Blocked Clause Elimination for Projected Model Counting

Jean-Marie Lagniez[1] Pierre Marquis[1] Armin Biere[2]
lagniez@cril.fr marquis@cril.fr biere@cs.uni-freiburg.de

[1] Univ. Artois, CNRS, CRIL, France [2] University Freiburg, Germany

SAT 2024
August 20, Pune, India
https://cca.informatik.uni-freiburg.de/biere/talks/Biere-SAT24-blocked-talk.pdf

Motivations

- Projected model counting (PMC) is a significant problem in many areas
  - ⇒ AI, formal verification, database, ...
- generalization of the standard model counting problem #SAT
- PMC is at least as complex as #SAT (#P-hard)
- In practice PMC turns out to be more challenging than #SAT
  - ⇒ additional constraints imposed on the branching heuristic
- Idea: use preprocessing techniques to enhance resolution of #SAT / PMC

Blocked-clause elimination appears as a valuable candidate, but ...

The SAT problem

$1: x_1 \vee x_2$
$2: \neg x_2 \vee x_3$
$3: \neg x_1 \vee \neg x_2 \vee \neg y_1$
$4: x_1 \vee \neg x_3 \vee y_1$
$5: x_2 \vee \neg x_3 \vee y_2$
$6: x_1 \vee \neg x_3 \vee \neg y_2$
$7: y_3 \vee x_2$
$8: \neg y_3 \vee \neg x_2 \vee \neg x_3$
$9: \neg y_3 \vee x_1$
$10: \neg y_3 \vee \neg y_2 \vee x_3$
$11: y_3 \vee y_2 \vee x_2$

- Propositional variables: a, b, c
- Literals: $a, \neg a$
- Clauses: $\alpha_i = a \vee \neg b$ (also viewed as sets of literals)
- CNF formula: $\mathbf{\Sigma}$
  (also viewed as an ordered set of clauses)
- SAT: can we find an interpretation ω of the variables that satisfies the input formula?
  - $\Rightarrow$ ω is called a model of $\mathbf{\Sigma}$
- $S_\ell(\mathbf{\Sigma}) = \{\alpha_i \in \mathbf{\Sigma} \mid \alpha_i \cap \ell \neq \emptyset\}$
- Let $\mathbf{\Sigma} = \{\alpha_1, \ldots, \alpha_m\}$ then
  $\mathbf{\Sigma}[i]$ denotes the clause α_i
- $\oplus$ denotes resolution
  $\alpha = \alpha_1 \oplus \alpha_2$ is the resolvent of α_1 and α_2

Model Counting

$1 : x_1 \lor x_2$	$2 : \neg x_2 \lor x_3$	$3 : \neg x_1 \lor \neg x_2 \lor \neg y_1$	$4 : x_1 \lor \neg x_3 \lor y_1$
$5 : x_2 \lor \neg x_3 \lor y_2$	$6 : x_1 \lor \neg x_3 \lor \neg y_2$	$7 : y_3 \lor x_2$	$8 : \neg y_3 \lor \neg x_2 \lor \neg x_3$
$9 : \neg y_3 \lor x_1$	$10 : \neg y_3 \lor \neg y_2 \lor x_3$	$11 : y_3 \lor y_2 \lor x_2$	

- Propositional model counting aims to determine the number of models $\|\Sigma\|$ of Σ over the set $\mathrm{Var}(\Sigma)$ of variables occurring in Σ
- For the running example, $\mathrm{Var}(\Sigma) = \{x_1, x_2, x_3, y_1, y_2, y_3\}$, $\|\Sigma\| = 9$ and the models of Σ are:

$$\{\neg x_1, x_2, x_3, y_1, \neg y_2, \neg y_3\}$$
$$\{x_1, \neg x_2, x_3, y_1, y_2, y_3\}$$
$$\{x_1, \neg x_2, x_3, \neg y_1, y_2, y_3\}$$
$$\{x_1, \neg x_2, \neg x_3, \neg y_1, y_2, y_3\}$$
$$\{x_1, \neg x_2, \neg x_3, y_1, \neg y_2, y_3\}$$
$$\{x_1, \neg x_2, \neg x_3, y_1, y_2, y_3\}$$
$$\{x_1, \neg x_2, \neg x_3, \neg y_1, \neg y_2, y_3\}$$
$$\{x_1, x_2, x_3, \neg y_1, y_2, \neg y_3\}$$
$$\{x_1, x_2, x_3, \neg y_1, \neg y_2, \neg y_3\}$$

Projected Model Counting

$1 : x_1 \lor x_2$	$2 : \neg x_2 \lor x_3$	$3 : \neg x_1 \lor \neg x_2 \lor \neg y_1$	$4 : x_1 \lor \neg x_3 \lor y_1$
$5 : x_2 \lor \neg x_3 \lor y_2$	$6 : x_1 \lor \neg x_3 \lor \neg y_2$	$7 : y_3 \lor x_2$	$8 : \neg y_3 \lor \neg x_2 \lor \neg x_3$
$9 : \neg y_3 \lor x_1$	$10 : \neg y_3 \lor \neg y_2 \lor x_3$	$11 : y_3 \lor y_2 \lor x_2$	

- Given (X, Y) a partition of $Var(\Sigma)$, PMC consists in computing $\|\exists Y.\Sigma\|$, the number of interpretations over X, which coincide on X with a model of Σ
- For the running example, $Var(\exists y_1 y_2 y_3.\Sigma) = \{x_1, x_2, x_3\}$, $\|\exists y_1 y_2 y_3.\Sigma\| = 4$ and the models of $\exists y_1 y_2 y_3.\Sigma$ are:

$$\{\neg x_1, x_2, x_3\}$$
$$\{x_1, \neg x_2, x_3\}$$
$$\{x_1, \neg x_2, \neg x_3\}$$
$$\{x_1, x_2, x_3\}$$

Blocked Clause Elimination

- Blocked Clause Elimination (BCE) removes specific clauses, the so-called blocked clauses, from CNF formulas

- A literal ℓ within a clause α is a blocking literal if ℓ blocks α w.r.t. $\mathbf{\Sigma}$
  - This occurs when, $\forall \alpha' \in \mathbf{\Sigma}$ s.t. $\bar{\ell} \in \alpha'$, the resulting resolvent $\alpha \oplus \alpha'$ of α and α' on ℓ is a tautology

- A clause is considered blocked if it contains a literal that blocks it

- Applying BCE to $\mathbf{\Sigma}$ leads to remove every clause containing a blocking literal and by repeating the process iteratively until no blocked literal exists

$1 : x_1 \vee x_2$	$2 : \neg x_2 \vee x_3$	$3 : \neg x_1 \vee \neg x_2 \vee \neg y_1$	$4 : x_1 \vee \neg x_3 \vee y_1$
$5 : x_2 \vee \neg x_3 \vee y_2$	$6 : x_1 \vee \neg x_3 \vee \neg y_2$	$7 : y_3 \vee x_2$	$8 : \neg y_3 \vee \neg x_2 \vee \neg x_3$
$9 : \neg y_3 \vee x_1$	$10 : \neg y_3 \vee \neg y_2 \vee x_3$	$11 : y_3 \vee y_2 \vee x_2$	

Blocked Clause Elimination

- Blocked Clause Elimination (BCE) removes specific clauses, the so-called blocked clauses, from CNF formulas

- A literal ℓ within a clause α is a blocking literal if ℓ blocks α w.r.t. $\mathbf{\Sigma}$
  - This occurs when, $\forall \alpha' \in \mathbf{\Sigma}$ s.t. $\bar{\ell} \in \alpha'$, the resulting resolvent $\alpha \oplus \alpha'$ of α and α' on ℓ is a tautology

- A clause is considered blocked if it contains a literal that blocks it

- Applying BCE to $\mathbf{\Sigma}$ leads to remove every clause containing a blocking literal and by repeating the process iteratively until no blocked literal exists

$1 : x_1 \lor x_2$	$2 : \neg x_2 \lor x_3$	$3 : \neg x_1 \lor \neg x_2 \lor \neg y_1$	$4 : x_1 \lor \neg x_3 \lor y_1$
$5 : x_2 \lor \neg x_3 \lor y_2$	$6 : x_1 \lor \neg x_3 \lor \neg y_2$	$7 : y_3 \lor x_2$	$8 : \neg y_3 \lor \neg x_2 \lor \neg x_3$
$9 : \neg y_3 \lor x_1$	$10 : \neg y_3 \lor \neg y_2 \lor x_3$	$11 : y_3 \lor y_2 \lor x_2$	

Blocked Clause Elimination

- Blocked Clause Elimination (BCE) removes specific clauses, the so-called blocked clauses, from CNF formulas

- A literal ℓ within a clause α is a blocking literal if ℓ blocks α w.r.t. $\mathbf{\Sigma}$
  - This occurs when, $\forall \alpha' \in \mathbf{\Sigma}$ s.t. $\bar{\ell} \in \alpha'$, the resulting resolvent $\alpha \oplus \alpha'$ of α and α' on ℓ is a tautology

- A clause is considered blocked if it contains a literal that blocks it

- Applying BCE to $\mathbf{\Sigma}$ leads to remove every clause containing a blocking literal and by repeating the process iteratively until no blocked literal exists

$1 : x_1 \vee x_2$	$2 : \neg x_2 \vee x_3$	$3 : \neg x_1 \vee \neg x_2 \vee \neg y_1$	$4 : x_1 \vee \neg x_3 \vee y_1$
$5 : x_2 \vee \neg x_3 \vee y_2$	$6 : x_1 \vee \neg x_3 \vee \neg y_2$	$7 : y_3 \vee x_2$	$8 : \neg y_3 \vee \neg x_2 \vee \neg x_3$
$9 : \neg y_3 \vee x_1$	$10 : \neg y_3 \vee \neg y_2 \vee x_3$	$11 : y_3 \vee y_2 \vee x_2$	

Blocked Clause Elimination

- Blocked Clause Elimination (BCE) removes specific clauses, the so-called blocked clauses, from CNF formulas

- A literal ℓ within a clause α is a blocking literal if ℓ blocks α w.r.t. $\boldsymbol{\Sigma}$
  - This occurs when, $\forall \alpha' \in \boldsymbol{\Sigma}$ s.t. $\bar{\ell} \in \alpha'$, the resulting resolvent $\alpha \oplus \alpha'$ of α and α' on ℓ is a tautology

- A clause is considered blocked if it contains a literal that blocks it

- Applying BCE to $\boldsymbol{\Sigma}$ leads to remove every clause containing a blocking literal and by repeating the process iteratively until no blocked literal exists

$1 : x_1 \vee x_2$	$2 : \neg x_2 \vee x_3$	$3 : \neg x_1 \vee \neg x_2 \vee \neg y_1$	$4 : x_1 \vee \neg x_3 \vee y_1$
$5 : x_2 \vee \neg x_3 \vee y_2$	$6 : x_1 \vee \neg x_3 \vee \neg y_2$	$7 : y_3 \vee x_2$	$8 : \neg y_3 \vee \neg x_2 \vee \neg x_3$
$9 : \neg y_3 \vee x_1$	$10 : \neg y_3 \vee \neg y_2 \vee x_3$	$11 : y_3 \vee y_2 \vee x_2$	

Blocked Clause Elimination

- Blocked Clause Elimination (BCE) removes specific clauses, the so-called blocked clauses, from CNF formulas
- A literal ℓ within a clause α is a blocking literal if ℓ blocks α w.r.t. $\mathbf{\Sigma}$
  - This occurs when, $\forall \alpha' \in \mathbf{\Sigma}$ s.t. $\bar{\ell} \in \alpha'$, the resulting resolvent $\alpha \oplus \alpha'$ of α and α' on ℓ is a tautology
- A clause is considered blocked if it contains a literal that blocks it
- Applying BCE to $\mathbf{\Sigma}$ leads to remove every clause containing a blocking literal and by repeating the process iteratively until no blocked literal exists

$1 : x_1 \vee x_2$	$2 : \neg x_2 \vee x_3$	$3 : \neg x_1 \vee \neg x_2 \vee \neg y_1$	$4 : x_1 \vee \neg x_3 \vee y_1$
$5 : x_2 \vee \neg x_3 \vee y_2$	$6 : x_1 \vee \neg x_3 \vee \neg y_2$	$7 : y_3 \vee x_2$	$8 : \neg y_3 \vee \neg x_2 \vee \neg x_3$
$9 : \neg y_3 \vee x_1$	$10 : \neg y_3 \vee \neg y_2 \vee x_3$	$11 : y_3 \vee y_2 \vee x_2$	

Blocked Clause Elimination

- Blocked Clause Elimination (BCE) removes specific clauses, the so-called blocked clauses, from CNF formulas

- A literal ℓ within a clause α is a blocking literal if ℓ blocks α w.r.t. $\boldsymbol{\Sigma}$
  - This occurs when, $\forall \alpha' \in \boldsymbol{\Sigma}$ s.t. $\bar{\ell} \in \alpha'$, the resulting resolvent $\alpha \oplus \alpha'$ of α and α' on ℓ is a tautology

- A clause is considered blocked if it contains a literal that blocks it

- Applying BCE to $\boldsymbol{\Sigma}$ leads to remove every clause containing a blocking literal and by repeating the process iteratively until no blocked literal exists

$1 : x_1 \vee x_2$	$2 : \neg x_2 \vee x_3$	$3 : \neg x_1 \vee \neg x_2 \vee \neg y_1$	$4 : x_1 \vee \neg x_3 \vee y_1$
$5 : x_2 \vee \neg x_3 \vee y_2$	$6 : x_1 \vee \neg x_3 \vee \neg y_2$	$7 : y_3 \vee x_2$	$8 : \neg y_3 \vee \neg x_2 \vee \neg x_3$
$9 : \neg y_3 \vee x_1$	$10 : \neg y_3 \vee \neg y_2 \vee x_3$	$11 : y_3 \vee y_2 \vee x_2$	

Blocked Clause Elimination

- Blocked Clause Elimination (BCE) removes specific clauses, the so-called blocked clauses, from CNF formulas
- A literal ℓ within a clause α is a blocking literal if ℓ blocks α w.r.t. $\boldsymbol{\Sigma}$
  - This occurs when, $\forall \alpha' \in \boldsymbol{\Sigma}$ s.t. $\bar{\ell} \in \alpha'$, the resulting resolvent $\alpha \oplus \alpha'$ of α and α' on ℓ is a tautology
- A clause is considered blocked if it contains a literal that blocks it
- Applying BCE to $\boldsymbol{\Sigma}$ leads to remove every clause containing a blocking literal and by repeating the process iteratively until no blocked literal exists

$1 : x_1 \vee x_2$	$2 : \neg x_2 \vee x_3$	$3 : \neg x_1 \vee \neg x_2 \vee \neg y_1$	$4 : x_1 \vee \neg x_3 \vee y_1$
$5 : x_2 \vee \neg x_3 \vee y_2$	$6 : x_1 \vee \neg x_3 \vee \neg y_2$	$7 : y_3 \vee x_2$	$8 : \neg y_3 \vee \neg x_2 \vee \neg x_3$
$9 : \neg y_3 \vee x_1$	$10 : \neg y_3 \vee \neg y_2 \vee x_3$	$11 : y_3 \vee y_2 \vee x_2$	

Blocked Clause Elimination

- Blocked Clause Elimination (BCE) removes specific clauses, the so-called blocked clauses, from CNF formulas
- A literal ℓ within a clause α is a blocking literal if ℓ blocks α w.r.t. Σ
  - This occurs when, $\forall \alpha' \in \Sigma$ s.t. $\bar{\ell} \in \alpha'$, the resulting resolvent $\alpha \oplus \alpha'$ of α and α' on ℓ is a tautology
- A clause is considered blocked if it contains a literal that blocks it
- Applying BCE to Σ leads to remove every clause containing a blocking literal and by repeating the process iteratively until no blocked literal exists

$1 : x_1 \lor x_2$	$2 : \neg x_2 \lor x_3$	$3 : \neg x_1 \lor \neg x_2 \lor \neg y_1$	$4 : x_1 \lor \neg x_3 \lor y_1$
$5 : x_2 \lor \neg x_3 \lor y_2$	$6 : x_1 \lor \neg x_3 \lor \neg y_2$	$7 : y_3 \lor x_2$	$8 : \neg y_3 \lor \neg x_2 \lor \neg x_3$
$9 : \neg y_3 \lor x_1$	$10 : \neg y_3 \lor \neg y_2 \lor x_3$	$11 : y_3 \lor y_2 \lor x_2$	

Blocked Clause Elimination

- Blocked Clause Elimination (BCE) removes specific clauses, the so-called blocked clauses, from CNF formulas
- A literal ℓ within a clause α is a blocking literal if ℓ blocks α w.r.t. Σ
  - This occurs when, $\forall \alpha' \in \Sigma$ s.t. $\overline{\ell} \in \alpha'$, the resulting resolvent $\alpha \oplus \alpha'$ of α and α' on ℓ is a tautology
- A clause is considered blocked if it contains a literal that blocks it
- Applying BCE to Σ leads to remove every clause containing a blocking literal and by repeating the process iteratively until no blocked literal exists

$1 : x_1 \vee x_2$	$2 : \neg x_2 \vee x_3$	$3 : \neg x_1 \vee \neg x_2 \vee \neg y_1$	$4 : x_1 \vee \neg x_3 \vee y_1$
$5 : x_2 \vee \neg x_3 \vee y_2$	$6 : x_1 \vee \neg x_3 \vee \neg y_2$	$7 : y_3 \vee x_2$	$8 : \neg y_3 \vee \neg x_2 \vee \neg x_3$
$9 : \neg y_3 \vee x_1$	$10 : \neg y_3 \vee \neg y_2 \vee x_3$	$11 : y_3 \vee y_2 \vee x_2$	

Blocked Clause Elimination

- Blocked Clause Elimination (BCE) removes specific clauses, the so-called blocked clauses, from CNF formulas

- A literal ℓ within a clause α is a blocking literal if ℓ blocks α w.r.t. Σ
  - This occurs when, $\forall \alpha' \in \Sigma$ s.t. $\bar{\ell} \in \alpha'$, the resulting resolvent $\alpha \oplus \alpha'$ of α and α' on ℓ is a tautology

- A clause is considered blocked if it contains a literal that blocks it

- Applying BCE to Σ leads to remove every clause containing a blocking literal and by repeating the process iteratively until no blocked literal exists

$1 : x_1 \vee x_2$	$2 : \neg x_2 \vee x_3$	$3 : \neg x_1 \vee \neg x_2 \vee \neg y_1$	$4 : x_1 \vee \neg x_3 \vee y_1$
$5 : x_2 \vee \neg x_3 \vee y_2$	$6 : x_1 \vee \neg x_3 \vee \neg y_2$	$7 : y_3 \vee x_2$	$8 : \neg y_3 \vee \neg x_2 \vee \neg x_3$
$9 : \neg y_3 \vee x_1$	$10 : \neg y_3 \vee \neg y_2 \vee x_3$	$11 : y_3 \vee y_2 \vee x_2$	

Blocked Clause Elimination

- Blocked Clause Elimination (BCE) removes specific clauses, the so-called blocked clauses, from CNF formulas
- A literal ℓ within a clause α is a blocking literal if ℓ blocks α w.r.t. $\mathbf{\Sigma}$
  - This occurs when, $\forall \alpha' \in \mathbf{\Sigma}$ s.t. $\bar{\ell} \in \alpha'$, the resulting resolvent $\alpha \oplus \alpha'$ of α and α' on ℓ is a tautology
- A clause is considered blocked if it contains a literal that blocks it
- Applying BCE to $\mathbf{\Sigma}$ leads to remove every clause containing a blocking literal and by repeating the process iteratively until no blocked literal exists

$1 : x_1 \vee x_2$	$2 : \neg x_2 \vee x_3$	$3 : \neg x_1 \vee \neg x_2 \vee \neg y_1$	$4 : x_1 \vee \neg x_3 \vee y_1$
$5 : x_2 \vee \neg x_3 \vee y_2$	$6 : x_1 \vee \neg x_3 \vee \neg y_2$	$7 : y_3 \vee x_2$	$8 : \neg y_3 \vee \neg x_2 \vee \neg x_3$
$9 : \neg y_3 \vee x_1$	$10 : \neg y_3 \vee \neg y_2 \vee x_3$	$11 : y_3 \vee y_2 \vee x_2$	

Blocked Clause Elimination

- Blocked Clause Elimination (BCE) removes specific clauses, the so-called blocked clauses, from CNF formulas
- A literal ℓ within a clause α is a blocking literal if ℓ blocks α w.r.t. $\boldsymbol{\Sigma}$
  - This occurs when, $\forall \alpha' \in \boldsymbol{\Sigma}$ s.t. $\bar{\ell} \in \alpha'$, the resulting resolvent $\alpha \oplus \alpha'$ of α and α' on ℓ is a tautology
- A clause is considered blocked if it contains a literal that blocks it
- Applying BCE to $\boldsymbol{\Sigma}$ leads to remove every clause containing a blocking literal and by repeating the process iteratively until no blocked literal exists

$1 : x_1 \vee x_2$	$2 : \neg x_2 \vee x_3$	$3 : \neg x_1 \vee \neg x_2 \vee \neg y_1$	$4 : x_1 \vee \neg x_3 \vee y_1$
$5 : x_2 \vee \neg x_3 \vee y_2$	$6 : x_1 \vee \neg x_3 \vee \neg y_2$	$7 : y_3 \vee x_2$	$8 : \neg y_3 \vee \neg x_2 \vee \neg x_3$
$9 : \neg y_3 \vee x_1$	$10 : \neg y_3 \vee \neg y_2 \vee x_3$	$11 : y_3 \vee y_2 \vee x_2$	

BCE for PMC

- BCE guarantees the preservation of satisfiability but not necessarily the preservation of the number of models
  - For the previous example, $BCE(\mathbf{\Sigma}) = \emptyset$: this corresponds to $2^6 = 64$ models over $Var(\mathbf{\Sigma})$, which differs from $\|\mathbf{\Sigma}\| = 9$

- The picture changes when addressing the projected model counting problem:

Proposition

If a non-tautological clause $\alpha_i \in \mathbf{\Sigma}$ is blocked by a literal $\ell \in \alpha_i$ with $\ell \in \{x, \neg x\}$, then $\exists x.\mathbf{\Sigma}$ is logically equivalent to $\exists x.\mathbf{\Sigma}'$, where $\mathbf{\Sigma}' = \mathbf{\Sigma} \setminus \{\alpha_i\}$.

- When focusing on subformulas containing only projected variables, the requirement is only to ensure satisfiability

BlockedClauseManager:initialization

- To efficiently identify clauses eligible for removal through BCE, we introduce a BlockedClauseManager
- This manager identifies clauses eligible for elimination because of a blocked literal, by using a mechanism akin to the one considered for watched literals
  - backtrack free handling!
- A clause α is not blocked on a literal $\bar{\ell} \in \alpha$ if there exists another clause α' such that $\bar{\ell} \in \alpha'$ and $\alpha \oplus \alpha'$ is not a tautology
  - $\forall \ell \in \alpha$ such that $\text{Var}(\ell) \subseteq X$, either ℓ is assigned or there must exist a clause α' where $\bar{\ell} \in \alpha'$, and $\alpha \oplus \alpha' \not\equiv \top$

$1 : x_1 \vee x_2$	$2 : \neg x_2 \vee x_3$	$3 : \neg x_1 \vee \neg x_2 \vee \neg y_1$	$4 : x_1 \vee \neg x_3 \vee y_1$
$5 : x_2 \vee \neg x_3 \vee y_2$	$6 : x_1 \vee \neg x_3 \vee \neg y_2$	$7 : y_3 \vee x_2$	$8 : \neg y_3 \vee \neg x_2 \vee \neg x_3$
$9 : \neg y_3 \vee x_1$	$10 : \neg y_3 \vee \neg y_2 \vee x_3$	$11 : y_3 \vee y_2 \vee x_2$	

- When evaluating α_3, literal $\neg y_1$ cannot be associated with a clause from Σ without resulting in a tautology.
  - α_3 can be safely removed from Σ.

BlockedClauseManager:initialization

- To efficiently identify clauses eligible for removal through BCE, we introduce a BlockedClauseManager
- This manager identifies clauses eligible for elimination because of a blocked literal, by using a mechanism akin to the one considered for watched literals
  - backtrack free handling!
- A clause α is not blocked on a literal $\bar{\ell} \in \alpha$ if there exists another clause α' such that $\bar{\ell} \in \alpha'$ and $\alpha \oplus \alpha'$ is not a tautology
  - $\forall \ell \in \alpha$ such that $\text{Var}(\ell) \subseteq X$, either ℓ is assigned or there must exist a clause α' where $\bar{\ell} \in \alpha'$, and $\alpha \oplus \alpha' \not\equiv \top$

$1 : x_1 \lor x_2$	$2 : \neg x_2 \lor x_3$	$3 : \neg x_1 \lor \neg x_2 \lor \neg y_1$	$4 : x_1 \lor \neg x_3 \lor y_1$
$5 : x_2 \lor \neg x_3 \lor y_2$	$6 : x_1 \lor \neg x_3 \lor \neg y_2$	$7 : y_3 \lor x_2$	$8 : \neg y_3 \lor \neg x_2 \lor \neg x_3$
$9 : \neg y_3 \lor x_1$	$10 : \neg y_3 \lor \neg y_2 \lor x_3$	$11 : y_3 \lor y_2 \lor x_2$	

- When evaluating α_3, literal $\neg y_1$ cannot be associated with a clause from Σ without resulting in a tautology.
  - α_3 can be safely removed from Σ.

BlockedClauseManager:watches

- The watches structure associates each clause $\alpha \in \Sigma$ with a set of triples watches[α] that are being watched by α
  - each triple $(\ell, \alpha, \mathcal{C})$ is s.t. $\alpha \in \Sigma$, $\ell \in \alpha$ and $\mathcal{C} = \{\alpha' \in \Sigma | \alpha \oplus \alpha' \neq \top\}$

$1 : x_1 \lor x_2$	$2 : \neg x_2 \lor x_3$	$3 : \neg x_1 \lor \neg x_2 \lor \neg y_1$	$4 : x_1 \lor \neg x_3 \lor y_1$
$5 : x_2 \lor \neg x_3 \lor y_2$	$6 : x_1 \lor \neg x_3 \lor \neg y_2$	$7 : y_3 \lor x_2$	$8 : \neg y_3 \lor \neg x_2 \lor \neg x_3$
$9 : \neg y_3 \lor x_1$	$10 : \neg y_3 \lor \neg y_2 \lor x_3$	$11 : y_3 \lor y_2 \lor x_2$	

$$\mathrm{watches}[6] = \{(y_2, 5, \{6\}), (y_2, 11, \{6\})\}$$
$$\mathrm{watches}[5] = \{(\neg y_2, 6, \{5, 11\})\}$$
$$\mathrm{watches}[9] = \{(y_3, 7, \{9, 10\}), (y_3, 11, \{9\})\}$$
$$\mathrm{watches}[7] = \{(\neg y_3, 9, \{7, 11\}), (\neg y_3, 10, \{7\})\}$$
$$\mathrm{watches}[3] = \emptyset$$
$$\mathrm{watches}[4] = \emptyset$$
$$\mathrm{watches}[8] = \emptyset$$
$$\mathrm{watches}[10] = \emptyset$$
$$\mathrm{watches}[11] = \emptyset$$

BlockedClauseManager:propagate

- When conditioning a formula by a literal ℓ, without rendering it unsatisfiable, there is no need to consider clauses shortened by this conditioning operation
  - $\alpha \setminus \{\bar{\ell}\} \in \Sigma_{|\ell}$ cannot be blocked by any literal $\ell' \in \alpha \setminus \{\ell\}$ in $\Sigma_{|\ell}$
- It is enough to trigger the clauses that are removed either because they are satisfied or removed by BCE

$1 : x_1 \vee x_2$	$2 : \neg x_2 \vee x_3$	$3 : \neg x_1 \vee \neg x_2 \vee \neg y_1$	$4 : x_1 \vee \neg x_3 \vee y_1$
$5 : x_2 \vee \neg x_3 \vee y_2$	$6 : x_1 \vee \neg x_3 \vee \neg y_2$	$7 : y_3 \vee x_2$	$8 : \neg y_3 \vee \neg x_2 \vee \neg x_3$
$9 : \neg y_3 \vee x_1$	$10 : \neg y_3 \vee \neg y_2 \vee x_3$	$11 : y_3 \vee y_2 \vee x_2$	

- If x_1 is set to 1, then clauses $1, 6, 9$ become satisfied $\Rightarrow \{11\}$ should be removed

...

$\text{watches}[5] = \{(\neg y_2, 10, \{5\})\}$
$\text{watches}[9] = \{(y_3, 11, \{9\})\}$
$\text{watches}[6] = \{((y_2, 11, \{6\})\}$
$\text{watches}[10] = \{(y_2, 5, \{6, 10\}), (y_3, 7, \{9, 10\})\}$

BlockedClauseManager:propagate

- When conditioning a formula by a literal ℓ, without rendering it unsatisfiable, there is no need to consider clauses shortened by this conditioning operation
  - $\alpha \setminus \{\bar{\ell}\} \in \Sigma_{|\ell}$ cannot be blocked by any literal $\ell' \in \alpha \setminus \{\ell\}$ in $\Sigma_{|\ell}$
- It is enough to trigger the clauses that are removed either because they are satisfied or removed by BCE

$1 : x_1 \vee x_2$	$2 : \neg x_2 \vee x_3$	$3 : \neg x_1 \vee \neg x_2 \vee \neg y_1$	$4 : x_1 \vee \neg x_3 \vee y_1$
$5 : x_2 \vee \neg x_3 \vee y_2$	$6 : x_1 \vee \neg x_3 \vee \neg y_2$	$7 : y_3 \vee x_2$	$8 : \neg y_3 \vee \neg x_2 \vee \neg x_3$
$9 : \neg y_3 \vee x_1$	$10 : \neg y_3 \vee \neg y_2 \vee x_3$	$11 : y_3 \vee y_2 \vee x_2$	

- If x_1 is set to 1, then clauses $1, 6, 9$ become satisfied $\Rightarrow \{11\}$ should be removed

...

$\text{watches}[5] = \{(\neg y_2, 10, \{5\})\}$
$\text{watches}[9] = \{(y_3, 11, \{9\})\}$
$\text{watches}[6] = \{((y_2, 11, \{6\})\}$
$\text{watches}[10] = \{(y_2, 5, \{6, 10\}), (y_3, 7, \{9, 10\})\}$

Experimental setup

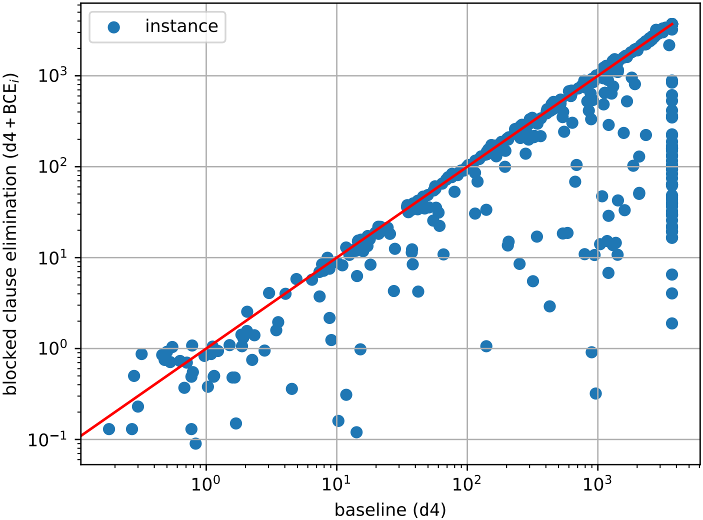
- We used 500 CNF instances from the 2021, 2022 and 2023 Model Counting competitions
  - 200 from the 2021 competition, 200 from the 2022, and 100 from the 2023 competition
- A time-out of 3600 seconds and a memory limit of 32 GiB have been considered
- Three different versions of the projected model counter d4 have been used for the evaluation:
  - d4: this is the standard version of d4
  - d4+BCE_p: this version of d4 incorporates BCE performed once during a preprocessing phase
  - d4+BCE_i: in this version of d4, BCE is performed dynamically throughout the search achieved by the model counter
- For all the versions under consideration, a preprocessing step of 60 seconds was conducted. This preprocessing involves running BiPe, followed by occurrence elimination and vivification preprocessings for 10 iterations.

Numbers of instances solved

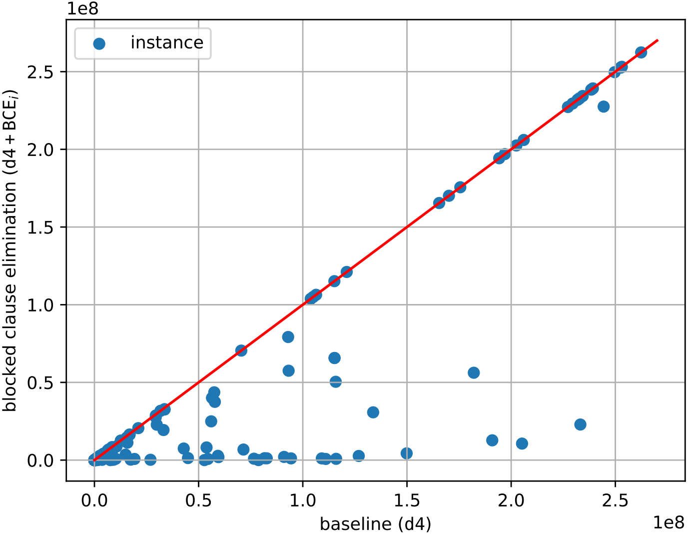
- The following table gives:
  - the number of instances solved
  - the number of memory out (MO) encountered

	2021 (200)	2022 (200)	2023 (100)	All (500)
d4	139 (56 MO)	149 (24 MO)	73 (9 MO)	361 (89 MO)
d4+BCE_p	139 (56 MO)	149 (25 MO)	73 (9 MO)	361 (90 MO)
d4+BCE_i	172 (23 MO)	163 (8 MO)	78 (4 MO)	413 (35 MO)

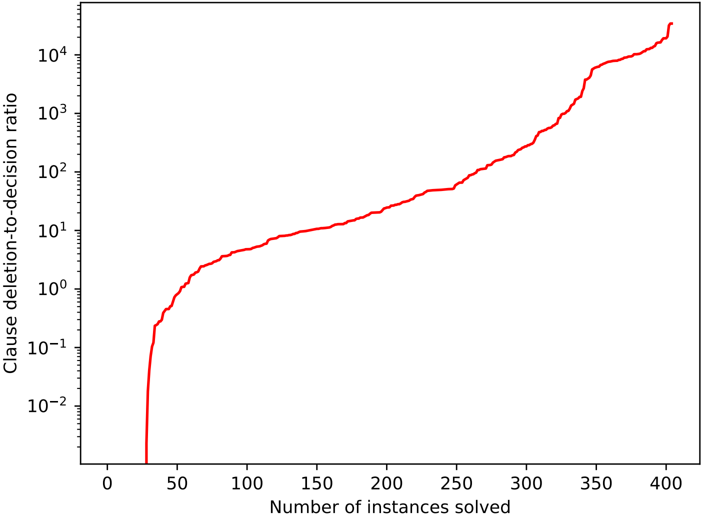
Run time comparison between d4 and d4+BCE$_i$



Number of decisions made by d4 vs. d4+BCE$_i$



Clause deletion / Number of decisions



Conclusion

- Blocked clause elimination can be used as a sound preprocessing technique for projected model counting
- Introduction of a new data structure and corresponding algorithms tailored for leveraging blocked clause elimination dynamically during projected model counting
- Implementation of this machinery into the projected model counter d4
- Our experimental results show the benefits of using blocked clause elimination for projected model counting

Perspectives

- Considering the elimination of resolution-asymmetric tautologies (RAT), of covered or propagation-redundant (PR) clauses
- Designing novel branching heuristics to prioritize the elimination of clauses that prevent the removal of blocked clauses
- Assessing the benefits of using blocked clause elimination for other reasoning tasks
  - weighted Max#SAT problem
  - counting tree models of QBF formulas
- Combining definability and blocking clause elimination for the model counting problem