JKU
JOHANNES KEPLER
UNIVERSITY LINZ

Recent Trends in Solving Quantified Boolean Formulas

Martina Seidl

Institute for Formal Models and Verification
Johannes Kepler University Linz

What are QBF?

- **Quantified Boolean formulas (QBF)** are

  formulas of propositional logic + quantifiers

- Examples for QBFs:
  - $\exists x \exists y.((x \vee \neg y) \wedge (\neg x \vee y))$

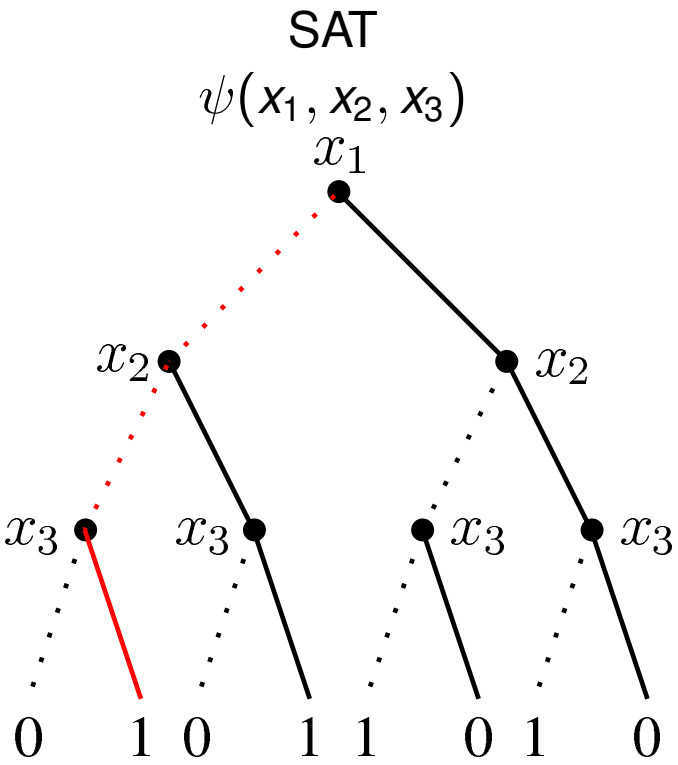
    Satisfiability checking in propositional logic
  - $\forall x \forall y.((x \vee \neg y) \wedge (\neg x \vee y))$

    Validity checking in propositional logic
  - $\exists x \forall y.((x \vee \neg y) \wedge (\neg x \vee y))$

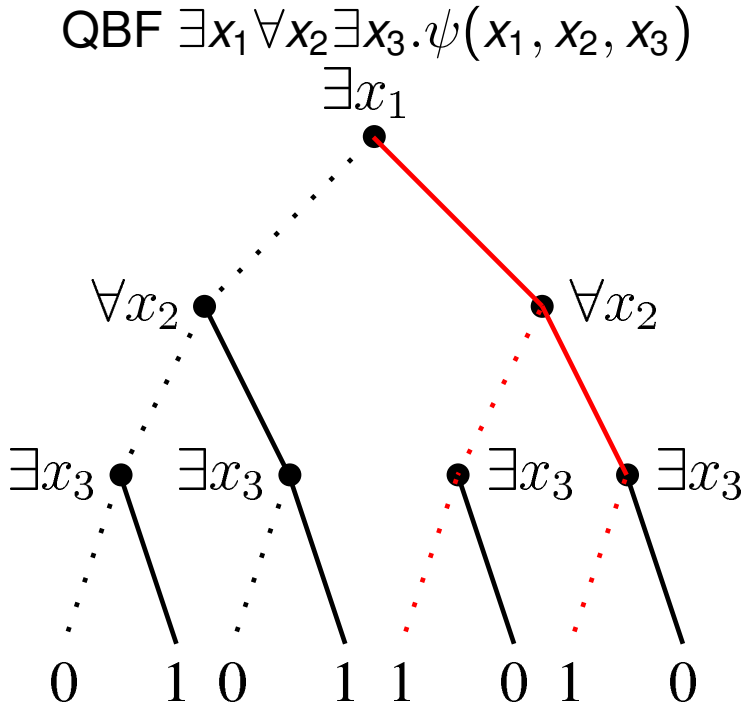
    Is there a value for x such that for all values of y the formula is true?
  - $\forall y \exists x.((x \vee \neg y) \wedge (\neg x \vee y))$

    For all values of y, is there a value for x such that the formula is true?

SAT vs. QSAT aka NP vs. PSPACE



Is there a satisfying assignment?

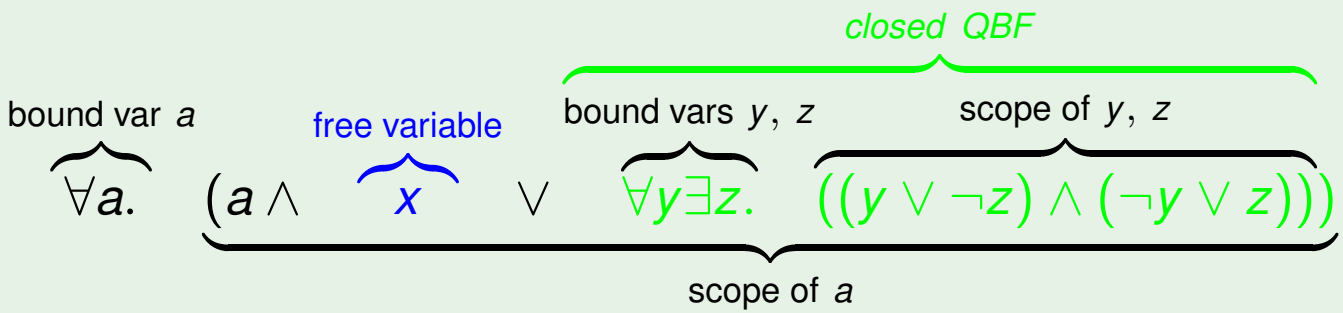


Is there a satisfying assignment **tree**?

Some Notions

- Let $Qv.\psi$ with $Q \in \{\forall, \exists\}$ be a subformula in a QBF ϕ. Then
  - ψ is the *scope* of v
  - Q is the *quantifier binding* of v
  - *quant(v) = Q*
- *free variable* w in ϕ: w has no quantifier binding in ϕ
- *bound variable* w in QBF ϕ: w has quantifier binding in ϕ
- *closed QBF*: no free variables

Example



QBF Semantics

Boolean split (QBF ϕ)

switch(ϕ)
  case $\top$: return **true**;
  case $\bot$: return **false**;
  case $\neg\psi$: return (**not** split(ψ));
  case $\psi' \wedge \psi''$: return split(ψ') && split(ψ'');
  case $\psi' \vee \psi''$: return split(ψ') || split(ψ'');
  case $QX.\psi$:
    select $x \in X$; $X' = X \backslash \{x\}$;
    if ($Q == \forall$)
      return (split($QX'\psi[x/\top]$) &&
              split($QX'\psi[x/\bot]$));
    else
      return (split($QX'\psi[x/\top]$) ||
              split($QX'\psi[x/\bot]$));

Prenex Conjunctive Normal Form (PCNF)

A QBF ϕ is in **prenex conjunctive normal form** iff

- ϕ is in *prenex normal form* $\phi = Q_1 v_1 \ldots Q_n v_n . \psi$
- matrix ψ is in *conjunctive normal form*, i.e.,

$$\psi = C_1 \wedge \cdots \wedge C_n$$

where C_i are clauses, i.e., disjunctions of literals.

Example

$$\underbrace{\forall x \exists y .}_{\text{prefix}} \underbrace{((x \vee \neg y) \wedge (\neg x \vee y))}_{\text{matrix in CNF}}$$

Universal Reduction

- Let ϕ be a QBF in PCNF and $C \in \phi$.
- Let $l \in C$ with
  - quant(l) $= \forall$
  - forall $k \in C$ with quant(k) $= \exists$, $k < l$, i.e., all existential variables k of C are to the left of l in the prefix.
- Then l may be removed from C.
- We write $UR(C)$ for the *universal reduct* of clause C.

Example

$\forall ab \exists x \forall c \exists yz \forall d.\{\{a, b, \neg c, x\}, \{a, \neg b, x\}, \{c, y, d\}, \{x, y\}, \{x, c, d\}\}\}$

After Universal Reduction:

$\forall ab \exists x \forall c \exists yz \forall d.\{\{a, b, x\}, \{a, \neg b, x\}, \{c, y\}, \{x, y\}, \{x\}\}\}$

Resolution for QBF

Q-Resolution: propositional resolution + universal reduction (UR).

Definition (Q-Resolution [KleineBüningKF95])

Let C_1, C_2 be clauses with existential literal $v \in C_1$ and $\neg v \in C_2$.

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

- Q-resolution is a sound and complete calculus.
- Dual variant for QBFs in QDNF.
- Universals as pivot are also possible.

Q-Resolution Example

Exclusive OR (XOR): QBF $\psi = \exists x \forall y (x \vee y) \wedge (\neg x \vee \neg y)$

Truth Table

x	y	ψ
0	0	0
0	1	1
1	0	1
1	1	0

Universal-Reduction $\longrightarrow$ $x \vee y$ $\quad \neg x \vee \neg y$

Resolution $\longrightarrow$ x $\quad \neg x$

unsat

$\emptyset$

Q-Resolution Proof

$\longrightarrow \quad y = x \quad \Rightarrow \quad \psi = 0$

$\longrightarrow \quad f_y(x) = x \quad$ (counter model)

Q-Resolution Example

Exclusive OR (XOR): QBF $\psi = \exists x \forall y (x \vee y) \wedge (\neg x \vee \neg y)$

Truth Table

x	y	ψ
0	0	0
0	1	1
1	0	1
1	1	0

unsat

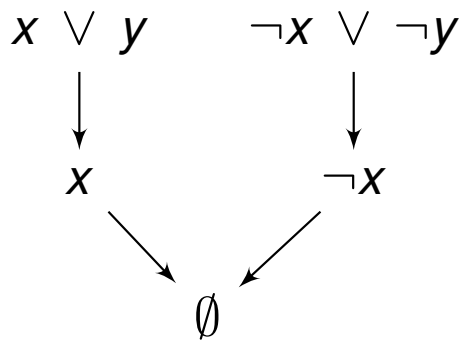
$\longrightarrow \quad y = x \quad \Rightarrow \quad \psi = 0$

$\longrightarrow \quad f_y(x) = x \quad$ (counter model)

Q-Resolution Example

Exclusive OR (XOR): QBF $\psi = \exists x \forall y (x \vee y) \wedge (\neg x \vee \neg y)$

Q-Resolution Proof



$\longrightarrow \quad y = x \quad \Rightarrow \quad \psi = 0$

$\longrightarrow \quad f_y(x) = x \quad$ (counter model)

Q-Resolution Example

Exclusive OR (XOR): QBF $\psi = \exists x \forall y (x \vee y) \wedge (\neg x \vee \neg y)$

Q-Resolution Proof

Universal-Reduction $\longrightarrow$ $x \vee y$ $\quad$ $\neg x \vee \neg y$

x $\quad$ $\neg x$

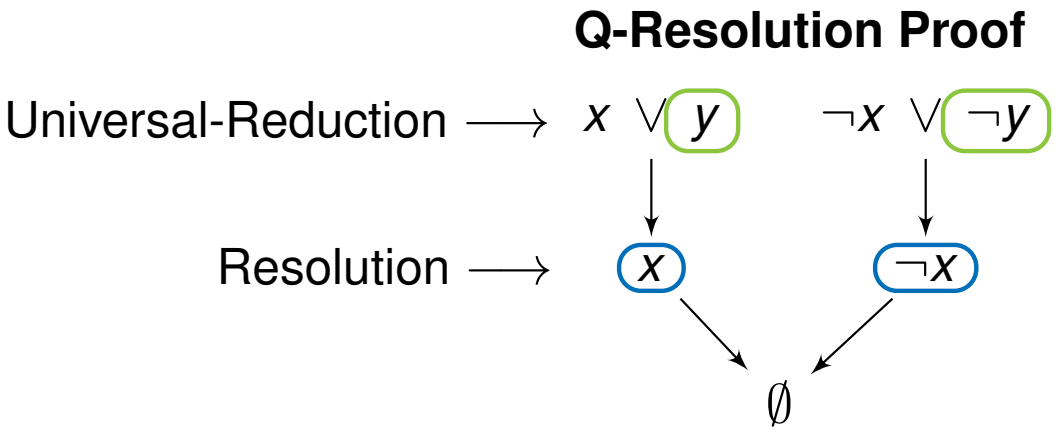
$\emptyset$

$\longrightarrow \quad y = x \quad \Rightarrow \quad \psi = 0$

$\longrightarrow \quad f_y(x) = x \quad$ (counter model)

Q-Resolution Example

Exclusive OR (XOR): QBF $\psi = \exists x \forall y (x \vee y) \wedge (\neg x \vee \neg y)$



$\longrightarrow \quad y = x \quad \Rightarrow \quad \psi = 0$

$\longrightarrow \quad f_y(x) = x \quad$ (counter model)

Q-Resolution Example

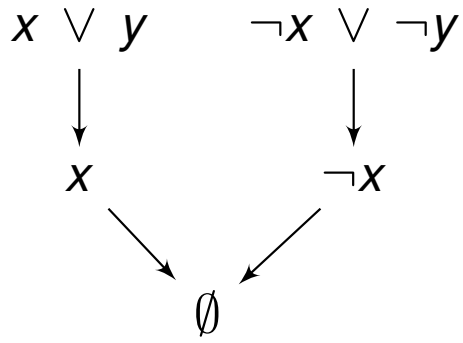
Exclusive OR (XOR): QBF $\psi = \exists x \forall y (x \vee y) \wedge (\neg x \vee \neg y)$

Truth Table

x	y	ψ
0	0	0
0	1	1
1	0	1
1	1	0

unsat

Q-Resolution Proof



$\longrightarrow \quad y = x \quad \Rightarrow \quad \psi = 0$

$\longrightarrow \quad f_y(x) = x \quad$ (counter model)

Q-Resolution Example

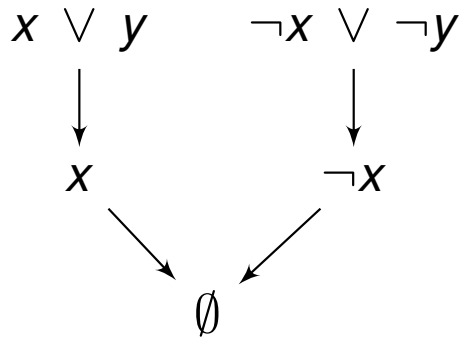
Exclusive OR (XOR): QBF $\psi = \exists x \forall y (x \vee y) \wedge (\neg x \vee \neg y)$

Truth Table

x	y	ψ
0	0	0
0	1	1
1	0	1
1	1	0

unsat

Q-Resolution Proof



$\longrightarrow \quad y = x \quad \Rightarrow \quad \psi = 0$

$\longrightarrow \quad f_y(x) = x \quad$ (counter model)

Unit Clauses

A clause C is called **unit** in a QBF ϕ iff

- C contains exactly one existential literal
- the universal literals of C are to the right of the existential literal in the prefix

The existential literal in the unit clause is called *unit literal*.

Example

$\forall ab \exists x \forall c \exists y \forall d . \{\{a, b, \neg c, \neg x\}, \{a, \neg b\}, \{c, y, d\}, \{x, y\}, \{x, c, d\}, \{y\}\}$

Unit literals: x, y

Pure Literals

A literal l is called **pure** in a QBF ϕ iff

- l occurs in ϕ
- the complement of l, i.e., $\bar{l}$ does not occur in ϕ

Example

$\forall ab \exists x \forall c \exists yz \forall d . \{\{a, b, \neg c\}, \{a, \neg b\}, \{c, \neg y, d\}, \{x, \neg y\}, \{x, c, d\}\}$

Pure: a, d, x, y

Important: Existential pure literals remove clauses, universal pure literals remove literals.

Transformation to PCNF

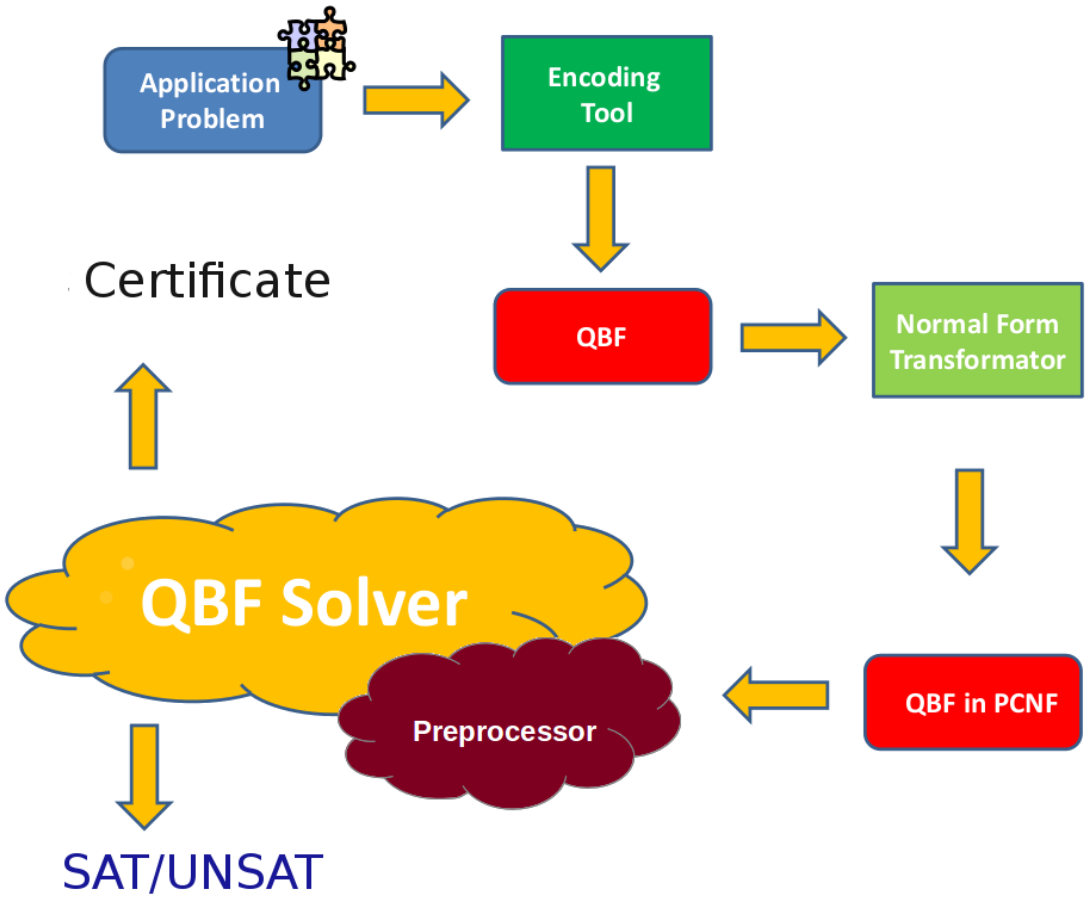
1. **Removal of Implications and Equivalences**
   - $\phi \rightarrow \psi$ becomes $\neg\phi \vee \psi$
   - $\phi \leftrightarrow \psi$ becomes $\phi \rightarrow \psi \wedge \psi \rightarrow \phi$

2. **Transformation to Negation Normal Form**
   - shift negations in front of variables
   - remove double negation

3. **Transformation to Prenex Normal Form**
   - shift quantifiers to the left
   - don't mess up the order, i.e., respect the dependencies

4. **Transformation to Conjunctive Normal Form**
   - application of Distributivity Law or
   - Tseitin Transformation (Plaisted-Greenbaum)

Workflow of QBF Solving



QBF Solver returns

1. yes/no
2. witnesses

Why QBFs?

- QSAT is the prototypical problem for *PSPACE*.
- In general, QBF allow more succinct encodings then SAT.

- QBFs are suitable as *host language* for the encoding of many application problems like
  - verification
  - artificial intelligence
  - knowledge representation
  - game solving
  - descriptive complexity

Ongoing QBF Research

- *Preprocessing*: [VanGelderWL12], [BiereLS12],[GiunchigliaMN10]
- *Certification*: [GoultiaevaVB11], [BalabanovJ11], [NiemetzPLSB12], [VanGelder13], [SeidlK14], [JanotaGM13]
- *Dependency Schemes*: [LonsingB10], [VanGelder11], [SlivovskyS12]
- *Duality Aware Solving*: [GoultieavaSB13], [GoultiaevaB13], [GoultiaevaB10], [KlieberSGC10]
- *Solving*: [JanotaM13], [JanotaKMC12], [EglyLW13], [LonsingEVG13]
- ...

Preprocessing

- *propositional techniques*
  - tautology elimination
  - subsumption
  - strengthening
  - asymmetric tautology elimination

- *QBF-variants of propositional techniques*
  - pure literal elimination
  - unit literal elimination
  - failed literal probing
  - equivalence substitution
  - equivalence rewritting
  - bounded variable elimination
  - blocked clause elimination (and related techniques)

- QBF specific techniques
  - universal reduction
  - universal expansion

Bounded Variable Elimination

- Application of Davis-Putnam approach on inner-most existential variables
- Add all possible resolvents of variable x to formula
  (tautologies may be omitted)
- Remove all clauses containing x and $\neg x$.

Example

$$\forall a \exists x \exists y.((x \vee y \vee a) \wedge (\neg x \vee \neg y) \wedge (\neg x \vee \neg a) \wedge (\neg x))$$

becomes

$$\forall a \exists y.(y \vee a)$$

becomes

$$\top$$

Universal Expansion

Given a QBF

$$\mathcal{Q} \forall a \exists X . \phi \wedge \psi$$

where

- $\mathcal{Q}$ is an arbitrary quantifier prefix,
- X is a set of variables,
- ϕ and ψ are formulas in CNF,
- ϕ contains only variables from $\mathcal{Q}$.

Then after expanding a we get the formula

$$\mathcal{Q} \exists X X' . \phi \wedge \psi[a/\top] \wedge \psi'[a/\bot]$$

where in ψ' each variable $x \in X$ is replaced by a variable $x' \in X'$.

QBCE Definition

Original BCE comes from SAT-domain

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 a *quantified blocking literal* if for every clause C' with $\neg l \in C'$, $C \otimes C'$ is tautologous wrt. some variable k such that $k \leq l$ in prefix ordering.

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

QBCE Definition

Original BCE comes from SAT-domain

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 a *quantified blocking literal* if for every clause C' with $\neg l \in C'$, $C \otimes C'$ is tautologous wrt. some variable k such that $k \leq l$ in prefix ordering.

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

QBCE Definition

Original BCE comes from SAT-domain

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 a *quantified blocking literal* if for every clause C' with $\neg l \in C'$, $C \otimes C'$ is tautologous wrt. some variable k such that $k \leq l$ in prefix ordering.

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

QBCE Definition

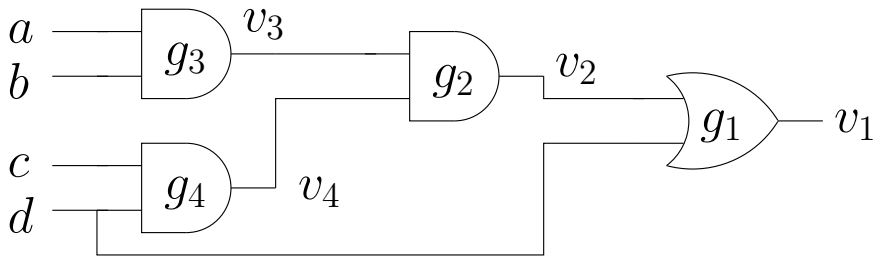
Original BCE comes from SAT-domain

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 a *quantified blocking literal* if for every clause C' with $\neg l \in C'$, $C \otimes C'$ is tautologous wrt. some variable k such that $k \leq l$ in prefix ordering.

$C_1 \in Occs(l)$ 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: QBCE Subsumes Plaisted-Greenbaum Encoding



Tseitin Encoding:

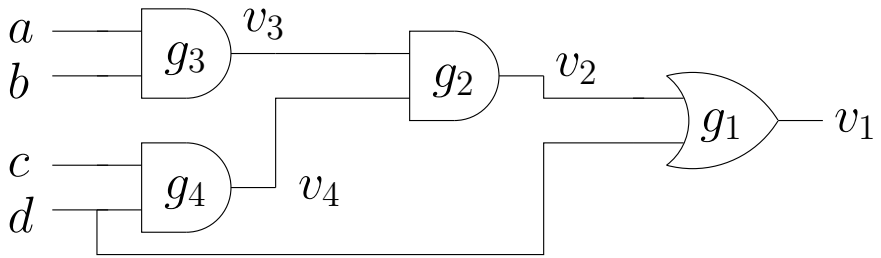
$v_1 \Leftrightarrow (v_2 \vee d)$:
$(\neg v_1 \vee v_2 \vee d) \wedge (v_1 \vee \neg v_2) \wedge (v_1 \vee \neg d)$

$v_2 \Leftrightarrow (v_3 \wedge v_4)$:
$(\neg v_2 \vee v_3) \wedge (\neg v_2 \vee v_4) \wedge (v_2 \vee \neg v_3 \vee \neg v_4)$

$v_3 \Leftrightarrow (a \wedge b)$:
$(\neg v_3 \vee a) \wedge (\neg v_3 \vee b) \wedge (v_3 \vee \neg a \vee \neg b)$

$v_4 \Leftrightarrow (c \wedge d)$:
$(\neg v_4 \vee c) \wedge (\neg v_4 \vee d) \wedge (v_4 \vee \neg c \vee \neg d)$

Example: QBCE Subsumes Plaisted-Greenbaum Encoding



Tseitin Encoding:

$v_1 \Leftrightarrow (v_2 \vee d)$:
$(\neg v_1 \vee v_2 \vee d) \wedge (v_1 \vee \neg v_2) \wedge (v_1 \vee \neg d)$

$v_2 \Leftrightarrow (v_3 \wedge v_4)$:
$(\neg v_2 \vee v_3) \wedge (\neg v_2 \vee v_4) \wedge (v_2 \vee \neg v_3 \vee \neg v_4)$

$v_3 \Leftrightarrow (a \wedge b)$:
$(\neg v_3 \vee a) \wedge (\neg v_3 \vee b) \wedge (v_3 \vee \neg a \vee \neg b)$

$v_4 \Leftrightarrow (c \wedge d)$:
$(\neg v_4 \vee c) \wedge (\neg v_4 \vee d) \wedge (v_4 \vee \neg c \vee \neg d)$

QBCE (Blocking Literals are Blue):

$v_1 \Rightarrow (v_2 \vee d)$:
$(\neg v_1 \vee v_2 \vee d)$ ~~$\wedge (v_1 \vee \neg v_2) \wedge (v_1 \vee \neg d)$~~

$v_2 \Rightarrow (v_3 \wedge v_4)$:
$(\neg v_2 \vee v_3) \wedge (\neg v_2 \vee v_4)$ ~~$\wedge (v_2 \vee \neg v_3 \vee \neg v_4)$~~

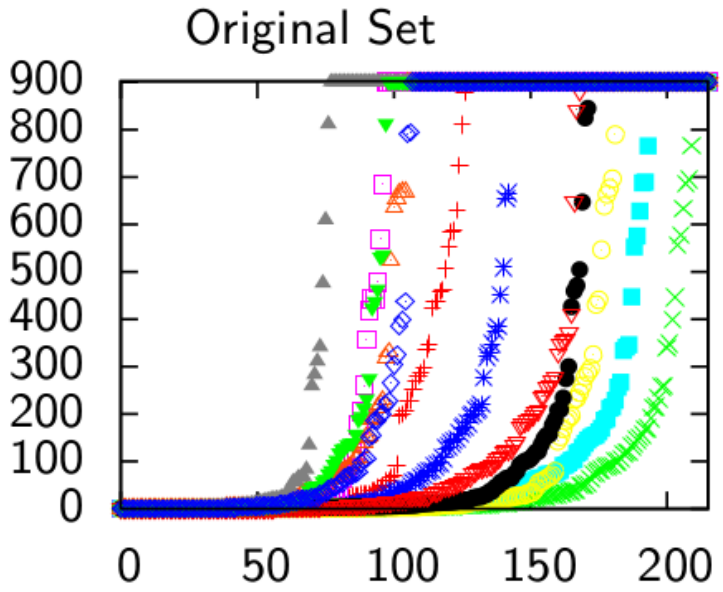
$v_3 \Rightarrow (a \wedge b)$:
$(\neg v_3 \vee a) \wedge (\neg v_3 \vee b)$ ~~$\wedge (v_3 \vee \neg a \vee \neg b)$~~

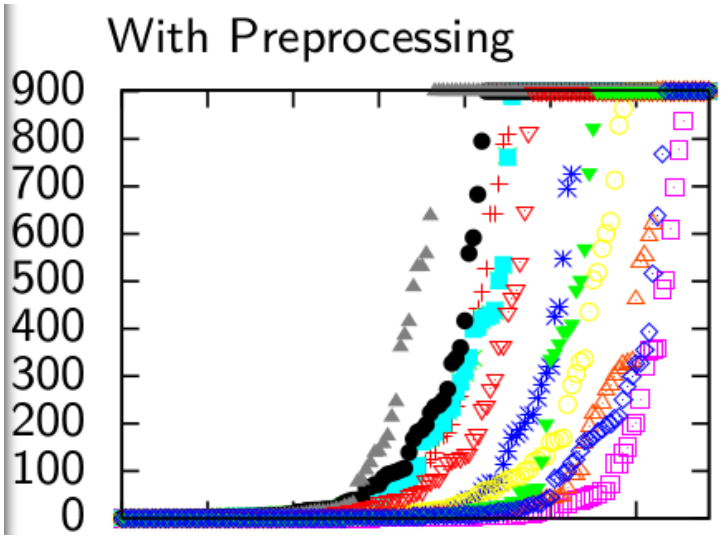
$v_4 \Rightarrow (c \wedge d)$:
$(\neg v_4 \vee c) \wedge (\neg v_4 \vee d)$ ~~$\wedge (v_4 \vee \neg c \vee \neg d)$~~

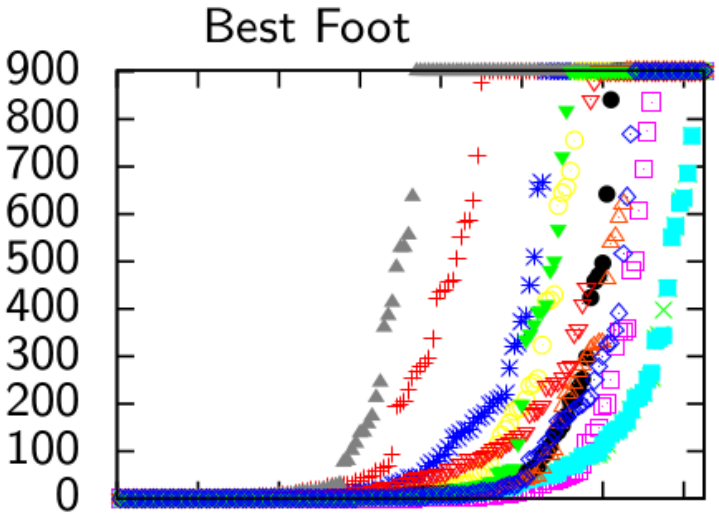
- QBCE removes clauses from direction "$\Leftarrow$" (same as PG: only "$\Rightarrow$").
- Apply Tseitin encoding to original circuit and optimize CNF by QBCE.

Impact of Preprocessing (from QBF Gallery 2013)

- Question: Is preprocessing always beneficial?
- Best foot evaluation (virtual experiment): solver chooses whether to use Bloqqer.







QBF Certificates

Goal

- Sometimes True/False is not enough as an answer
- Certificates are required ...
  - ... to verify correctness of a QBF solver's result
  - ... as concrete solutions (certificates), e.g. counter examples, strategies
    $\longrightarrow$ Skolem/Herbrand function-based certificates

QBF Certificates

- as set of Skolem/Herbrand functions (e.g. $f_y(x) = x$ in prev. example)
- representation of model/counter model
- novel approach by [GoultiaevaVB11] and [BalabanovJ11]
  $\longrightarrow$ extraction of Skolem/Herbrand functions from Q-resolution proofs

QBF Certificates

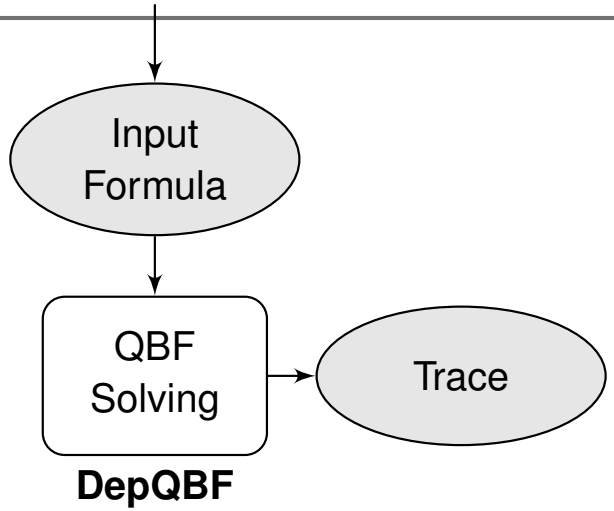
Goal

- Sometimes True/False is not enough as an answer
- Certificates are required ...
  - ... to verify correctness of a QBF solver's result
  - ... as concrete solutions (certificates), e.g. counter examples, strategies
    $\longrightarrow$ Skolem/Herbrand function-based certificates

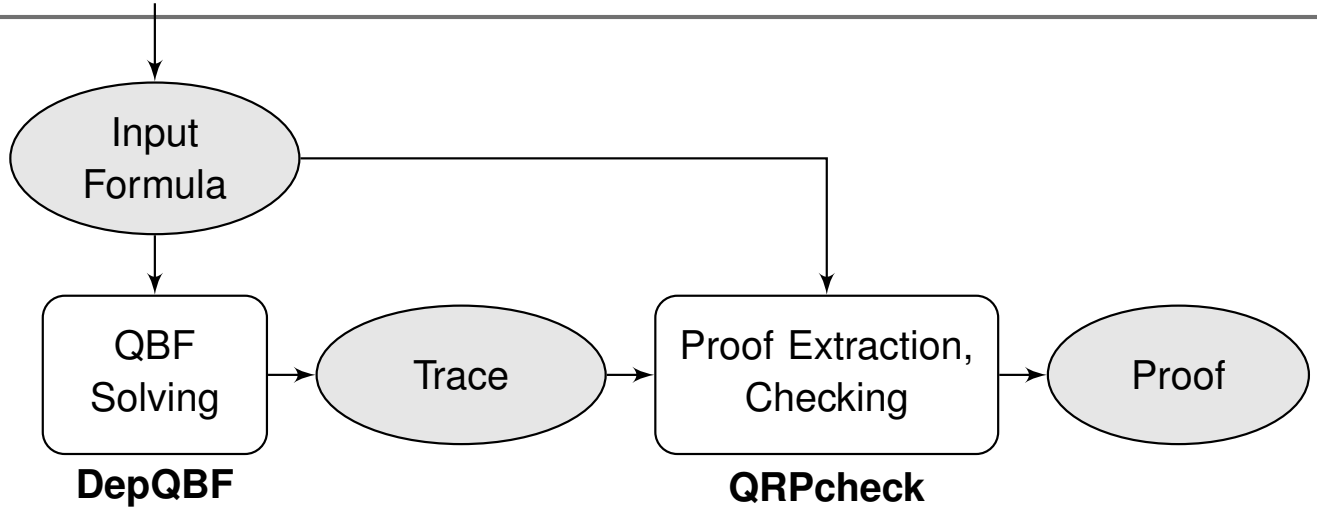
QBF Certificates

- as set of Skolem/Herbrand functions (e.g. $f_y(x) = x$ in prev. example)
- representation of model/counter model
- novel approach by [GoultiaevaVB11] and [BalabanovJ11]
  $\longrightarrow$ extraction of Skolem/Herbrand functions from Q-resolution proofs

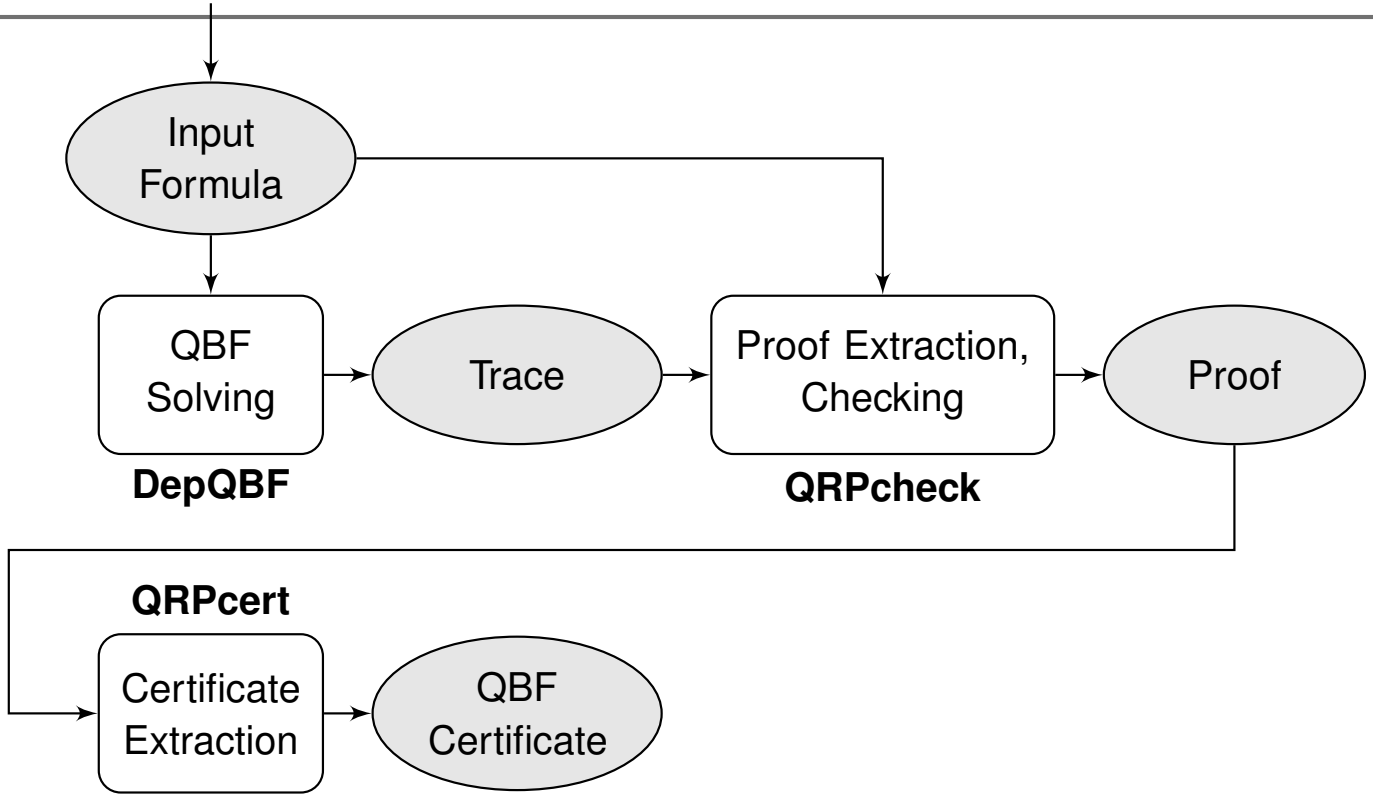
Certificaton Workflow



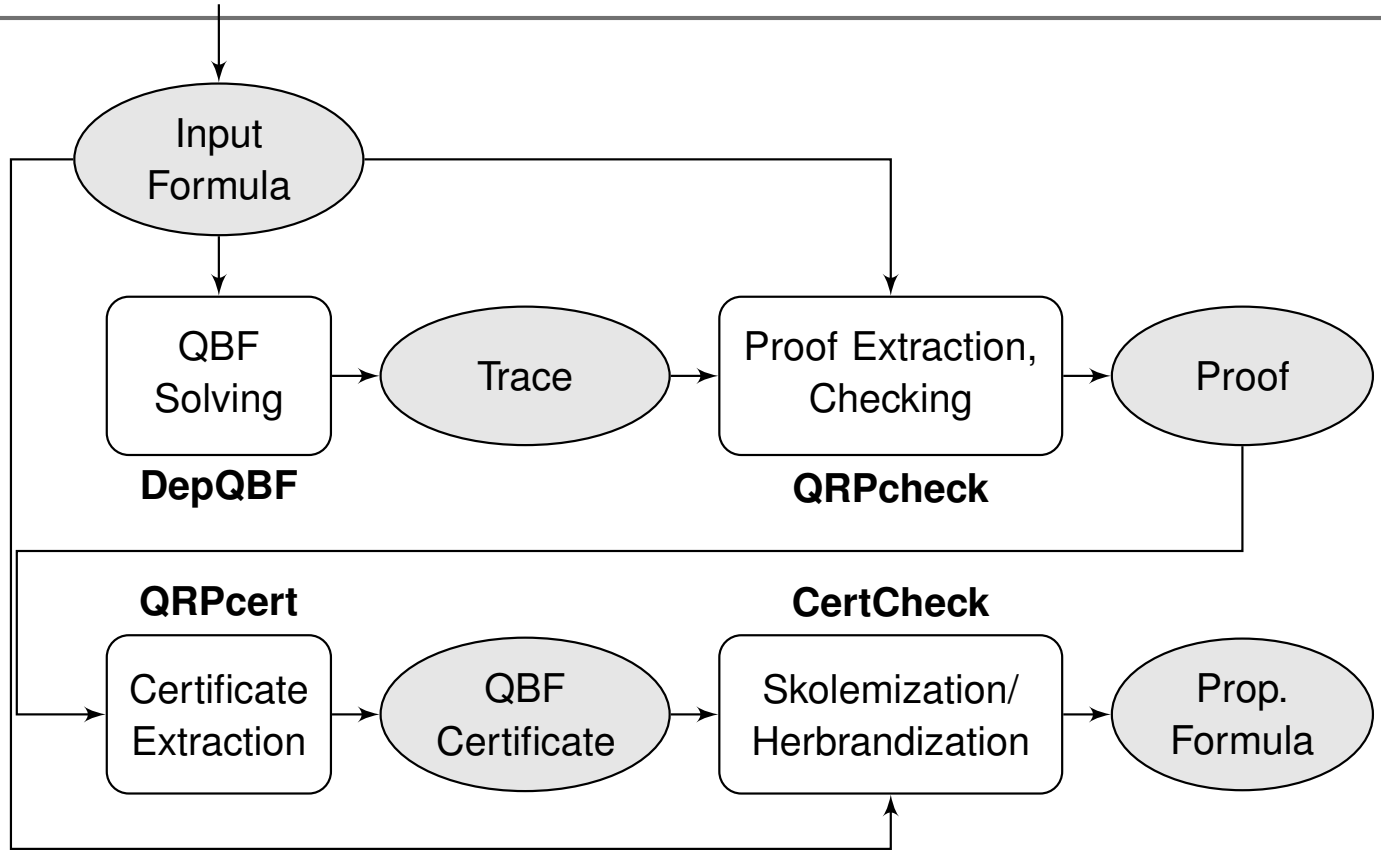
Certificaton Workflow



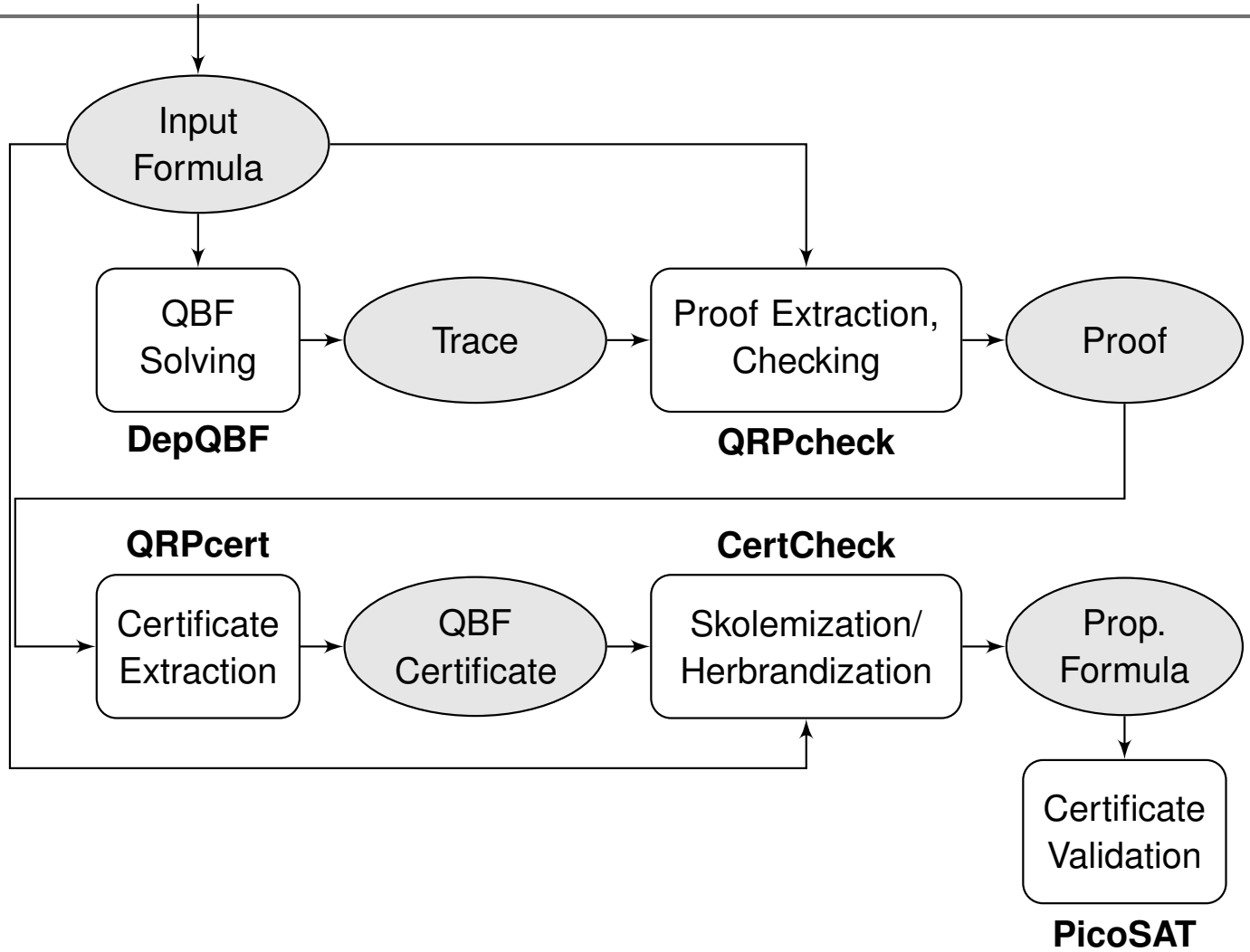
Certificaton Workflow



Certificaton Workflow



Certificaton Workflow



Certification by Example

Input Formula

$$\exists x_1 \forall y_1 \exists x_2 x_3 \forall y_2 \exists x_4 x_5 . (\neg x_1 \vee \neg x_5) \wedge (y_1 \vee x_4 \vee x_5) \wedge (x_2 \vee \neg y_2 \vee \neg x_4) \wedge (x_3 \vee \neg y_2 \vee \neg x_4) \wedge (\neg x_2 \vee \neg x_3 \vee y_2) \wedge (x_1 \vee x_4)$$

Q-Resolution Proof DAG

Extracted Herbrand Functions

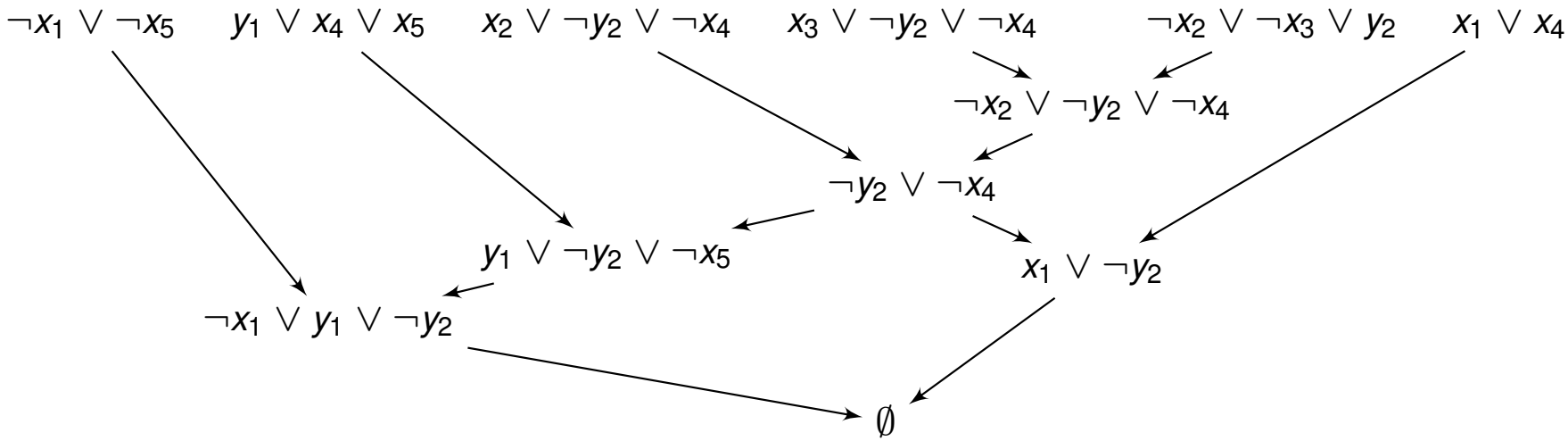
$$\left.\begin{array}{l} f_{y_1}(x_1) = \neg x_1 \\ f_{y_2}(x_1, x_2, x_3, y_1) = (\neg x_2 \vee \neg x_3) \wedge ((x_1 \wedge \neg y_1) \vee \neg x_1) \Rightarrow \\ f_{y_2}(x_2, x_3) = \neg x_2 \vee \neg x_3 \end{array}\right\} \text{Certificate}$$

Certification by Example

Input Formula

$$\exists x_1 \forall y_1 \exists x_2 x_3 \forall y_2 \exists x_4 x_5 . (\neg x_1 \vee \neg x_5) \wedge (y_1 \vee x_4 \vee x_5) \wedge (x_2 \vee \neg y_2 \vee \neg x_4) \wedge (x_3 \vee \neg y_2 \vee \neg x_4) \wedge (\neg x_2 \vee \neg x_3 \vee y_2) \wedge (x_1 \vee x_4)$$

Q-Resolution Proof DAG



Extracted Herbrand Functions

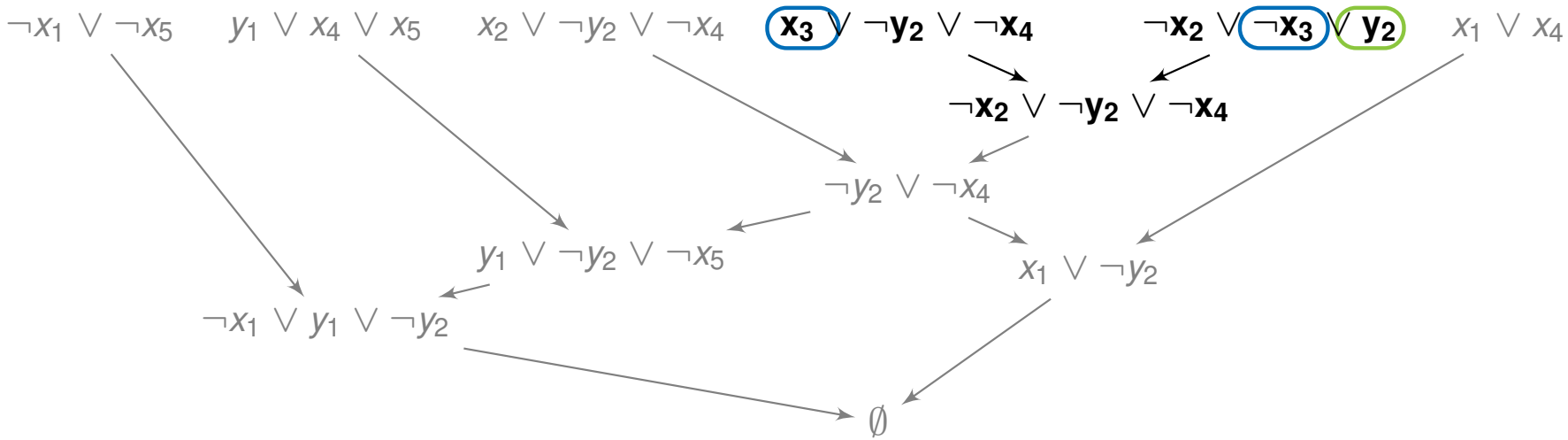
$$\left.\begin{array}{l} f_{y_1}(x_1) = \neg x_1 \\ f_{y_2}(x_1, x_2, x_3, y_1) = (\neg x_2 \vee \neg x_3) \wedge ((x_1 \wedge \neg y_1) \vee \neg x_1) \Rightarrow \\ f_{y_2}(x_2, x_3) = \neg x_2 \vee \neg x_3 \end{array}\right\} \text{Certificate}$$

Certification by Example

Input Formula

$$\exists x_1 \forall y_1 \exists x_2 x_3 \forall y_2 \exists x_4 x_5 . (\neg x_1 \vee \neg x_5) \wedge (y_1 \vee x_4 \vee x_5) \wedge (x_2 \vee \neg y_2 \vee \neg x_4) \wedge (x_3 \vee \neg y_2 \vee \neg x_4) \wedge (\neg x_2 \vee \neg x_3 \vee y_2) \wedge (x_1 \vee x_4)$$

Q-Resolution Proof DAG



Extracted Herbrand Functions

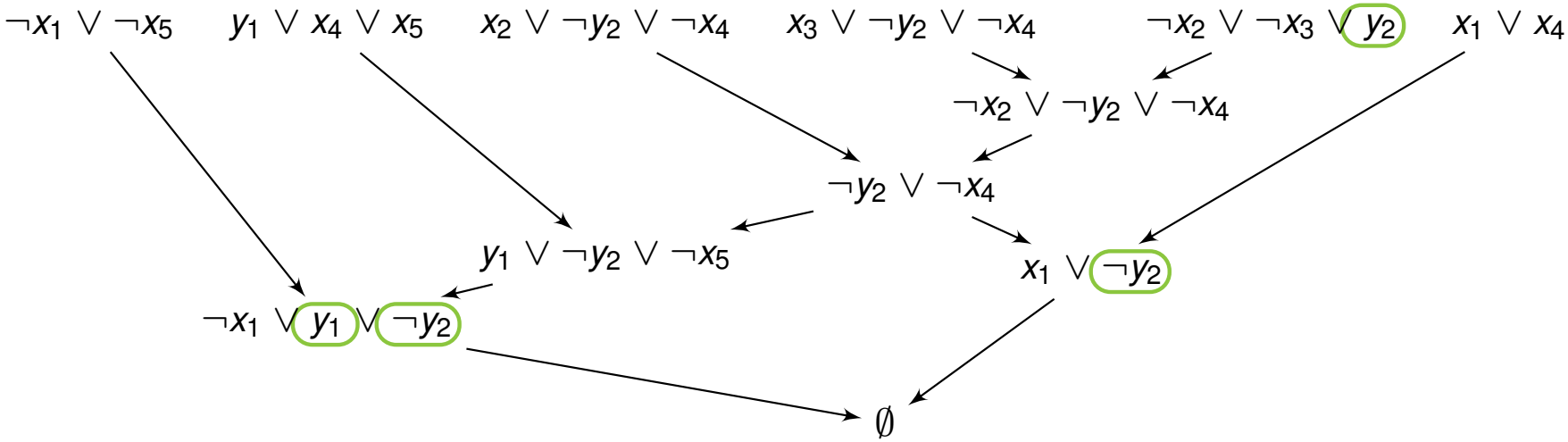
$$\left.\begin{array}{l} f_{y_1}(x_1) = \neg x_1 \\ f_{y_2}(x_1, x_2, x_3, y_1) = (\neg x_2 \vee \neg x_3) \wedge ((x_1 \wedge \neg y_1) \vee \neg x_1) \Rightarrow \\ f_{y_2}(x_2, x_3) = \neg x_2 \vee \neg x_3 \end{array}\right\} \text{Certificate}$$

Certification by Example

Input Formula

$$\exists x_1 \forall y_1 \exists x_2 x_3 \forall y_2 \exists x_4 x_5 . (\neg x_1 \vee \neg x_5) \wedge (y_1 \vee x_4 \vee x_5) \wedge (x_2 \vee \neg y_2 \vee \neg x_4) \wedge (x_3 \vee \neg y_2 \vee \neg x_4) \wedge (\neg x_2 \vee \neg x_3 \vee y_2) \wedge (x_1 \vee x_4)$$

Q-Resolution Proof DAG



Extracted Herbrand Functions

$$\left.\begin{array}{l} f_{y_1}(x_1) = \neg x_1 \\ f_{y_2}(x_1, x_2, x_3, y_1) = (\neg x_2 \vee \neg x_3) \wedge ((x_1 \wedge \neg y_1) \vee \neg x_1) \Rightarrow \\ f_{y_2}(x_2, x_3) = \neg x_2 \vee \neg x_3 \end{array}\right\} \text{Certificate}$$

Certification by Example

Certificate

Extracted Certificate: AIG Representation

$$f_{y_1}(x_1) = \neg x_1$$

$$f_{y_2}(x_2, x_3) = \neg x_2 \vee \neg x_3 = \neg(x_2 \wedge x_3)$$

Experimental Results

Benchmarks: QBFEVAL'10 set (568 formulas)
Limits: 1800 seconds and 7 GB limits

1. **Proof Extraction, Checking**
   - out of 362 solved instances, 348 proofs extracted and checked by QRPcheck
   - 14 instances lost due to memory out
2. **Certificate Extraction**
   - out of 348 proofs, 337 certificates extracted
   - 11 instances lost due to memory out
   - AND-Gates: max. 147 Mill., avg. 8 Mill., med. 369
3. **Skolemization/Herbrandization**
   - out of 337 certificates, 337 formulas skolemized/herbrandized
   - Clauses: max. 441 Mill., avg. 25 Mill., med. 71000
4. **Certificate Validation**
   - out of 337 skolemized/herbrandized formulas, 275 checked successfully
   - 45 (17) certificates not validated due to memory (time) out
     $\rightarrow$ out of these 62, 57 instances were satisfiable
   - $>$ 70% of the total runtime

Experimental Results

Benchmarks: QBFEVAL'10 set (568 formulas)
Limits: 1800 seconds and 7 GB limits

1. **Proof Extraction, Checking**
   - out of 362 solved instances, 348 proofs extracted and checked by QRPcheck
   - 14 instances lost due to memory out
2. **Certificate Extraction**
   - out of 348 proofs, 337 certificates extracted
   - 11 instances lost due to memory out
   - AND-Gates: max. 147 Mill., avg. 8 Mill., med. 369
3. **Skolemization/Herbrandization**
   - out of 337 certificates, 337 formulas skolemized/herbrandized
   - Clauses: max. 441 Mill., avg. 25 Mill., med. 71000
4. **Certificate Validation**
   - out of 337 skolemized/herbrandized formulas, 275 checked successfully
   - 45 (17) certificates not validated due to memory (time) out
     $\rightarrow$ out of these 62, 57 instances were satisfiable
   - $>$ 70% of the total runtime

Experimental Results

Benchmarks: QBFEVAL'10 set (568 formulas)
Limits: 1800 seconds and 7 GB limits

1. **Proof Extraction, Checking**
   - out of 362 solved instances, 348 proofs extracted and checked by QRPcheck
   - 14 instances lost due to memory out
2. **Certificate Extraction**
   - out of 348 proofs, 337 certificates extracted
   - 11 instances lost due to memory out
   - AND-Gates: max. 147 Mill., avg. 8 Mill., med. 369
3. **Skolemization/Herbrandization**
   - out of 337 certificates, 337 formulas skolemized/herbrandized
   - Clauses: max. 441 Mill., avg. 25 Mill., med. 71000
4. **Certificate Validation**
   - out of 337 skolemized/herbrandized formulas, 275 checked successfully
   - 45 (17) certificates not validated due to memory (time) out
     $\rightarrow$ out of these 62, 57 instances were satisfiable
   - $>$ 70% of the total runtime

Experimental Results

Benchmarks: QBFEVAL'10 set (568 formulas)
Limits: 1800 seconds and 7 GB limits

1. **Proof Extraction, Checking**
   - out of 362 solved instances, 348 proofs extracted and checked by QRPcheck
   - 14 instances lost due to memory out
2. **Certificate Extraction**
   - out of 348 proofs, 337 certificates extracted
   - 11 instances lost due to memory out
   - AND-Gates: max. 147 Mill., avg. 8 Mill., med. 369
3. **Skolemization/Herbrandization**
   - out of 337 certificates, 337 formulas skolemized/herbrandized
   - Clauses: max. 441 Mill., avg. 25 Mill., med. 71000
4. **Certificate Validation**
   - out of 337 skolemized/herbrandized formulas, 275 checked successfully
   - 45 (17) certificates not validated due to memory (time) out
     $\rightarrow$ out of these 62, 57 instances were satisfiable
   - $>$ 70% of the total runtime

Conclusion

- QBF research has seen much progress over the last years

- *Many open challenges* - some examples follow
  - **Preprocessing:** Integration in certification process
  - **Solving:** Symmetric handling of true and false QBF
  - **Encodings:** Dual representation
  - **Certificates:** Size, alternative representation
  - ...

References

[KleineBüningKF95] H. Kleine Büning et al., Resolution for Quantified Boolean Formulas. Inf. Comput. 117(1), 1995
[NiemetzPLSB12] Aina Niemetz, Mathias Preiner, Florian Lonsing, Martina Seidl, Armin Biere: Resolution-Based Certificate Extraction for QBF - (Tool Presentation). SAT 2012
[GoultiaevaVB11] A. Goultiaeva, A. Van Gelder, F. Bacchus: A Uniform Approach for Generating Proofs and Strategies for Both True and False QBF Formulas. IJCAI 2011
[BalabanovJ11] V. Balabanov, J. R. Jiang: Resolution Proofs and Skolem Functions in QBF Evaluation and Applications. CAV 2011
[VanGelder13] Allen Van Gelder: Certificate Extraction from Variable Elimination QBF Preprocessors, QBF 2013
[SeidlK14] M. Seidl, R. Köninghofer, Partial Witnesses from Preprocessed QBF, DATE 14
[JanotaGM13] M. Janota, R. Grigore, J. Marques-Silva: On QBF Proofs and Preprocessing, LPAR 2013
[JanotaM13] M. Janota, J. Marques-Silva: On Propositional QBF Expansions and Q-Resolution, ECCC13
[JanotaKMC12] MikolÃ¡s Janota, William Klieber, JoÃ£o Marques-Silva, Edmund M. Clarke: Solving QBF with Counterexample Guided Refinement. SAT 2012
[EglyLW13] Uwe Egly, Florian Lonsing, Magdalena Widl: Long-Distance Resolution: Proof Generation and Strategy Extraction in Search-Based QBF Solving. LPAR 2013
[LonsingEVG13] Florian Lonsing, Uwe Egly, Allen Van Gelder: Efficient Clause Learning for Quantified Boolean Formulas via QBF Pseudo Unit Propagation. SAT 2013
[LonsingB10] Florian Lonsing, Armin Biere: DepQBF: A Dependency-Aware QBF Solver. JSAT 7(2-3): 71-76 (2010)
[VanGelderWL12] Allen Van Gelder, Samuel B. Wood, Florian Lonsing: Extended Failed-Literal Preprocessing for Quantified Boolean Formulas. SAT 2012
[BiereLS12] Armin Biere, Florian Lonsing, Martina Seidl: Blocked Clause Elimination for QBF. CADE 2011
[GiunchigliaMN10] Enrico Giunchiglia, Paolo Marin, Massimo Narizzano: sQueezeBF: An Effective Preprocessor for QBFs Based on Equivalence Reasoning. SAT 2010
[VanGelder11] Allen Van Gelder: Variable Independence and Resolution Paths for Quantified Boolean Formulas. CP 2011
[SlivoskyS12] Friedrich Slivovsky, Stefan Szeider: Computing Resolution-Path Dependencies in Linear Time , . SAT 2012
[GoultieavaSB13] Alexandra Goultiaeva, Martina Seidl, Armin Biere: Bridging the gap between dual propagation and CNF-based QBF solving. DATE 2013
[GoultiaevaB13] Alexandra Goultiaeva, Fahiem Bacchus: Recovering and Utilizing Partial Duality in QBF. SAT 2013
[GoultiaevaB10] Alexandra Goultiaeva, Fahiem Bacchus: Exploiting QBF Duality on a Circuit Representation. AAAI 2010

[KlieberSGC10] William Klieber, Samir Sapra, Sicun Gao, Edmund M. Clarke: A Non-prenex, Non-clausal QBF Solver with Game-State Learning. SAT 2010: