\lor c}) & \land & (\overline{a} \lor \overline{b} \lor \overline{d}) \land \\ (a \lor \overline{c} \lor d) & \land & (a \lor \overline{c} \lor \overline{d}) \end{array}$$

Self-Subsuming Example

Self-Subsuming Resolution

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

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

Example: Remove literals using self-subsumption

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

Self-Subsuming Example

Self-Subsuming Resolution

$$\frac{C \lor x \qquad D \lor \overline{x}}{D}\ C \subseteq D \qquad \frac{(a \lor b \lor x) \qquad (a \lor b \lor c \lor \overline{x})}{(a \lor b \lor c)}$$

resolvent D subsumes second antecedent $D \lor \overline{x}$

Example: Remove literals using self-subsumption

$$\begin{array}{llllll}
(& b \lor c) & \land & (\overline{a} \lor b \lor c) & \land \\
(\overline{a} &) & \land & (a \lor c) & \land \\
(\overline{a} \lor \overline{b} &) & \land & (\overline{a} \lor \overline{b} \lor \overline{d}) & \land \\
(a \lor \overline{c} &) & \land & (a \lor \overline{c} \lor \overline{d}) &
\end{array}$$

Self-Subsuming Example

Self-Subsuming Resolution

$$\frac{C \lor x \qquad D \lor \overline{x}}{D}\; C \subseteq D \qquad \frac{(a \lor b \lor x) \qquad (a \lor b \lor c \lor \overline{x})}{(a \lor b \lor c)}$$

resolvent D subsumes second antecedent $D \lor \overline{x}$

Example: Remove literals using self-subsumption

$$\begin{array}{l}
(\quad b \lor c) \land (\overline{a} \lor b \lor c) \land \\
(\overline{a} \quad) \land (a \quad) \land \\
(\overline{a} \lor \overline{b} \quad) \land (\overline{a} \lor \overline{b} \lor \overline{d}) \land \\
(a \lor \overline{c} \quad) \land (a \lor \overline{c} \lor \overline{d})
\end{array}$$

Self-Subsuming Example

Self-Subsuming Resolution

$$\frac{C \vee x \qquad D \vee \overline{x}}{D} \; C \subseteq D \qquad \frac{(a \vee b \vee x) \qquad (a \vee b \vee c \vee \overline{x})}{(a \vee b \vee c)}$$

resolvent D subsumes second antecedent $D \vee \overline{x}$

Example: Remove literals using self-subsumption

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

Implementing Subsumption

Definition (Subsumption)

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

Example

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

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 Subsumption

Definition (Subsumption)

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

Example

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

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 Subsumption

Definition (Subsumption)

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

Example

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

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 x in C
  remove all clauses D in F_x that are subsumed by C

Variable Elimination [DavisPutnam'60]

Definition (Resolution)

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

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

Variable Elimination [DavisPutnam'60]

Definition (Resolution)

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

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

Definition (Variable elimination (VE))

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

Variable Elimination [DavisPutnam'60]

Definition (Resolution)

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

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

Definition (Variable elimination (VE))

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

Proof procedure [DavisPutnam60]

VE is a complete proof procedure. Applying VE until fixpoint results in either 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 $F_{\overline{x}}$ by $F_x \bowtie_x F_{\overline{x}}$

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 $F_{\overline{x}}$ by $F_x \bowtie_x F_{\overline{x}}$

Example of clause distribution

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

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 $F_{\overline{x}}$ by $F_x \bowtie_x F_{\overline{x}}$

Example of clause distribution

	$\overbrace{(x \lor c) \qquad (x \lor \overline{d}) \qquad (x \lor \overline{a} \lor \overline{b})}^{F_x}$		
	$(x \lor c)$	$(x \lor \overline{d})$	$(x \lor \overline{a} \lor \overline{b})$
$F_{\overline{x}}$: $(\overline{x} \lor a)$	$(a \lor c)$	$(a \lor \overline{d})$	$\cancel{(a \lor \overline{a} \lor \overline{b})}$
$(\overline{x} \lor b)$	$(b \lor c)$	$(b \lor \overline{d})$	$\cancel{(b \lor \overline{a} \lor \overline{b})}$
$(\overline{x} \lor \overline{e} \lor f)$	$(c \lor \overline{e} \lor f)$	$(\overline{d} \lor \overline{e} \lor f)$	$(\overline{a} \lor \overline{b} \lor \overline{e} \lor f)$

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 $F_{\overline{x}}$ by $F_x \bowtie_x F_{\overline{x}}$

Example of clause distribution

	F_x: $(x \vee c)$	F_x: $(x \vee \overline{d})$	F_x: $(x \vee \overline{a} \vee \overline{b})$
$F_{\overline{x}}$: $(\overline{x} \vee a)$	$(a \vee c)$	$(a \vee \overline{d})$	$\cancel{(a \vee \overline{a} \vee \overline{b})}$
$F_{\overline{x}}$: $(\overline{x} \vee b)$	$(b \vee c)$	$(b \vee \overline{d})$	$\cancel{(b \vee \overline{a} \vee \overline{b})}$
$F_{\overline{x}}$: $(\overline{x} \vee \overline{e} \vee f)$	$(c \vee \overline{e} \vee f)$	$(\overline{d} \vee \overline{e} \vee f)$	$(\overline{a} \vee \overline{b} \vee \overline{e} \vee f)$

In the example: $|F_x \bowtie F_{\overline{x}}| > |F_x| + |F_{\overline{x}}|$

Exponential growth of clauses in general

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

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

Variable elimination by substitution [EenBiere07]

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

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

VE by substitution [EenBiere'07]

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

$$F_x = (x \lor c) \land (x \lor \overline{d}) \land (x \lor \overline{a} \lor \overline{b})$$
$$F_{\overline{x}} = (\overline{x} \lor a) \land (\overline{x} \lor b) \land (\overline{x} \lor \overline{e} \lor f)$$

VE by substitution [EenBiere'07]

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

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

Example of substitution

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

VE by substitution [EenBiere'07]

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

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

Example of substitution

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

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

Bounded Variable Addition

Main Idea

Given a CNF formula F, can we construct a (semi)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 (semi)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

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

by

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

Bounded Variable Addition

Main Idea

Given a CNF formula F, can we construct a (semi)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

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

by

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

Challenge: how to find suitable patterns for replacement?

Factoring Out Subclauses

Example

Replace

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

by

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

adds 1 variable and 1 clause *reduces* number of literals by 2

Not compatible with VE, which would eliminate x immediately!

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

Example

Smallest pattern that is compatible: Replace

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

by

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

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 $\bigwedge_{i=1}^{n} (y \vee X_i) \wedge \bigwedge_{j=1}^{k} (\overline{y} \vee L_j)$

- 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

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

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

- 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 AtMostOne (1)

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

$$
\begin{array}{l}
(\overline{x}_1 \vee \overline{x}_2) \wedge (\overline{x}_9 \vee \overline{x}_{10}) \wedge (\overline{x}_8 \vee \overline{x}_{10}) \wedge (\overline{x}_7 \vee \overline{x}_{10}) \wedge (\overline{x}_6 \vee \overline{x}_{10}) \wedge \\
(\overline{x}_1 \vee \overline{x}_3) \wedge (\overline{x}_2 \vee \overline{x}_3) \wedge (\overline{x}_8 \vee \overline{x}_9) \wedge (\overline{x}_7 \vee \overline{x}_9) \wedge (\overline{x}_6 \vee \overline{x}_9) \wedge \\
(\overline{x}_1 \vee \overline{x}_4) \wedge (\overline{x}_2 \vee \overline{x}_4) \wedge (\overline{x}_3 \vee \overline{x}_4) \wedge (\overline{x}_7 \vee \overline{x}_8) \wedge (\overline{x}_6 \vee \overline{x}_8) \wedge \\
(\overline{x}_1 \vee \overline{x}_5) \wedge (\overline{x}_2 \vee \overline{x}_5) \wedge (\overline{x}_3 \vee \overline{x}_5) \wedge (\overline{x}_4 \vee \overline{x}_5) \wedge (\overline{x}_6 \vee \overline{x}_7) \wedge \\
(\overline{x}_1 \vee \overline{x}_6) \wedge (\overline{x}_2 \vee \overline{x}_6) \wedge (\overline{x}_3 \vee \overline{x}_6) \wedge (\overline{x}_4 \vee \overline{x}_6) \wedge (\overline{x}_5 \vee \overline{x}_6) \wedge \\
(\overline{x}_1 \vee \overline{x}_7) \wedge (\overline{x}_2 \vee \overline{x}_7) \wedge (\overline{x}_3 \vee \overline{x}_7) \wedge (\overline{x}_4 \vee \overline{x}_7) \wedge (\overline{x}_5 \vee \overline{x}_7) \wedge \\
(\overline{x}_1 \vee \overline{x}_8) \wedge (\overline{x}_2 \vee \overline{x}_8) \wedge (\overline{x}_3 \vee \overline{x}_8) \wedge (\overline{x}_4 \vee \overline{x}_8) \wedge (\overline{x}_5 \vee \overline{x}_8) \wedge \\
(\overline{x}_1 \vee \overline{x}_9) \wedge (\overline{x}_2 \vee \overline{x}_9) \wedge (\overline{x}_3 \vee \overline{x}_9) \wedge (\overline{x}_4 \vee \overline{x}_9) \wedge (\overline{x}_5 \vee \overline{x}_9) \wedge \\
(\overline{x}_1 \vee \overline{x}_{10}) \wedge (\overline{x}_2 \vee \overline{x}_{10}) \wedge (\overline{x}_3 \vee \overline{x}_{10}) \wedge (\overline{x}_4 \vee \overline{x}_{10}) \wedge (\overline{x}_5 \vee \overline{x}_{10})
\end{array}
$$

Bounded Variable Addition on AtMostOne (1)

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

$$
\begin{array}{l}
(\overline{x}_1 \vee \overline{x}_2) \wedge (\overline{x}_9 \vee \overline{x}_{10}) \wedge (\overline{x}_8 \vee \overline{x}_{10}) \wedge (\overline{x}_7 \vee \overline{x}_{10}) \wedge (\overline{x}_6 \vee \overline{x}_{10}) \wedge \\
(\overline{x}_1 \vee \overline{x}_3) \wedge (\overline{x}_2 \vee \overline{x}_3) \wedge (\overline{x}_8 \vee \overline{x}_9) \wedge (\overline{x}_7 \vee \overline{x}_9) \wedge (\overline{x}_6 \vee \overline{x}_9) \wedge \\
(\overline{x}_1 \vee \overline{x}_4) \wedge (\overline{x}_2 \vee \overline{x}_4) \wedge (\overline{x}_3 \vee \overline{x}_4) \wedge (\overline{x}_7 \vee \overline{x}_8) \wedge (\overline{x}_6 \vee \overline{x}_8) \wedge \\
(\overline{x}_1 \vee \overline{x}_5) \wedge (\overline{x}_2 \vee \overline{x}_5) \wedge (\overline{x}_3 \vee \overline{x}_5) \wedge (\overline{x}_4 \vee \overline{x}_5) \wedge (\overline{x}_6 \vee \overline{x}_7) \wedge \\
(\overline{x}_1 \vee \overline{x}_6) \wedge (\overline{x}_2 \vee \overline{x}_6) \wedge (\overline{x}_3 \vee \overline{x}_6) \wedge (\overline{x}_4 \vee \overline{x}_6) \wedge (\overline{x}_5 \vee \overline{x}_6) \wedge \\
(\overline{x}_1 \vee \overline{x}_7) \wedge (\overline{x}_2 \vee \overline{x}_7) \wedge (\overline{x}_3 \vee \overline{x}_7) \wedge (\overline{x}_4 \vee \overline{x}_7) \wedge (\overline{x}_5 \vee \overline{x}_7) \wedge \\
(\overline{x}_1 \vee \overline{x}_8) \wedge (\overline{x}_2 \vee \overline{x}_8) \wedge (\overline{x}_3 \vee \overline{x}_8) \wedge (\overline{x}_4 \vee \overline{x}_8) \wedge (\overline{x}_5 \vee \overline{x}_8) \wedge \\
(\overline{x}_1 \vee \overline{x}_9) \wedge (\overline{x}_2 \vee \overline{x}_9) \wedge (\overline{x}_3 \vee \overline{x}_9) \wedge (\overline{x}_4 \vee \overline{x}_9) \wedge (\overline{x}_5 \vee \overline{x}_9) \wedge \\
(\overline{x}_1 \vee \overline{x}_{10}) \wedge (\overline{x}_2 \vee \overline{x}_{10}) \wedge (\overline{x}_3 \vee \overline{x}_{10}) \wedge (\overline{x}_4 \vee \overline{x}_{10}) \wedge (\overline{x}_5 \vee \overline{x}_{10})
\end{array}
$$

Replace $(\overline{x}_i \vee \overline{x}_j)$ with $i \in \{1..5\}, j \in \{6..10\}$ by $(\overline{x}_i \vee y)$, $(\overline{x}_j \vee \overline{y})$

Bounded Variable Addition on AtMostOne (2)

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

$$
\begin{array}{l}
(\overline{x}_1 \lor \overline{x}_2) \land (\overline{x}_9 \lor \overline{x}_{10}) \land (\overline{x}_8 \lor \overline{x}_{10}) \land (\overline{x}_7 \lor \overline{x}_{10}) \land (\overline{x}_6 \lor \overline{x}_{10}) \land \\
(\overline{x}_1 \lor \overline{x}_3) \land (\overline{x}_2 \lor \overline{x}_3) \land (\overline{x}_8 \lor \overline{x}_9) \land (\overline{x}_7 \lor \overline{x}_9) \land (\overline{x}_6 \lor \overline{x}_9) \land \\
(\overline{x}_1 \lor \overline{x}_4) \land (\overline{x}_2 \lor \overline{x}_4) \land (\overline{x}_3 \lor \overline{x}_4) \land (\overline{x}_7 \lor \overline{x}_8) \land (\overline{x}_6 \lor \overline{x}_8) \land \\
(\overline{x}_1 \lor \overline{x}_5) \land (\overline{x}_2 \lor \overline{x}_5) \land (\overline{x}_3 \lor \overline{x}_5) \land (\overline{x}_4 \lor \overline{x}_5) \land (\overline{x}_6 \lor \overline{x}_7) \land \\
(\overline{x}_1 \lor y) \land (\overline{x}_2 \lor y) \land (\overline{x}_3 \lor y) \land (\overline{x}_4 \lor y) \land (\overline{x}_5 \lor y) \land \\
(\overline{x}_6 \lor \overline{y}) \land (\overline{x}_7 \lor \overline{y}) \land (\overline{x}_8 \lor \overline{y}) \land (\overline{x}_9 \lor \overline{y}) \land (\overline{x}_{10} \lor \overline{y})
\end{array}
$$

Bounded Variable Addition on AtMostOne (2)

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

$$(\overline{x}_1 \vee \overline{x}_2) \wedge (\overline{x}_9 \vee \overline{x}_{10}) \wedge (\overline{x}_8 \vee \overline{x}_{10}) \wedge (\overline{x}_7 \vee \overline{x}_{10}) \wedge (\overline{x}_6 \vee \overline{x}_{10}) \wedge$$
$$(\overline{x}_1 \vee \overline{x}_3) \wedge (\overline{x}_2 \vee \overline{x}_3) \wedge (\overline{x}_8 \vee \overline{x}_9) \wedge (\overline{x}_7 \vee \overline{x}_9) \wedge (\overline{x}_6 \vee \overline{x}_9) \wedge$$
$$(\overline{x}_1 \vee \overline{x}_4) \wedge (\overline{x}_2 \vee \overline{x}_4) \wedge (\overline{x}_3 \vee \overline{x}_4) \wedge (\overline{x}_7 \vee \overline{x}_8) \wedge (\overline{x}_6 \vee \overline{x}_8) \wedge$$
$$(\overline{x}_1 \vee \overline{x}_5) \wedge (\overline{x}_2 \vee \overline{x}_5) \wedge (\overline{x}_3 \vee \overline{x}_5) \wedge (\overline{x}_4 \vee \overline{x}_5) \wedge (\overline{x}_6 \vee \overline{x}_7) \wedge$$
$$(\overline{x}_1 \vee y) \wedge (\overline{x}_2 \vee y) \wedge (\overline{x}_3 \vee y) \wedge (\overline{x}_4 \vee y) \wedge (\overline{x}_5 \vee y) \wedge$$
$$(\overline{x}_6 \vee \overline{y}) \wedge (\overline{x}_7 \vee \overline{y}) \wedge (\overline{x}_8 \vee \overline{y}) \wedge (\overline{x}_9 \vee \overline{y}) \wedge (\overline{x}_{10} \vee \overline{y})$$

Replace matched pattern

$$(\overline{x}_1 \vee z) \wedge (\overline{x}_2 \vee z) \wedge (\overline{x}_3 \vee z) \wedge$$
$$(\overline{x}_4 \vee \overline{z}) \wedge (\overline{x}_5 \vee \overline{z}) \wedge (y \vee \overline{z})$$

Bounded Variable Addition on AtMostOne (3)

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

$$
\begin{array}{l}
(\overline{x}_1 \vee \overline{x}_2) \wedge (\overline{x}_9 \vee \overline{x}_{10}) \wedge (\overline{x}_8 \vee \overline{x}_{10}) \wedge (\overline{x}_7 \vee \overline{x}_{10}) \wedge (\overline{x}_6 \vee \overline{x}_{10}) \wedge \\
(\overline{x}_1 \vee \overline{x}_3) \wedge (\overline{x}_2 \vee \overline{x}_3) \wedge (\overline{x}_8 \vee \overline{x}_9) \wedge (\overline{x}_7 \vee \overline{x}_9) \wedge (\overline{x}_6 \vee \overline{x}_9) \wedge \\
(\overline{x}_1 \vee z) \wedge (\overline{x}_2 \vee z) \wedge (\overline{x}_3 \vee z) \wedge (\overline{x}_7 \vee \overline{x}_8) \wedge (\overline{x}_6 \vee \overline{x}_8) \wedge \\
(\overline{x}_4 \vee \overline{z}) \wedge (\overline{x}_5 \vee \overline{z}) \wedge (y \vee \overline{z}) \wedge (\overline{x}_4 \vee \overline{x}_5) \wedge (\overline{x}_6 \vee \overline{x}_7) \wedge \\
(\overline{x}_4 \vee y) \wedge (\overline{x}_5 \vee y) \wedge (\overline{x}_6 \vee \overline{y}) \wedge (\overline{x}_7 \vee \overline{y}) \wedge (\overline{x}_8 \vee \overline{y}) \\
(\overline{x}_9 \vee \overline{y}) \wedge (\overline{x}_{10} \vee \overline{y})
\end{array}
$$

Bounded Variable Addition on AtMostOne (3)

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

$$
\begin{array}{l}
(\overline{x}_1 \vee \overline{x}_2) \wedge (\overline{x}_9 \vee \overline{x}_{10}) \wedge (\overline{x}_8 \vee \overline{x}_{10}) \wedge (\overline{x}_7 \vee \overline{x}_{10}) \wedge (\overline{x}_6 \vee \overline{x}_{10}) \wedge \\
(\overline{x}_1 \vee \overline{x}_3) \wedge (\overline{x}_2 \vee \overline{x}_3) \wedge (\overline{x}_8 \vee \overline{x}_9) \wedge (\overline{x}_7 \vee \overline{x}_9) \wedge (\overline{x}_6 \vee \overline{x}_9) \wedge \\
(\overline{x}_1 \vee z) \wedge (\overline{x}_2 \vee z) \wedge (\overline{x}_3 \vee z) \wedge (\overline{x}_7 \vee \overline{x}_8) \wedge (\overline{x}_6 \vee \overline{x}_8) \wedge \\
(\overline{x}_4 \vee \overline{z}) \wedge (\overline{x}_5 \vee \overline{z}) \wedge (y \vee \overline{z}) \wedge (\overline{x}_4 \vee \overline{x}_5) \wedge (\overline{x}_6 \vee \overline{x}_7) \wedge \\
(\overline{x}_4 \vee y) \wedge (\overline{x}_5 \vee y) \wedge (\overline{x}_6 \vee \overline{y}) \wedge (\overline{x}_7 \vee \overline{y}) \wedge (\overline{x}_8 \vee \overline{y}) \\
(\overline{x}_9 \vee \overline{y}) \wedge (\overline{x}_{10} \vee \overline{y})
\end{array}
$$

Replace matched pattern

$$
\begin{array}{l}
(\overline{x}_6 \vee w) \wedge (\overline{x}_7 \vee w) \wedge (\overline{x}_8 \vee w) \wedge \\
(\overline{x}_9 \vee \overline{w}) \wedge (\overline{x}_{10} \vee \overline{w}) \wedge (\overline{y} \vee \overline{w})
\end{array}
$$

Blocked Clauses [Kullmann 1999]

Definition (Block Clause)

A clause $(C \vee x)$ is a blocked on x w.r.t. a CNF formula F if for every clause $(D \vee \overline{x}) \in F$, resolvent $C \vee D$ is a tautology.

Blocked Clauses [Kullmann 1999]

Definition (Block Clause)

A clause $(C \vee x)$ is a blocked on x w.r.t. a CNF formula F if for every clause $(D \vee \overline{x}) \in F$, resolvent $C \vee D$ is a tautology.

Example

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

Blocked Clauses [Kullmann 1999]

Definition (Block Clause)

A clause $(C \vee x)$ is a blocked on x w.r.t. a CNF formula F if for every clause $(D \vee \overline{x}) \in F$, resolvent $C \vee D$ is a tautology.

Example

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

Theorem

Adding or removing a blocked clause preserves (un)satisfiability.

Blocked Clause Elimination (BCE)

Definition (BCE)

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

Example

Consider $(a \vee b) \wedge (a \vee \overline{b} \vee \overline{c}) \wedge (\overline{a} \vee c)$.
After removing either $(a \vee \overline{b} \vee \overline{c})$ *or* $(\overline{a} \vee c)$, *the clause* $(a \vee b)$ *becomes blocked (no clause with either* $\overline{b}$ *or* $\overline{a}$*).*

An extreme case in which BCE *removes all clauses!*

Blocked Clause Elimination (BCE)

Definition (BCE)

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

Example

Consider $(a \vee b) \wedge (a \vee \overline{b} \vee \overline{c}) \wedge (\overline{a} \vee c)$.
After removing either $(a \vee \overline{b} \vee \overline{c})$ *or* $(\overline{a} \vee c)$, *the clause* $(a \vee b)$ *becomes blocked (no clause with either* $\overline{b}$ *or* $\overline{a}$*).*

An extreme case in which BCE *removes all clauses!*

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, etc.

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, etc.

Example of circuit simplification by BCE on Tseitin encoding

(y)

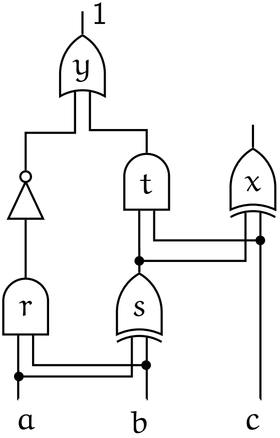
$(\overline{y} \vee t \vee \overline{r})$
$(y \vee \overline{t})$
$(y \vee r)$

$(\overline{x} \vee s \vee c)$
$(\overline{x} \vee \overline{s} \vee \overline{c})$
$(x \vee s \vee \overline{c})$
$(x \vee \overline{s} \vee c)$

$(t \vee \overline{s} \vee \overline{c})$
$(\overline{t} \vee s)$
$(\overline{t} \vee c)$

$(r \vee \overline{a} \vee \overline{b})$
$(\overline{r} \vee a)$
$(\overline{r} \vee b)$

$(\overline{s} \vee a \vee b)$
$(\overline{s} \vee \overline{a} \vee \overline{b})$
$(s \vee a \vee \overline{b})$
$(s \vee \overline{a} \vee b)$



BCE very effective on circuits [JärvisaloBiereHeule'10]

BCE converts the Tseitin encoding to Plaisted Greenbaum

BCE simulates Pure literal elimination, Cone of influence, etc.

Example of circuit simplification by BCE on Tseitin encoding

(y)

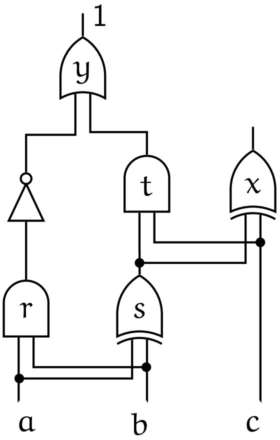
$(\bar{y} \lor t \lor \bar{r})$
$\cancel{(y \lor \bar{t})}$
$\cancel{(y \lor r)}$

$(\bar{x} \lor s \lor c)$
$(\bar{x} \lor \bar{s} \lor \bar{c})$
$(x \lor s \lor \bar{c})$
$(x \lor \bar{s} \lor c)$

$(t \lor \bar{s} \lor \bar{c})$
$(\bar{t} \lor s)$
$(\bar{t} \lor c)$

$(r \lor \bar{a} \lor \bar{b})$
$(\bar{r} \lor a)$
$(\bar{r} \lor b)$

$(\bar{s} \lor a \lor b)$
$(\bar{s} \lor \bar{a} \lor \bar{b})$
$(s \lor a \lor \bar{b})$
$(s \lor \bar{a} \lor b)$



BCE very effective on circuits [JärvisaloBiereHeule'10]

BCE converts the Tseitin encoding to Plaisted Greenbaum

BCE simulates Pure literal elimination, Cone of influence, etc.

Example of circuit simplification by BCE on Tseitin encoding

(y)

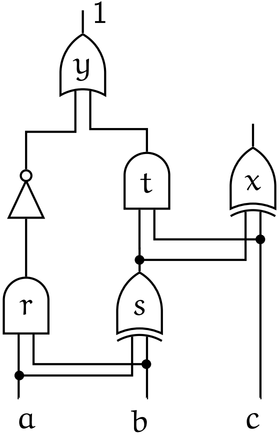
$(\bar{y} \lor t \lor \bar{r})$
$\cancel{(y \lor \bar{t})}$
$\cancel{(y \lor r)}$

$\cancel{(\bar{x} \lor s \lor c)}$
$\cancel{(\bar{x} \lor \bar{s} \lor \bar{c})}$
$\cancel{(x \lor s \lor \bar{c})}$
$\cancel{(x \lor \bar{s} \lor c)}$

$(t \lor \bar{s} \lor \bar{c})$
$(\bar{t} \lor s)$
$(\bar{t} \lor c)$

$(r \lor \bar{a} \lor \bar{b})$
$(\bar{r} \lor a)$
$(\bar{r} \lor b)$

$(\bar{s} \lor a \lor b)$
$(\bar{s} \lor \bar{a} \lor \bar{b})$
$(s \lor a \lor \bar{b})$
$(s \lor \bar{a} \lor b)$



BCE very effective on circuits [JärvisaloBiereHeule'10]

BCE converts the Tseitin encoding to Plaisted Greenbaum
BCE simulates Pure literal elimination, Cone of influence, etc.

Example of circuit simplification by BCE on Tseitin encoding

(y)

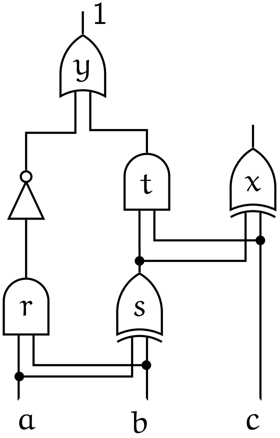
$(\bar{y} \vee t \vee \bar{r})$
$\cancel{(y \vee \bar{t})}$
$\cancel{(y \vee r)}$

$\cancel{(\bar{x} \vee s \vee c)}$
$\cancel{(\bar{x} \vee \bar{s} \vee \bar{c})}$
$\cancel{(x \vee s \vee \bar{c})}$
$\cancel{(x \vee \bar{s} \vee c)}$

$\cancel{(t \vee \bar{s} \vee \bar{c})}$
$(\bar{t} \vee s)$
$(\bar{t} \vee c)$

$(r \vee \bar{a} \vee \bar{b})$
$(\bar{r} \vee a)$
$(\bar{r} \vee b)$

$(\bar{s} \vee a \vee b)$
$(\bar{s} \vee \bar{a} \vee \bar{b})$
$(s \vee a \vee \bar{b})$
$(s \vee \bar{a} \vee b)$



BCE very effective on circuits [JärvisaloBiereHeule'10]

BCE converts the Tseitin encoding to Plaisted Greenbaum

BCE simulates Pure literal elimination, Cone of influence, etc.

Example of circuit simplification by BCE on Tseitin encoding

(y)

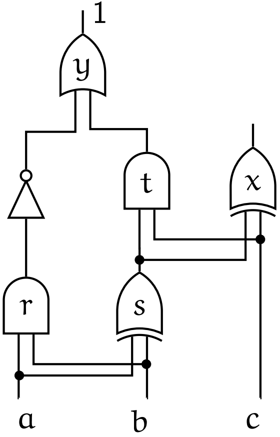
$(\bar{y} \lor t \lor \bar{r})$
$\cancel{(y \lor \bar{t})}$
$\cancel{(y \lor r)}$

$\cancel{(\bar{x} \lor s \lor c)}$
$\cancel{(\bar{x} \lor \bar{s} \lor \bar{c})}$
$\cancel{(x \lor s \lor \bar{c})}$
$\cancel{(x \lor \bar{s} \lor c)}$

$\cancel{(t \lor \bar{s} \lor \bar{c})}$
$(\bar{t} \lor s)$
$(\bar{t} \lor c)$

$(r \lor \bar{a} \lor \bar{b})$
$\cancel{(\bar{r} \lor a)}$
$\cancel{(\bar{r} \lor b)}$

$(\bar{s} \lor a \lor b)$
$(\bar{s} \lor \bar{a} \lor \bar{b})$
$(s \lor a \lor \bar{b})$
$(s \lor \bar{a} \lor b)$



BCE very effective on circuits [JärvisaloBiereHeule'10]

- BCE converts the Tseitin encoding to Plaisted Greenbaum
- BCE simulates Pure literal elimination, Cone of influence, etc.

Example of circuit simplification by BCE on Tseitin encoding

(y)

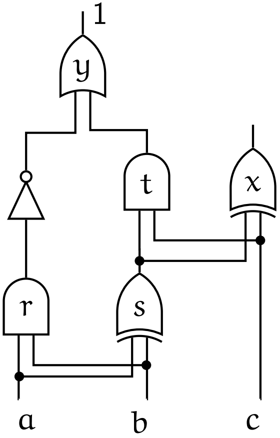
$(\bar{y} \lor t \lor \bar{r})$
$\cancel{(y \lor \bar{t})}$
$\cancel{(y \lor r)}$

$\cancel{(\bar{x} \lor s \lor c)}$
$\cancel{(\bar{x} \lor \bar{s} \lor \bar{c})}$
$\cancel{(x \lor s \lor \bar{c})}$
$\cancel{(x \lor \bar{s} \lor c)}$

$\cancel{(t \lor \bar{s} \lor \bar{c})}$
$(\bar{t} \lor s)$
$(\bar{t} \lor c)$

$(r \lor \bar{a} \lor \bar{b})$
$\cancel{(\bar{r} \lor a)}$
$\cancel{(\bar{r} \lor b)}$

$(\bar{s} \lor a \lor b)$
$(\bar{s} \lor \bar{a} \lor \bar{b})$
$\cancel{(s \lor a \lor \bar{b})}$
$\cancel{(s \lor \bar{a} \lor b)}$



BCE very effective on circuits [JärvisaloBiereHeule'10]

BCE converts the Tseitin encoding to Plaisted Greenbaum

BCE simulates Pure literal elimination, Cone of influence, etc.

Example of circuit simplification by BCE on Tseitin encoding

(y)

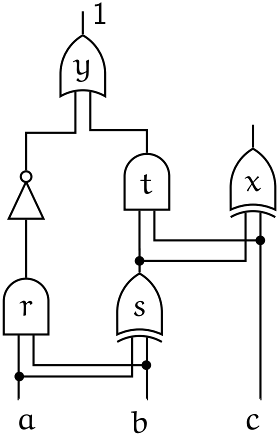
$(\bar{y} \lor t \lor \bar{r})$
$\cancel{(y \lor \bar{t})}$
$\cancel{(y \lor r)}$

$\cancel{(\bar{x} \lor s \lor c)}$
$\cancel{(\bar{x} \lor \bar{s} \lor \bar{c})}$
$\cancel{(x \lor s \lor \bar{c})}$
$\cancel{(x \lor \bar{s} \lor c)}$

$\cancel{(t \lor \bar{s} \lor \bar{c})}$
$(\bar{t} \lor s)$
$(\bar{t} \lor c)$

$\cancel{(r \lor \bar{a} \lor \bar{b})}$
$\cancel{(\bar{r} \lor a)}$
$\cancel{(\bar{r} \lor b)}$

$(\bar{s} \lor a \lor b)$
$(\bar{s} \lor \bar{a} \lor \bar{b})$
$\cancel{(s \lor a \lor \bar{b})}$
$\cancel{(s \lor \bar{a} \lor b)}$



BCE very effective on circuits [JärvisaloBiereHeule'10]

BCE converts the Tseitin encoding to Plaisted Greenbaum
BCE simulates Pure literal elimination, Cone of influence, etc.

Example of circuit simplification by BCE on Tseitin encoding

(y)

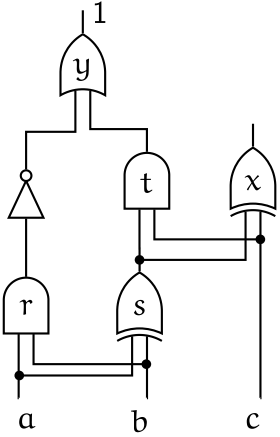
$\cancel{(\bar{y} \lor t \lor \bar{r})}$
$\cancel{(y \lor \bar{t})}$
$\cancel{(y \lor r)}$

$\cancel{(\bar{x} \lor s \lor c)}$
$\cancel{(\bar{x} \lor \bar{s} \lor \bar{c})}$
$\cancel{(x \lor s \lor \bar{c})}$
$\cancel{(x \lor \bar{s} \lor c)}$

$\cancel{(t \lor \bar{s} \lor \bar{c})}$
$(\bar{t} \lor s)$
$(\bar{t} \lor c)$

$\cancel{(r \lor \bar{a} \lor \bar{b})}$
$\cancel{(\bar{r} \lor a)}$
$\cancel{(\bar{r} \lor b)}$

$(\bar{s} \lor a \lor b)$
$(\bar{s} \lor \bar{a} \lor \bar{b})$
$\cancel{(s \lor a \lor \bar{b})}$
$\cancel{(s \lor \bar{a} \lor b)}$



BCE very effective on circuits [JärvisaloBiereHeule'10]

BCE converts the Tseitin encoding to Plaisted Greenbaum
BCE simulates Pure literal elimination, Cone of influence, etc.

Example of circuit simplification by BCE on Tseitin encoding

$\cancel{(y)}$

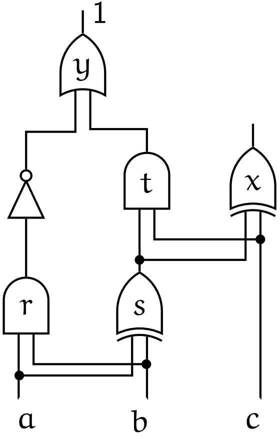
$\cancel{(\bar{y} \lor t \lor \bar{r})}$
$\cancel{(y \lor \bar{t})}$
$\cancel{(y \lor r)}$

$\cancel{(\bar{x} \lor s \lor c)}$
$\cancel{(\bar{x} \lor \bar{s} \lor \bar{c})}$
$\cancel{(x \lor s \lor \bar{c})}$
$\cancel{(x \lor \bar{s} \lor c)}$

$\cancel{(t \lor \bar{s} \lor \bar{c})}$
$(\bar{t} \lor s)$
$(\bar{t} \lor c)$

$\cancel{(r \lor \bar{a} \lor \bar{b})}$
$\cancel{(\bar{r} \lor a)}$
$\cancel{(\bar{r} \lor b)}$

$(\bar{s} \lor a \lor b)$
$(\bar{s} \lor \bar{a} \lor \bar{b})$
$\cancel{(s \lor a \lor \bar{b})}$
$\cancel{(s \lor \bar{a} \lor b)}$



BCE very effective on circuits [JärvisaloBiereHeule'10]

BCE converts the Tseitin encoding to Plaisted Greenbaum

BCE simulates Pure literal elimination, Cone of influence, etc.

Example of circuit simplification by BCE on Tseitin encoding

$\cancel{(y)}$

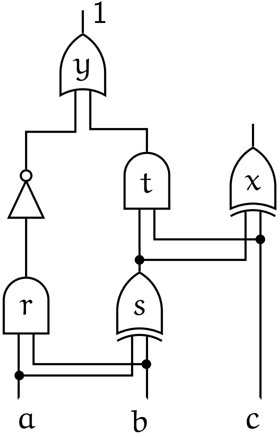
$\cancel{(\bar{y} \vee t \vee \bar{r})}$
$\cancel{(y \vee \bar{t})}$
$\cancel{(y \vee r)}$

$\cancel{(\bar{x} \vee s \vee c)}$
$\cancel{(\bar{x} \vee \bar{s} \vee \bar{c})}$
$\cancel{(x \vee s \vee \bar{c})}$
$\cancel{(x \vee \bar{s} \vee c)}$

$\cancel{(t \vee \bar{s} \vee \bar{c})}$
$\cancel{(\bar{t} \vee s)}$
$\cancel{(\bar{t} \vee c)}$

$\cancel{(r \vee \bar{a} \vee \bar{b})}$
$\cancel{(\bar{r} \vee a)}$
$\cancel{(\bar{r} \vee b)}$

$(\bar{s} \vee a \vee b)$
$(\bar{s} \vee \bar{a} \vee \bar{b})$
$\cancel{(s \vee a \vee \bar{b})}$
$\cancel{(s \vee \bar{a} \vee b)}$



BCE very effective on circuits [JärvisaloBiereHeule'10]

BCE converts the Tseitin encoding to Plaisted Greenbaum
BCE simulates Pure literal elimination, Cone of influence, etc.

Example of circuit simplification by BCE on Tseitin encoding

$\cancel{(y)}$

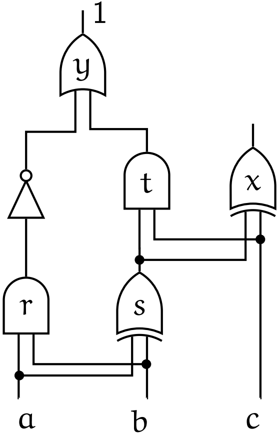
$\cancel{(\bar{y} \vee t \vee \bar{r})}$
$\cancel{(y \vee \bar{t})}$
$\cancel{(y \vee r)}$

$\cancel{(\bar{x} \vee s \vee c)}$
$\cancel{(\bar{x} \vee \bar{s} \vee \bar{c})}$
$\cancel{(x \vee s \vee \bar{c})}$
$\cancel{(x \vee \bar{s} \vee c)}$

$\cancel{(t \vee \bar{s} \vee \bar{c})}$
$\cancel{(\bar{t} \vee s)}$
$\cancel{(\bar{t} \vee c)}$

$\cancel{(r \vee \bar{a} \vee \bar{b})}$
$\cancel{(\bar{r} \vee a)}$
$\cancel{(\bar{r} \vee b)}$

$\cancel{(\bar{s} \vee a \vee b)}$
$\cancel{(\bar{s} \vee \bar{a} \vee \bar{b})}$
$\cancel{(s \vee a \vee \bar{b})}$
$\cancel{(s \vee \bar{a} \vee b)}$



BCE: Solution Reconstruction

Input:

- stack S of eliminated blocked clauses
- formula F (without the blocked clauses)
- assignment α that satisfies F

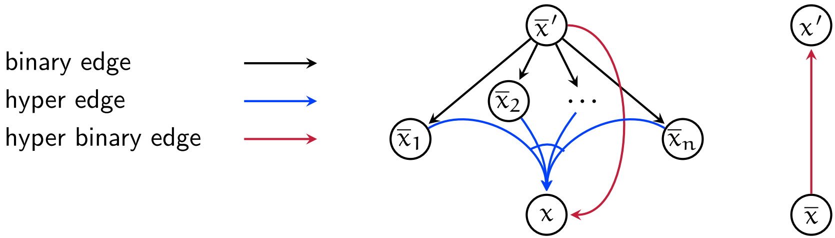
Output: an assignment that satisfies $F \wedge S$

while S.size () **do**
  $\langle C, l \rangle :=$ S.pop ()
  if α falsifies C **then** $\alpha := \alpha_l$
end while
return α

Hyper Binary Resolution [Bacchus-AAAI02]

Definition (Hyper Binary Resolution Rule)

$$\frac{(x \lor x_1 \lor x_2 \lor \cdots \lor x_n) \quad (\bar{x}_1 \lor x') \quad (\bar{x}_2 \lor x') \quad \ldots \quad (\bar{x}_n \lor x')}{(x \lor x')}$$



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 $x = x'$

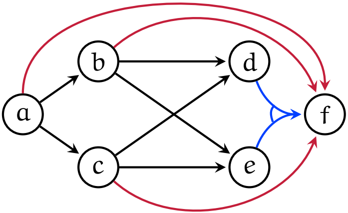
Hyper Binary Resolution (HBR)

Definition (Hyper Binary Resolution)

Apply the hyper binary resolution rule until fixpoint

Example

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



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

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

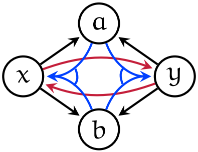
Structural Hashing of AND-gates via HBR

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

Definition (Structural Hashing of AND-gates)

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

Example



x = AND(a,b) : $(\overline{x} \vee a) \wedge (\overline{x} \vee b) \wedge (x \vee \overline{a} \vee \overline{b})$

y = AND(a,b) : $(\overline{y} \vee a) \wedge (\overline{y} \vee b) \wedge (y \vee \overline{a} \vee \overline{b})$

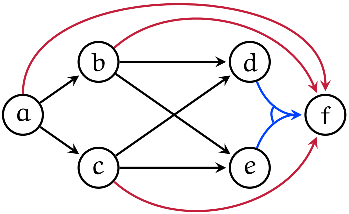
the two HBRs $(\overline{x} \vee y)$ and $(x \vee \overline{y})$ express that $x = y$

Non-transitive Hyper Binary Resolution (NHBR)

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

Example

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



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

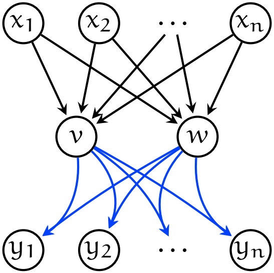
Space Complexity of NHBR: Quadratic

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



#variables: $2n + 2$

#clauses: $3n$

#literals: $7n$

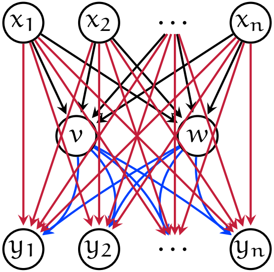
Space Complexity of NHBR: Quadratic

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



#variables: $2n + 2$

#clauses: $3n$

#literals: $7n$

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

Redundancy

Redundant clauses:

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

Redundant literals:

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

Redundancy

Redundant clauses:

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

Redundant literals:

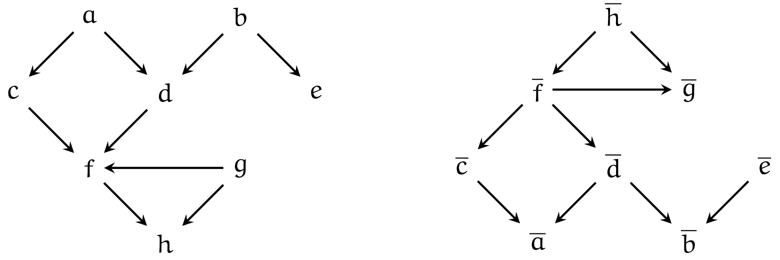
- Removal of $x \in C$ preserves satisfiability of F
- Assign all $x' \in C \setminus \{x\}$ to false and check for a conflict in F

Redundancy elimination during pre- and in-processing

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

Unhide: Binary implication graph (BIG)

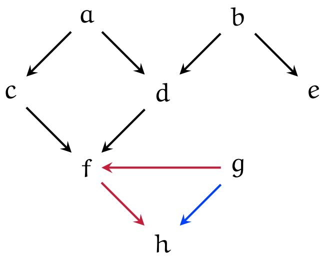
unhide: use the binary clauses to detect redundant clauses and literals

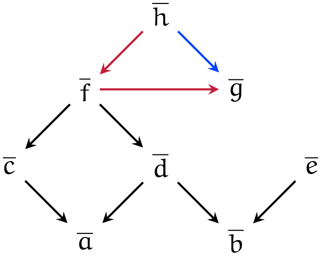


$(\overline{a} \vee c) \wedge (\overline{a} \vee d) \wedge (\overline{b} \vee d) \wedge (\overline{b} \vee e) \wedge$
$(\overline{c} \vee f) \wedge (\overline{d} \vee f) \wedge (\overline{g} \vee f) \wedge (\overline{f} \vee h) \wedge$
$(\overline{g} \vee h) \wedge \underbrace{(\overline{a} \vee \overline{e}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

Unhide: Transitive reduction (TRD)

transitive reduction: remove shortcuts in the binary implication graph



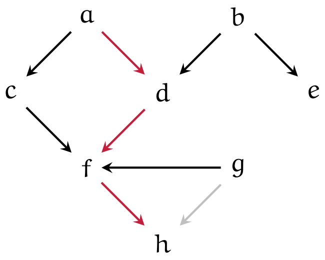


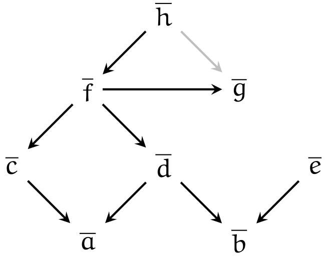
$$(\bar{a} \vee c) \wedge (\bar{a} \vee d) \wedge (\bar{b} \vee d) \wedge (\bar{b} \vee e) \wedge$$
$$(\bar{c} \vee f) \wedge (\bar{d} \vee f) \wedge (\bar{g} \vee f) \wedge (\bar{f} \vee h) \wedge$$
$$\underbrace{\cancel{(\bar{g} \vee h)}}_{\text{TRD}} \wedge (\bar{a} \vee \bar{e} \vee h) \wedge (\bar{b}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

$g \rightarrow f \rightarrow h$

Unhide: Hidden tautology elimination (HTE) (1)

HTE removes clauses that are subsumed by an implication in BIG

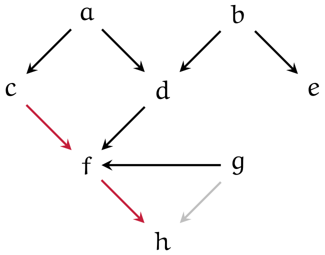


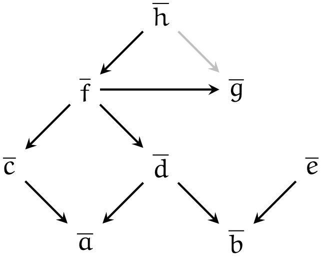


$$
\begin{aligned}
&(\bar{a} \lor c) \land (\bar{a} \lor d) \land (\bar{b} \lor d) \land (\bar{b} \lor e) \land \\
&(\bar{c} \lor f) \land (\bar{d} \lor f) \land (\bar{g} \lor f) \land (\bar{f} \lor h) \land \\
&\qquad \underbrace{\cancel{(\bar{a} \lor \bar{e} \lor h)}}_{\substack{\text{HTE} \\ a \rightarrow d \rightarrow f \rightarrow h}} \land (\bar{b} \lor \bar{c} \lor h) \land (a \lor b \lor c \lor d \lor e \lor f \lor g \lor h)
\end{aligned}
$$

Unhide: Hidden tautology elimination (HTE) (2)

HTE removes clauses that are subsumed by an implication in BIG

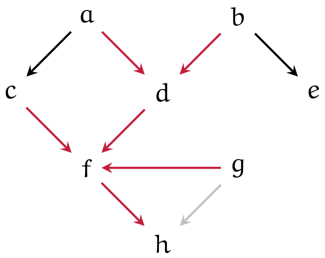


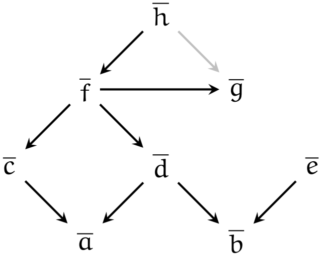


$$(\bar{a} \vee c) \wedge (\bar{a} \vee d) \wedge (\bar{b} \vee d) \wedge (\bar{b} \vee e) \wedge$$
$$(\bar{c} \vee f) \wedge (\bar{d} \vee f) \wedge (\bar{g} \vee f) \wedge (\bar{f} \vee h) \wedge$$
$$\underbrace{\cancel{(\bar{b} \vee \bar{c} \vee h)}}_{\substack{\text{HTE} \\ c \rightarrow f \rightarrow h}} \wedge (a \vee b \vee c \vee d \vee e \vee f \vee g \vee h)$$

Unhide: Hidden literal elimination (HLE)

HLE removes literal using the implication in BIG





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

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

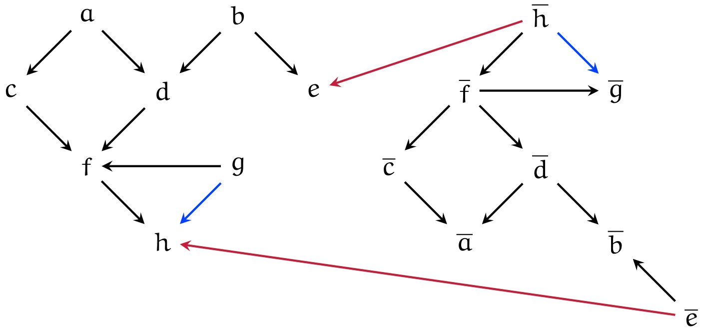
HLE

all but e imply h

also b implies e

Unhide: TRD + HTE + HLE

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



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

Many Techniques

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
- ...

Many Techniques

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
- ...

... and the list is growing:

- Propagation Redundant Clauses [CADE'17]