# Preprocessing QBF: Failed Literals and Quantified Blocked Clause Elimination

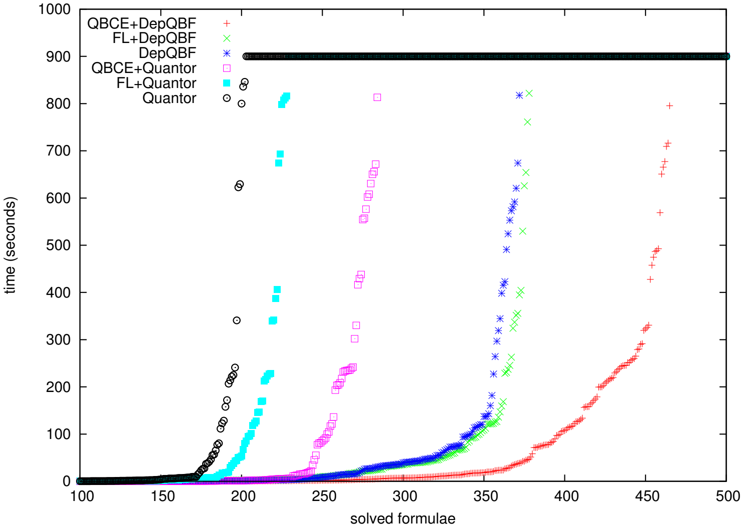
Florian Lonsing
(joint work with Armin Biere and Martina Seidl)

Institute for Formal Models and Verification (FMV)
Johannes Kepler University, Linz, Austria
`http://fmv.jku.at`

Deduction at Scale Seminar
Ringberg Castle, Tegernsee, Germany
March 7 - 11, 2011

**Preprocessing Techniques for Quantified Boolean Formulae (QBF)**

- Failed literals (FL) and quantified blocked clause elimination (QBCE).
- Positive effects on search- and elimination-based solvers.



# Overview

**Part 1: Preliminaries**

- From propositional logic (SAT) to QBF.
- QBF semantics.

**Part 2: Failed Literal Detection (FL)**

- Paper submitted to SAT'11.
- Necessary assignments and QBF models.

**Part 3: Quantified Blocked Clause Elimination (QBCE)**

- Paper submitted to CADE'11.
- From BCE for SAT to QBCE for QBF.

*Part 1: Preliminaries*

**Propositional Logic (SAT):**

- Our focus: formulae in conjunctive normal form (CNF).
- Set of Boolean variables $V := \{x_1, \ldots, x_m\}$.
- Literals $l := v$ or $l := \neg v$ for $v \in V$.
- Clauses $C_i := (l_1 \vee \ldots \vee l_{k_i})$.
- CNF $\phi := \bigwedge C_i$.

**Quantified Boolean Formulae (QBF):**

- Prenex CNF: quantifier-free CNF over quantified Boolean variables.
- PCNF $Q_1 S_1 \ldots Q_n S_n.\ \phi$, where $Q_i \in \{\exists, \forall\}$, scopes $S_i$.
- Scope $S_i$: set of quantified variables.
- $Q_i S_i \leq Q_{i+1} S_{i+1}$: scopes are linearly ordered.

**Example**

Clauses (CNFs) are sets of literals (clauses).
A CNF: $\{x, \overline{y}\}, \{\overline{x}, y\}$ and a PCNF: $\forall x \exists y.\ \{x, \overline{y}\}, \{\overline{x}, y\}$.

**Assignment Trees (AT):**

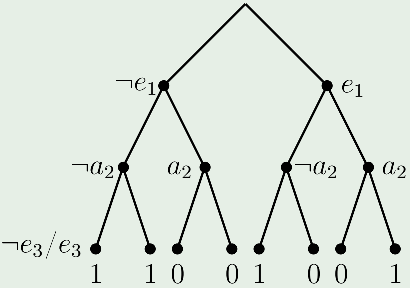
- Assignment $A : V \rightarrow \{true, false\}$ maps variables to truth values.
- Paths from root to a leaf in AT represent assignments.
- Nodes along path (except root) assign truth values to variables.

**CNF-Model:**

- A path in the assignment tree of a CNF $\phi$ which satisfies all clauses.
- CNF $\phi$ is satisfiable iff it has a CNF-model $m$: $m \models \phi$.

## Example

$$\phi := \{e_1, \neg a_2, e_3\}, \{e_1, \neg a_2, \neg e_3\}, \{\neg e_1, a_2, \neg e_3\}, \{\neg e_1, \neg a_2, e_3\}$$



**Assignment Trees (AT):**

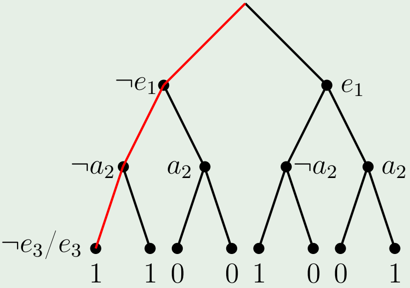
- Assignment $A : V \to \{true, false\}$ maps variables to truth values.
- Paths from root to a leaf in AT represent assignments.
- Nodes along path (except root) assign truth values to variables.

**CNF-Model:**

- A path in the assignment tree of a CNF $\phi$ which satisfies all clauses.
- CNF $\phi$ is satisfiable iff it has a CNF-model $m$: $m \models \phi$.

## Example

$$\phi := \{e_1, \neg a_2, e_3\}, \{e_1, \neg a_2, \neg e_3\}, \{\neg e_1, a_2, \neg e_3\}, \{\neg e_1, \neg a_2, e_3\}$$



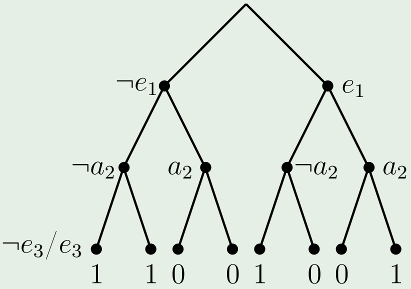
# Semantics: From SAT to QBF

**PCNF-Model:** $\psi := Q_1 S_1 \ldots Q_n S_n.\ \phi$

- An (incomplete) AT where *every* path is a CNF-model of CNF part $\phi$.
- Restriction: nodes which assign $\forall$-variables have exactly one sibling.
- PCNF $\psi$ is satisfiable iff it has a PCNF-model $m$: $m \models \psi$.

## Example

$$
\begin{array}{rcl}
\psi & := & \exists e_1 \forall a_2 \exists e_3.\ \phi \\
\phi & := & \{e_1, \neg a_2, e_3\}, \\
 & & \{e_1, \neg a_2, \neg e_3\}, \\
 & & \{\neg e_1, a_2, \neg e_3\}, \\
 & & \{\neg e_1, \neg a_2, e_3\}
\end{array}
$$

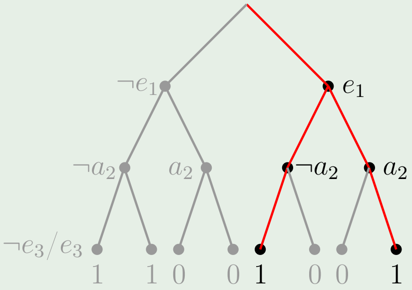


**PCNF-Model:** $\psi := Q_1 S_1 \ldots Q_n S_n.\ \phi$

- An (incomplete) AT where *every* path is a CNF-model of CNF part $\phi$.
- Restriction: nodes which assign $\forall$-variables have exactly one sibling.
- PCNF $\psi$ is satisfiable iff it has a PCNF-model $m$: $m \models \psi$.

## Example

$$
\begin{aligned}
\psi &:= \exists e_1 \forall a_2 \exists e_3.\ \phi \\
\phi &:= \{e_1, \neg a_2, e_3\}, \\
&\quad \{e_1, \neg a_2, \neg e_3\}, \\
&\quad \{\neg e_1, a_2, \neg e_3\}, \\
&\quad \{\neg e_1, \neg a_2, e_3\}
\end{aligned}
$$



## Definition (Assignments of literals)

Given a PCNF $\psi$, the *assignment of a literal l* yields the formula $\psi[l]$ where clauses $Occs(l)$ and literals $\neg l$ in $Occs(\neg l)$ are deleted.

## Example

$$
\begin{aligned}
\psi &:= \exists e_1 \forall a_2 \exists e_3, e_4.\ \phi \\
\phi &:= \{e_1, a_2, e_3, e_4\}, \\
&\quad \{e_1, a_2, \neg e_4\}, \\
&\quad \{\neg e_1, e_3, \neg e_4\}, \\
&\quad \{\neg a_2, \neg e_3\}
\end{aligned}
$$

$\psi[e_4]$

# QBF Inference Rules (2/5)

## Definition (Universal Reduction)

Given a clause $C$, $UR(C) := C \setminus \{l_u \in L_\forall(C) \mid \nexists l_e \in L_\exists(C), l_u < l_e\}$.

## Example

$$\psi \quad := \quad \exists e_1 \forall a_2 \exists e_3, e_4.\ \phi$$

$$\phi \quad :=$$

$$\{e_1, a_2\},$$

$$\{\neg e_1, e_3\},$$

$$\{\neg a_2, \neg e_3\}$$

$$UR(\{e_1, a_2\})$$

**Definition (Pure Literal Rule)**

Given a PCNF $\psi$, a literal $l$ where $Occs(l) \neq \emptyset$ and $Occs(\neg l) = \emptyset$ is *pure*: if $q(l) = \exists$ then $\psi \equiv \psi[l]$, and if $q(l) = \forall$ then $\psi \equiv \psi[\neg l]$.

**Example**

$$\psi := \exists e_1 \forall a_2 \exists e_3, e_4.\ \phi$$

$$\phi := \{e_1\}, \{\neg e_1, e_3\}, \{\neg a_2, \neg e_3\}$$

Variable $a_2$ is pure: $\psi[a_2]$ (shortening clauses).

# QBF Inference Rules (4/5)

## Definition (Unit Clause Rule)

Given a PCNF $\psi$. A clause $C \in \psi$ where $UR(C) = \{l\}$ is *unit* and $\psi \equiv \psi[l]$.

## Example

$$\begin{aligned}
\psi &:= \exists e_1 \forall a_2 \exists e_3, e_4.\ \phi \\
\phi &:= \\
&\quad \{e_1\}, \\
&\quad \{\neg e_1, e_3\}, \\
&\quad \{\neg e_3\}
\end{aligned}$$

Clauses $\{e_1\}$ and $\{\neg e_3\}$ are unit: $\psi[e_1][\neg e_3]$.

# QBF Inference Rules (5/5)

## Definition (Boolean Constraint Propagation)

Given a PCNF $\psi$ and a literal $x$ called *assumption*. Formula $BCP(\psi, x)$ is obtained from $\psi[x]$ by applying UR, unit clause and pure literal rule.

## Example

$$\psi := \exists e_1 \forall a_2 \exists e_3, e_4.\ \phi$$

$$\phi :=$$

$$\{\}$$

Empty clause derived from assumption $e_4$:
$\emptyset \in BCP(\psi, e_4)$.

*Part 2: Failed Literal Detection (FL)*

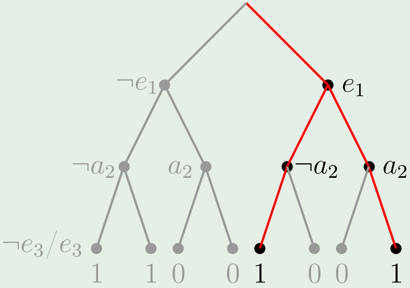
## Definition

Given PCNF $\psi$ and $x_i \in V$. Assignment $x_i \mapsto t$, where $t \in \{\mathit{false}, \mathit{true}\}$, is necessary for satisfiability of $\psi$ iff $x_i \mapsto t$ is part of *every* path in every PCNF-model of $\psi$.

## Example

$$\begin{aligned}
\psi &:= \exists e_1 \forall a_2 \exists e_3.\ \phi \\
\phi &:= \{e_1, \neg a_2, e_3\}, \\
&\quad\ \{e_1, \neg a_2, \neg e_3\}, \\
&\quad\ \{\neg e_1, a_2, \neg e_3\}, \\
&\quad\ \{\neg e_1, \neg a_2, e_3\}
\end{aligned}$$

- $e_1 \mapsto$ *true* is necessary for satisfiability of $\psi$.




**GOAL: Detection of (Subset of) Necessary Assignments in QBFs.**

- Exponential reduction of search space.

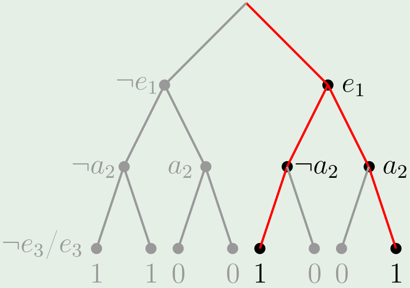
# Models and Necessary Assignments

## Definition

Given PCNF $\psi$ and $x_i \in V$. Assignment $x_i \mapsto t$, where $t \in \{false, true\}$, is necessary for satisfiability of $\psi$ iff $x_i \mapsto t$ is part of *every* path in every PCNF-model of $\psi$.

## Example

$$\begin{aligned}\psi &:= \exists e_1 \forall a_2 \exists e_3.\ \phi \\ \phi &:= \{e_1, \neg a_2, e_3\}, \\ &\quad \{e_1, \neg a_2, \neg e_3\}, \\ &\quad \{\neg e_1, a_2, \neg e_3\}, \\ &\quad \{\neg e_1, \neg a_2, e_3\}\end{aligned}$$




- $e_1 \mapsto$ *true* is necessary for satisfiability of $\psi$.

**GOAL: Detection of (Subset of) Necessary Assignments in QBFs.**

- Exponential reduction of search space.

# Motivation

**Failed Literal Detection (FL) for SAT:**

- BCP-based approach to detect subset of necessary assignments.
- Def. failed literal $x$ for CNF $\phi$: if $\emptyset \in BCP(\phi, x)$ then $\phi \equiv \phi \wedge \{\neg x\}$.
- FL based on deriving empty clause from assumption and BCP.

**FL for QBF:**

- Def.: failed literal $x$ for PCNF $\psi$: if $\psi \equiv \psi \wedge \{\neg x\}$.
- Problem: BCP-based approach like for SAT is unsound due to $\exists/\forall$ prefix.

**Example**

$\psi := \forall x \exists y. \{x, \neg y\}, \{\neg x, y\}$. We have $\emptyset \in BCP(\psi, y)$ but $\psi \not\equiv \psi \wedge \{\neg y\}$.

Our Work:

- Two orthogonal FL approaches for QBF.
- Soundness established by abstraction and Q-resolution.

**Failed Literal Detection (FL) for SAT:**

- BCP-based approach to detect subset of necessary assignments.
- Def. failed literal $x$ for CNF $\phi$: if $\emptyset \in BCP(\phi, x)$ then $\phi \equiv \phi \wedge \{\neg x\}$.
- FL based on deriving empty clause from assumption and BCP.

**FL for QBF:**

- Def.: failed literal $x$ for PCNF $\psi$: if $\psi \equiv \psi \wedge \{\neg x\}$.
- Problem: BCP-based approach like for SAT is unsound due to $\exists/\forall$ prefix.

**Example**

$\psi := \forall x \exists y. \{x, \neg y\}, \{\neg x, y\}$. We have $\emptyset \in BCP(\psi, y)$ but $\psi \not\equiv \psi \wedge \{\neg y\}$.

Our Work:

- Two orthogonal FL approaches for QBF.
- Soundness established by abstraction and Q-resolution.

**Problem:** $BCP(\psi, x)$ with assumption $x$ for FL on PCNF $\psi$ is unsound.

**Definition (Quantifier Abstraction)**

For $\psi := Q_1 S_1 \ldots Q_{i-1} S_{i-1} Q_i S_i \ldots \ldots Q_n S_n.\ \phi$, the quantifier abstraction of $\psi$ with respect to $S_i$ is $Abs(\psi, i) := \exists(S_1 \cup \ldots \cup S_{i-1}) Q_i S_i \ldots Q_n S_n.\ \phi$.

**Idea:** carry out BCP on abstraction of $\psi$.

- If $x \in S_i$ then treat all variables smaller than $x$ as existentially quantified.
- Example: $Abs(\exists x \forall y \exists z.\ \phi, 3) = \exists x \exists y \exists z.\ \phi$.
- Overapproximation: if $m \models \psi$ then $m \models Abs(\psi, i)$.

**Theorem**

*Given PCNF* $\psi := Q_1 S_1 \ldots Q_n S_n.\ \phi$ *and literal* $x$ *where* $v(x) \in S_i$. *If* $\emptyset \in BCP(Abs(\psi, i), x)$ *then* $\psi \equiv \psi \wedge \{\neg x\}$.

**Practical Application:**

- FL using BCP on abstraction is sound and runs in polynomial-time.

### Definition (Q-resolution)

Let $C_1, C_2$ be clauses with $v \in C_1, \neg v \in C_2$ and $q(v) = \exists$ [BKF95].

1. $C_1 \otimes C_2 := (UR(C_1) \cup UR(C_2)) \setminus \{v, \neg v\}$.
2. If $\{x, \neg x\} \subseteq C_1 \otimes C_2$ (tautology) then no Q-resolvent exists.
3. Otherwise, Q-resolvent $C := UR(C_1 \otimes C_2)$ of $C_1$ and $C_2$ on $v$: $\{C_1, C_2\} \vdash^* C$.

**Q-Resolution:** combination of propositional resolution and UR.

- For PCNF $\psi$, clause $C$: if $\psi \vdash^* C$ then $\psi \equiv \psi \wedge C$.

**Idea:** (heuristically) validate $\emptyset \in BCP(\psi, x)$ on *original* PCNF.

- Try to derive the negated assumption $\{\neg x\}$ by Q-resolution.
- Resolution candidates are selected from clauses "touched" by BCP.
- Like conflict-driven clause learning (CDCL) in search-based solvers.

# Novel Approach: BCP-Guided Q-Resolution (2/2)

## Corollary

*Given PCNF* $\psi := Q_1 S_1 \ldots Q_n S_n.\ \phi$ *and literal* $x$ *where* $v(x) \in S_i$. *If* $\emptyset \in BCP(\psi, x)$ *and* $\psi \vdash^* \{\neg x\}$ *then* $\psi \equiv \psi \wedge \{\neg x\}$.

## Example

$\psi := \exists e_1, e_2 \forall a_3 \exists e_4, e_5.\ \{a_3, e_5\}, \{\neg e_2, e_4\}, \{\neg e_1, e_4\}, \{e_1, e_2, \neg e_5\}$.
With assumption $\neg e_4$ we get $\emptyset \in BCP(\psi, \neg e_4)$ since $\{\neg e_1\}$, $\{\neg e_2\}$ and $\{\neg e_5\}$ become unit. Finally $\{a_3, e_5\}$ is empty by UR.
The negated assumption $\{e_4\}$ is then derived by resolving clauses in reverse-chronological order as they were affected by assignments:
$(\{a_3, e_5\}, \{e_1, e_2, \neg e_5\}) \vdash \{e_1, e_2\}$, $(\{e_1, e_2\}, \{\neg e_2, e_4\}) \vdash \{e_1, e_4\}$, $(\{e_1, e_4\}, \{\neg e_1, e_4\}) \vdash \{e_4\}$.

**Practical Application:**

- Advantage: original prefix allows full propagation power in BCP.
- BCP-based selection of resolution candidates is only a heuristic.

# Comparing FL Approaches

## Proposition

*Abstraction-based FL and BCP-guided Q-resolution are orthogonal to each other with respect to detecting necessary assignments.*

**Consequences:**

- There are PCNFs where one approach can detect a necessary assignment the other one cannot.
- No approach can detect all necessary assignments.
- Crucial observation: Q-resolution for CDCL is not optimal (see below)!
- (How) Can we apply quantifier abstraction for clause learning?

## Example

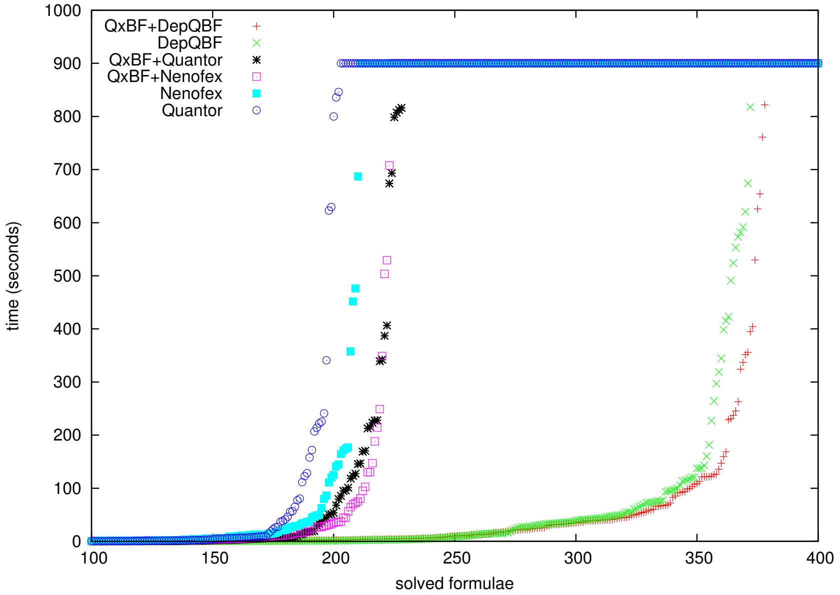
$\psi := \forall a_1 \exists e_2, e_3 \forall a_4 \exists e_5.\ \{a_1, e_2\}, \{\neg a_1, e_3\}, \{e_3, \neg e_5\}, \{a_1, e_2, \neg e_3\}$, $\{\neg e_2, a_4, e_5\}$. We have $\emptyset \in BCP(Abs(\psi, 2), \neg e_3)$ but $\psi \not\vdash^* \{e_3\}$: assignment $\{e_3\} \mapsto$ *true* is necessary but Q-resolution can *not* derive clause $\{e_3\}$.

**Tool "QxBF":** FL-based preprocessor operating in rounds.
**SAT-Based FL:** using SAT solver to detect necessary assignments.

| *QBFEVAL'10: 568 formulae* | | | | | |
|---|---|---|---|---|---|
| *Preprocessing* | *Solver* | *Solved* | *Time (Preproc.)* | *SAT* | *UNSAT* |
| SAT | DepQBF | 379 | 322.31 (7.17) | 167 | 212 |
| QRES+SAT | | 378 | 322.83 (6.22) | 167 | 211 |
| ABS+SAT | | 378 | 323.19 (7.21) | 167 | 211 |
| ABS | | 375 | 327.64 (3.33) | 168 | 207 |
| QRES | | 374 | 327.63 (1.83) | 167 | 207 |
| None | | 372 | 334.60 (—) | 166 | 206 |
| ABS+SAT | Quantor | 229 | 553.65 (7.21) | 112 | 117 |
| | Nenofex | 224 | 553.37 (7.21) | 104 | 120 |
| none | | 211 | 573.65 (—) | 103 | 108 |
| | Quantor | 203 | 590.15 (—) | 99 | 104 |
| ABS+SAT | squolem | 154 | 658.28 (7.21) | 63 | 91 |
| None | | 124 | 708.80 (—) | 53 | 71 |

Table: Solver performance with(out) time-limited failed literal preprocessing. Search-based solver DepQBF, elimination-based solvers Quantor, squolem, Nenofex. No preprocessing ("none"), SAT-based FL ("SAT"), abstraction-based FL ("ABS") and BCP-guided Q-resolution ("QRES").

# FL Times Plot



*Part 3: Quantified Blocked Clause Elimination (QBCE)*

**Blocked Clause Elimination (BCE) for SAT** [JBH10]

- Allows CNF-level simplifications after circuit-to-CNF transformation.
- At least as effective as many circuit-level preprocessing techniques.
- Simulates pure literal rule, Plaisted-Greenbaum encoding, ...

**Quantified Blocked Clause Elimination (QBCE) for QBF**

- Paper submitted to CADE'11 (joint work with Armin Biere, Martina Seidl).
- Generalizes BCE to QBF: minor but crucial adaption of BCE definition.
- Implementation: tool "bloqqer" combines QBCE and extensions with variable elimination, self-subsuming resolution, subsumption,...

**Definition of QBCE:** based checking possible Q-resolvents.

**Definition**

Let $C_1, C_2$ be clauses with $v \in C_1, \neg v \in C_2$ and $q(v) = \exists$.

1. Tentative resolvent: $C_1 \otimes C_2 := (UR(C_1) \cup UR(C_2)) \setminus \{v, \neg v\}$.

**Blocked Clause Elimination (BCE) for SAT** [JBH10]

- Allows CNF-level simplifications after circuit-to-CNF transformation.
- At least as effective as many circuit-level preprocessing techniques.
- Simulates pure literal rule, Plaisted-Greenbaum encoding, . . .

**Quantified Blocked Clause Elimination (QBCE) for QBF**

- Paper submitted to CADE'11 (joint work with Armin Biere, Martina Seidl).
- Generalizes BCE to QBF: minor but crucial adaption of BCE definition.
- Implementation: tool "bloqqer" combines QBCE and extensions with variable elimination, self-subsuming resolution, subsumption,. . .

**Definition of QBCE:** based checking possible Q-resolvents.

### Definition

Let $C_1, C_2$ be clauses with $v \in C_1, \neg v \in C_2$ and $q(v) = \exists$.

1. Tentative resolvent: $C_1 \otimes C_2 := (UR(C_1) \cup UR(C_2)) \setminus \{v, \neg v\}$.

## Definition (Quantified Blocking Literal)

Given PCNF $\psi := Q_1 S_1 \dots Q_n S_n.\ \phi$, a literal l in a clause $C \in \psi$ is called *quantified blocking literal* if for every clause $C'$ with $\neg l \in C'$, there exists a literal $k$ such that $\{k, \neg k\} \subseteq C \otimes C'$ with $k \leq l$.

## Definition (Quantified Blocked Clause)

Given PCNF $Q_1 S_1 \dots Q_n S_n.\ (\phi \wedge C)$. Clause $C$ is *quantified blocked* if it contains a quantified blocking literal.

Then $Q_1 S_1 \dots Q_n S_n.\ (\phi \wedge C) \overset{sat}{\equiv} Q_1 S_1 \dots Q_n S_n.\ \phi$.

| $C_1 \in Occs(l)$ blocked? | $C_2 \in Occs(\neg l)$ | $C_1 \otimes C_2$ |
|---|---|---|
| $(x_1 \vee x_2 \vee \dots \vee x_n \vee \dots \vee l \vee \dots)$ | $(\dots \neg x_1 \vee \dots \vee \neg l \vee \dots)$ | $\{x_1, \neg x_1\} \in C_1 \otimes C_2$ |
| | $(\dots \neg x_2 \vee \dots \vee \neg l \vee \dots)$ | $\{x_2, \neg x_2\} \in C_1 \otimes C_2$ |
| | $\dots$ | |
| | $(\dots \neg x_n \vee \dots \vee \neg l \vee \dots)$ | $\{x_n, \neg x_n\} \in C_1 \otimes C_2$ |

## Example

All clauses blocked: $\forall x \exists y((x \vee \neg y) \wedge (\neg x \vee y))$.
No clause blocked: $\exists x \forall y((x \vee \neg y) \wedge (\neg x \vee y))$.

# QBCE Definition

## Definition (Quantified Blocking Literal)

Given PCNF $\psi := Q_1 S_1 \ldots Q_n S_n.\ \phi$, a literal l in a clause $C \in \psi$ is called *quantified blocking literal* if for every clause $C'$ with $\neg l \in C'$, there exists a literal $k$ such that $\{k, \neg k\} \subseteq C \otimes C'$ with $k \leq l$.

## Definition (Quantified Blocked Clause)

Given PCNF $Q_1 S_1 \ldots Q_n S_n.\ (\phi \wedge C)$. Clause $C$ is *quantified blocked* if it contains a quantified blocking literal.

Then $Q_1 S_1 \ldots Q_n S_n.\ (\phi \wedge C) \overset{sat}{\equiv} Q_1 S_1 \ldots Q_n S_n.\ \phi$.

| $C_1 \in Occs(l)$ blocked? | $C_2 \in Occs(\neg l)$ | $C_1 \otimes C_2$ |
|---|---|---|
| $(x_1 \vee x_2 \vee \ldots \vee x_n \vee \ldots \vee l \vee \ldots)$ | $(\ldots \neg x_1 \vee \ldots \vee \neg l \vee \ldots)$ | $\{x_1, \neg x_1\} \in C_1 \otimes C_2$ |
| | $(\ldots \neg x_2 \vee \ldots \vee \neg l \vee \ldots)$ | $\{x_2, \neg x_2\} \in C_1 \otimes C_2$ |
| | $\ldots$ | |
| | $(\ldots \neg x_n \vee \ldots \vee \neg l \vee \ldots)$ | $\{x_n, \neg x_n\} \in C_1 \otimes C_2$ |

## Example

All clauses blocked: $\forall x \exists y((x \vee \neg y) \wedge (\neg x \vee y))$.
No clause blocked: $\exists x \forall y((x \vee \neg y) \wedge (\neg x \vee y))$.

# QBCE Definition

## Definition (Quantified Blocking Literal)

Given PCNF $\psi := Q_1 S_1 \ldots Q_n S_n.\ \phi$, a literal l in a clause $C \in \psi$ is called *quantified blocking literal* if for every clause $C'$ with $\neg l \in C'$, there exists a literal $k$ such that $\{k, \neg k\} \subseteq C \otimes C'$ with $k \leq l$.

## Definition (Quantified Blocked Clause)

Given PCNF $Q_1 S_1 \ldots Q_n S_n.\ (\phi \wedge C)$. Clause $C$ is *quantified blocked* if it contains a quantified blocking literal.

Then $Q_1 S_1 \ldots Q_n S_n.\ (\phi \wedge C) \overset{sat}{\equiv} Q_1 S_1 \ldots Q_n S_n.\ \phi$.

| $C_1 \in Occs(l)$ blocked? | $C_2 \in Occs(\neg l)$ | $C_1 \otimes C_2$ |
|---|---|---|
| $(x_1 \vee x_2 \vee \ldots \vee x_n \vee \ldots \vee l \vee \ldots)$ | $(\ldots \neg x_1 \vee \ldots \vee \neg l \vee \ldots)$ | $\{x_1, \neg x_1\} \in C_1 \otimes C_2$ |
| | $(\ldots \neg x_2 \vee \ldots \vee \neg l \vee \ldots)$ | $\{x_2, \neg x_2\} \in C_1 \otimes C_2$ |
| | $\ldots$ | |
| | $(\ldots \neg x_n \vee \ldots \vee \neg l \vee \ldots)$ | $\{x_n, \neg x_n\} \in C_1 \otimes C_2$ |

## Example

All clauses blocked: $\forall x \exists y ((x \vee \neg y) \wedge (\neg x \vee y))$.
No clause blocked: $\exists x \forall y ((x \vee \neg y) \wedge (\neg x \vee y))$.

# QBCE Definition

## Definition (Quantified Blocking Literal)

Given PCNF $\psi := Q_1 S_1 \ldots Q_n S_n.\ \phi$, a literal l in a clause $C \in \psi$ is called *quantified blocking literal* if for every clause $C'$ with $\neg l \in C'$, there exists a literal $k$ such that $\{k, \neg k\} \subseteq C \otimes C'$ with $k \leq l$.

## Definition (Quantified Blocked Clause)

Given PCNF $Q_1 S_1 \ldots Q_n S_n.\ (\phi \wedge C)$. Clause $C$ is *quantified blocked* if it contains a quantified blocking literal.

Then $Q_1 S_1 \ldots Q_n S_n.\ (\phi \wedge C) \overset{sat}{\equiv} Q_1 S_1 \ldots Q_n S_n.\ \phi$.

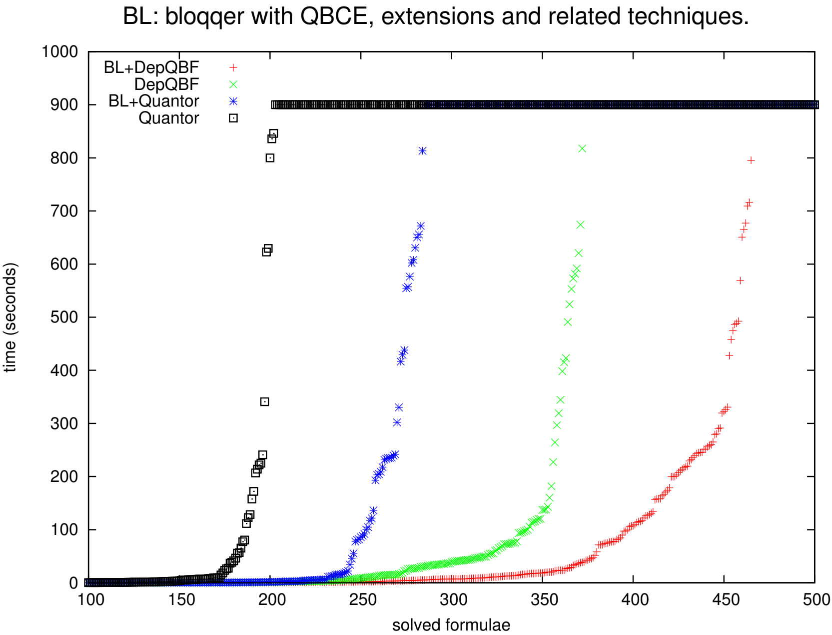
| $C_1 \in Occs(l)$ blocked? | $C_2 \in Occs(\neg l)$ | $C_1 \otimes C_2$ |
|---|---|---|
| $(x_1 \vee x_2 \vee \ldots \vee x_n \vee \ldots \vee l \vee \ldots)$ | $(\ldots \neg x_1 \vee \ldots \vee \neg l \vee \ldots)$ | $\{x_1, \neg x_1\} \in C_1 \otimes C_2$ |
| | $(\ldots \neg x_2 \vee \ldots \vee \neg l \vee \ldots)$ | $\{x_2, \neg x_2\} \in C_1 \otimes C_2$ |
| | $\ldots$ | |
| | $(\ldots \neg x_n \vee \ldots \vee \neg l \vee \ldots)$ | $\{x_n, \neg x_n\} \in C_1 \otimes C_2$ |

## Example

All clauses blocked: $\forall x \exists y((x \vee \neg y) \wedge (\neg x \vee y))$.
No clause blocked: $\exists x \forall y((x \vee \neg y) \wedge (\neg x \vee y))$.

Table: Bloqqer (QBCE, extensions and related techniques) combined with search- (DepQBF, QuBE) and elimination-based (Nenofex, Quantor) solvers.

| | | # formulas | | | runtime (sec) | | |
|---|---|---|---|---|---|---|---|
| | preprocessor | SOLVED | SAT | UNSAT | $\sum$ ($10^3$) | AVG | MEDIAN |
| DepQBF | bloqqer | 467 | 224 | 243 | 112 | 198 | 5 |
| | no preprocessing | 373 | 167 | 206 | 189 | 332 | 26 |
| QuBE | bloqqer | 444 | 200 | 244 | 139 | 246 | 5 |
| | no preprocessing | 332 | 135 | 197 | 242 | 426 | 258 |
| Quantor | bloqqer | 288 | 145 | 143 | 266 | 468 | 34 |
| | no preprocessing | 206 | 100 | 106 | 333 | 587 | 38 |
| Nenofex | bloqqer | 268 | 132 | 136 | 276 | 487 | 23 |
| | no preprocessing | 221 | 107 | 114 | 319 | 561 | 113 |


BL: bloqqer with QBCE, extensions and related techniques.
BL+DepQBF
DepQBF
BL+Quantor
Quantor
time (seconds)
solved formulae
0
100
200
300
400
500
600
700
800
900
1000
100
150
200
250
300
350
400
450
500

**Preprocessing QBF:**

- Positive effects on elimination- and search-based QBF solvers.

**Failed Literal Detection (FL):**

- Detecting a subset of necessary assignments.
- Exponential reduction of search-space.
- Soundness by abstraction and Q-resolution.
- Orthogonality: current CDCL approaches in QBF are not optimal.

**Quantified Blocked Clause Elimination (QBCE):**

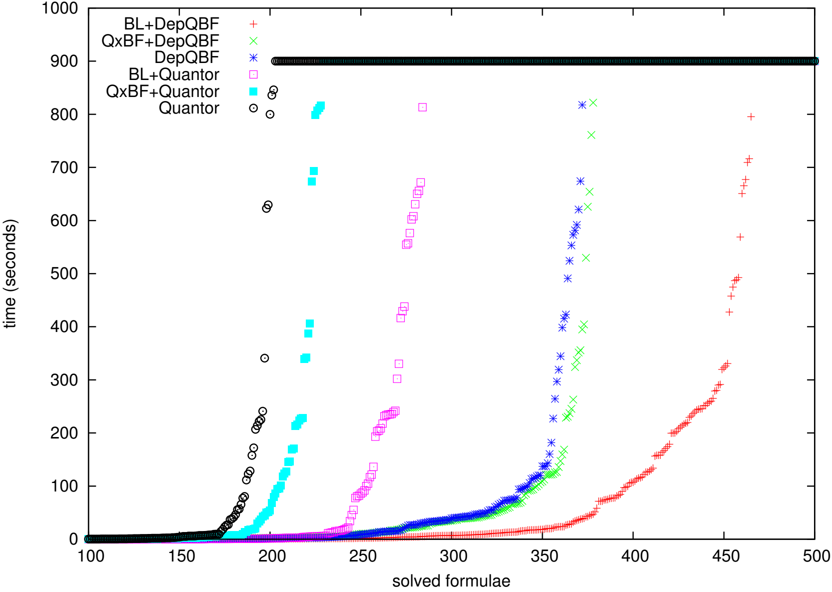
- Generalizes BCE for SAT to QBF.
- Best performance when combined with variable elimination,. . .

**Work in Progress:**

- Papers submitted to SAT'11 (FL) and CADE'11 (QBCE).
- Source code of our preprocessors will be published.
- Dynamic applications of FL and QBCE.

.

# QxBF (FL) and bloqqer (QBCE)



# *References*

U. Bubeck and H. Kleine Büning.
Bounded Universal Expansion for Preprocessing QBF.
In J. Marques-Silva and K. A. Sakallah, editors, *SAT*, volume 4501 of *LNCS*, pages 244–257. Springer, 2007.

D. Le Berre.
Exploiting the Real Power of Unit Propagation Lookahead.
*Electronic Notes in Discrete Mathematics*, 9:59–80, 2001.

A. Biere.
Resolve and Expand.
In H. H. Hoos and D. G. Mitchell, editors, *SAT (Selected Papers)*, volume 3542 of *LNCS*, pages 59–70. Springer, 2004.

H. Kleine Büning, M. Karpinski, and A. Flögel.
Resolution for Quantified Boolean Formulas.
*Inf. Comput.*, 117(1):12–18, 1995.

F. Bacchus and J. Winter.
Effective Preprocessing with Hyper-Resolution and Equality Reduction.
In E. Giunchiglia and A. Tacchella, editors, *SAT*, volume 2919 of *LNCS*, pages 341–355. Springer, 2003.

E. Giunchiglia, P. Marin, and Massimo Narizzano.

sQueezeBF: An Effective Preprocessor for QBFs Based on Equivalence Reasoning.
In O. Strichman and S. Szeider, editors, *SAT*, volume 6175 of *LNCS*, pages 85–98. Springer, 2010.

M. Järvisalo, A. Biere, and M. Heule.
Blocked Clause Elimination.
In J. Esparza and R. Majumdar, editors, *TACAS*, volume 6015 of *LNCS*, pages 129–144. Springer, 2010.

I. Lynce and J. P. Marques Silva.
Probing-Based Preprocessing Techniques for Propositional Satisfiability.
In *ICTAI*, pages 105–. IEEE Computer Society, 2003.

F. Pigorsch and C. Scholl.
An AIG-Based QBF-Solver Using SAT for Preprocessing.
In S. S. Sapatnekar, editor, *DAC*, pages 170–175. ACM, 2010.

J. Rintanen.
Improvements to the Evaluation of Quantified Boolean Formulae.
In T. Dean, editor, *IJCAI*, pages 1192–1197. Morgan Kaufmann, 1999.

H. Samulowitz and F. Bacchus.
Using SAT in QBF.
In P. van Beek, editor, *CP*, volume 3709 of *LNCS*, pages 578–592. Springer, 2005.

H. Samulowitz and F. Bacchus.
Binary Clause Reasoning in QBF.
In A. Biere and C. P. Gomes, editors, *SAT*, volume 4121 of *LNCS*, pages 353–367. Springer, 2006.

H. Samulowitz, J. Davies, and F. Bacchus.
Preprocessing QBF.
In F. Benhamou, editor, *CP*, volume 4204 of *LNCS*, pages 514–529. Springer, 2006.