Solving (Problems with) Quantified Boolean Formulas: Recent Trends and Challenges

Florian Lonsing

Knowledge-Based Systems Group, Vienna University of Technology, Austria
http://www.kr.tuwien.ac.at/staff/lonsing/

25th International Joint Conference on Artificial Intelligence (IJCAI-16)
July 9-15 2016, New York City, USA

This work is supported by the Austrian Science Fund (FWF) under grant S11409-N23.
Propositional Logic:

- Formula $\phi$ over propositional variables, Boolean domain $\mathcal{B} = \{\top, \bot\}$.
- Satisfiability problem (SAT): is $\phi$ satisfiable?
- NP-completeness of SAT.
- Modelling NP-complete problems in formal verification, AI, . . .
- A SAT solver returns a model of $\phi$ or a proof that $\phi$ has no model.

Example

Propositional formulas in conjunctive normal form (CNF):

- $\phi := (x \lor \neg y) \land (\neg x \lor y)$.
- $\phi$ is satisfiable: models $M := \{x, y\}$ or $M' := \{\bar{x}, \bar{y}\}$.
- $\phi' := (x) \land (\neg x)$ is unsatisfiable (i.e., has no model).
Success Story of SAT Solving:

- Origins: backtracking DPLL algorithm 1960 [DP60] and 1962 [DLL62].
- Clause learning (CDCL): [SS96, SS99].
- Efficient data structures and heuristics: [MMZ⁺01].
- Despite intractability: many (industrial) applications.
- SAT solver exploit structure of formulas.

SAT Research Community:

- Handbook of Satisfiability [BHvMW09].
- http://satlive.org/
- http://www.satcompetition.org/
- http://satassociation.org/
- Introductory articles [MZ09, VWM15].
Quantified Boolean Formulas (QBF):

- Existential (∃) / universal (∀) quantification of propositional variables.
- QBF satisfiability: PSPACE-completeness.
- Potentially more succinct encodings than propositional logic.
- Applications to presumably harder problems, e.g. NEXPTIME.

Example

- CNF $\phi := (x \lor \neg y) \land (\neg x \lor y)$.
- Quantifier prefix $\hat{Q} := \forall x \exists y$.
- QBF $\psi := \hat{Q} \cdot \phi$ in prenex conjunctive normal form (PCNF).
- $\psi = \forall x \exists y.(x \lor \neg y) \land (\neg x \lor y)$. 

Admittedly, the theory and results of this paper emphasize the need for further research in QBF solvers [...] Since the first complete QBF solver was presented decades after the first complete engine to solve SAT, research in this field remains at its infancy.

See e.g. [BM08] for references to further comparisons of SAT and QBF.
The Beginning of QBF Solving:

- 1998: backtracking DPLL for QBF [CGS98].
- 2002: clause learning for QBF (proofs) [GNT02, Let02, ZM02a].
- 2002: expansion (elimination) of variables [AB02].

⇒ compared to SAT (1960s), QBF still is a young field of research!
Increased Interest in QBF:

- QBFEVAL’16: largest number of participants ever.
- QBF proof systems: theoretical frameworks of solving techniques.
- CDCL (clause learning) and expansion: orthogonal solving approaches.
- QBF solving by counterexample guided abstraction refinement (CEGAR) [CGJ+03, JM15b, JKMSC16, RT15].
- New approaches, e.g., Skolem function computation [RS16].
- 10 QBF-related papers at SAT 2016 conference (27%).

QBF Research Community:

- QBFLIB: http://www.qbflib.org/index.php
- QBF Workshop 2016: http://fmv.jku.at/qbf16/
- Beyond NP Workshop: http://beyondnp.org/
Introduction (6): Motivating QBF Applications

Synthesis and Realizability of Distributed Systems:


Solving dependency quantified boolean formulas (NEXPTIME):

Introduction (6): Motivating QBF Applications

Formal verification and synthesis:


Outline

Preliminaries:
- QBF syntax and semantics.

QBF Proof Systems:
- Results in QBF proof complexity.
- Understanding and analyzing techniques implemented in QBF solvers.

A Typical QBF Workflow:
- How to encode problems as a QBF?
- How to simplify and solve a QBF?
- How to obtain the solution to a problem from a solved QBF?

Outlook and Future Work:
- Open problems and possible research directions.
Preliminaries
### Definition (Basic Definitions)

- **Boolean domain** $\mathcal{B} = \{\top, \bot\}$: truth values "true" and "false".
- **Boolean variables** $\text{Vars} = \{x, y, \ldots\}$ (arbitrarily many but finite).
- **Assignment** $A : \text{Vars} \rightarrow \mathcal{B}$
Definition (Propositional Formulas (PF))

- $\top$ and $\bot$ are PFs.
- For propositional variables $\textit{Vars}$, $(x)$ where $x \in \textit{Vars}$ is a PF.
- If $\psi$ is a PF then $\neg(\psi)$ is a PF.
- If $\psi_1$ and $\psi_2$ are PFs then $(\psi_1 \circ \psi_2)$ is a PF, $\circ \in \{\land, \lor, \rightarrow, \leftrightarrow\}$. 
Definition (Conjunctive Normal Form (CNF))

- A literal $l$ is a variable $x$ or its negation $\bar{x}$.
- A clause $C = (l_1 \lor \ldots \lor l_m)$ is a disjunction over literals.
- A formula is in conjunctive normal form (CNF) if it consists of a conjunction of clauses.

Example

$$\phi := (x \lor \neg y) \land (\neg x \lor y).$$
Propositional Logic (2)

### Definition (CNF Semantics)

- Given a CNF $\phi$ and an assignment $A$ to the variables in $\phi$.
- $\phi[A]$: replace variables $x$ in $\phi$ by $\top$ ($\bot$) if $A(x) = \top$ ($A(x) = \bot$).
- CNF $\phi$ is satisfiable iff there exists $A$ such that $\phi[A] = \top$. Otherwise, $\phi$ is unsatisfiable.

### Example

- $\phi := (x \lor \neg y) \land (\neg x \lor y)$.
- Models of $\phi$: $M := \{x, y\}$ where $M(x) = M(y) = \top$ or $M' := \{\overline{x}, \overline{y}\}$ where $M'(x) = M'(y) = \bot$. 
QBFs as Quantified Circuits:

- \(\top\) and \(\bot\) are QBFs.
- For propositional variables \(\text{Vars}\), \(\langle x \rangle\) where \(x \in \text{Vars}\) is a QBF.
- If \(\psi\) is a QBF then \(\neg(\psi)\) is a QBF.
- If \(\psi_1\) and \(\psi_2\) are QBFs then \((\psi_1 \circ \psi_2)\) is a QBF, \(\circ \in \{\land, \lor, \rightarrow, \leftrightarrow\}\).
- If \(\psi\) is a QBF and \(x \in \text{Vars}(\psi)\), then \(\forall x.(\psi)\) and \(\exists x.(\psi)\) are QBFs.

Example

\[\psi := (\forall z.(\exists y.(y \land z))) \rightarrow \neg(\forall x.(x)).\]
**QBFs in Prenex CNF:** $\psi := \hat{Q}.\phi$

- **Quantifier prefix** $\hat{Q} = Q_1B_1 \ldots Q_nB_n$, $Q_i \in \{\forall, \exists\}$, $Q_i \neq Q_j$, $B_i \subseteq \text{Vars}$, $(B_i \cap B_j) = \emptyset$.

- **Linear ordering of variables:** $x_i < x_j$ iff $x_i \in B_i$, $x_j \in B_j$, and $i < j$.

- **Quantifier-free CNF** $\phi$ over propositional variables $x_i$.

- **Assume:** $\phi$ does not contain free variables, all $x_i$ in $\hat{Q}$ appear in $\phi$. 
Example (QDIMACS Format)

\[ \exists x_1, x_3, x_4 \forall y_5 \exists x_2. \]
\[ (\bar{x}_1 \lor x_2) \land (x_3 \lor y_5 \lor \bar{x}_2) \land (x_4 \lor \bar{y}_5 \lor \bar{x}_2) \land (\bar{x}_3 \lor \bar{x}_4) \]

- Extension of DIMACS format used in SAT solving.
- Literals of variables encoded as signed integers.
- One quantifier block per line, terminated by zero.
- “a” labels \( \forall \), “e” labels \( \exists \).
- One clause per line, terminated by zero.

QDIMACS format: [http://www.qbflib.org/qdimacs.html](http://www.qbflib.org/qdimacs.html)
Semantics (1)

Recursive Definition:

- Assume that a QBF does not contain free variables.
- The QBF $\bot$ is unsatisfiable, the QBF $\top$ is satisfiable.
- The QBF $\neg(\psi)$ is satisfiable iff the QBF $\psi$ is unsatisfiable.
- The QBF $\psi_1 \land \psi_2$ is satisfiable iff $\psi_1$ and $\psi_2$ are satisfiable.
- The QBF $\psi_1 \lor \psi_2$ is satisfiable iff $\psi_1$ or $\psi_2$ is satisfiable.
- The QBF $\forall x.(\psi)$ is satisfiable iff $\psi[\neg x]$ and $\psi[x]$ are satisfiable. The QBF $\psi[\neg x]$ ($\psi[x]$) results from $\psi$ by replacing $x$ in $\psi$ by $\bot$ ($\top$).
- The QBF $\exists x.(\psi)$ is satisfiable iff $\psi[\neg x]$ or $\psi[x]$ is satisfiable.

Example

$$\psi = \forall u \exists x. (\bar{u} \lor x) \land (u \lor \bar{x})$$

satisfiable iff

- $\psi[\bar{u}] = \exists x. (\bar{x})$ satisfiable and
- $\psi[u] = \exists x. (x)$ satisfiable.
Semantics (1)

Game-Based View:

- Player $P_\exists \ (P_\forall)$ assigns existential (universal) variables.
- Goal: $P_\exists \ (P_\forall)$ wants to satisfy (falsify) the formula.
- Players pick variables from left to right wrt. quantifier ordering.
- QBF $\psi$ is satisfiable (unsatisfiable) iff $P_\exists \ (P_\forall)$ has a winning strategy.
- Winning strategy: $P_\exists \ (P_\forall)$ can satisfy (falsify) the formula regardless of opponent’s choice of assignments.
- Close relation between winning strategies and QBF certificates.

Example

$\psi = \forall u \exists x. (\bar{u} \lor x) \land (u \lor \bar{x})$.

- $P_\exists$ wins by setting $x$ to the same value as $u$. 
**Definition (Skolem/Herbrand Function)**

Let $\psi$ be a PCNF, $x$ ($y$) a universal (existential) variable.

- Let $D^\psi(v) := \{ w \in \psi \mid q(v) \neq q(w) \text{ and } w < v \}$, $q(v) \in \{\forall, \exists\}$.
- Skolem function $f_y(x_1, \ldots, x_k)$ of $y$: $D^\psi(y) = \{x_1, \ldots, x_k\}$.
- Herbrand function $f_x(y_1, \ldots, y_k)$ of $x$: $D^\psi(x) = \{y_1, \ldots, y_k\}$.

**Definition (Skolem Function Model)**

A PCNF $\psi$ with existential variables $y_1, \ldots, y_m$ is satisfiable iff $\psi[y_1/f_y(D^\psi(y_1)), \ldots, y_m/f_y(D^\psi(y_m))]$ is satisfiable.

**Definition (Herbrand Function Countermodel)**

A PCNF $\psi$ with universal variables $x_1, \ldots, x_m$ is unsatisfiable iff $\psi[x_1/f_x(D^\psi(x_1)), \ldots, x_m/f_x(D^\psi(x_m))]$ is unsatisfiable.
Example (Skolem Function Model)

\[ \psi = \exists x \forall u \exists y. (\bar{\bar{x}} \lor u \lor \bar{y}) \land (\bar{\bar{x}} \lor \bar{u} \lor y) \land (x \lor u \lor y) \land (x \lor \bar{u} \lor \bar{y}) \]

- Skolem function \( f_x = \bot \) of \( x \) with \( D_\psi(x) = \emptyset \).
- Skolem function \( f_y(u) = \bar{u} \) of \( y \) with \( D_\psi(y) = \{ u \} \).
- \( \psi[x/f_x, y/f_y(u)] = \forall u. (\bot \lor u \lor \bar{u}) \land (\bot \lor \bar{u} \lor u) \)
- Satisfiable: \( \psi[x/f_x, y/f_y(u)] = \top \)

Example (Herbrand Function Countermodel)

\[ \psi = \exists x \forall u \exists y. (x \lor u \lor y) \land (x \lor u \lor \bar{y}) \land (\bar{x} \lor \bar{u} \lor y) \land (\bar{x} \lor \bar{u} \lor \bar{y}) \]

- Herbrand function \( f_u(x) = (x) \) of \( u \) with \( D_\psi(u) = \{ x \} \).
- \( \psi[u/f_u(x)] = \exists x, y. (x \lor x \lor y) \land (x \lor x \lor \bar{y}) \land (\bar{x} \lor \bar{x} \lor y) \land (\bar{x} \lor \bar{x} \lor \bar{y}) \)
- Unsatisfiable: \( \psi[u/f_u(x)] = \exists x, y. (x \lor y) \land (x \lor \bar{y}) \land (\bar{x} \lor y) \land (\bar{x} \lor \bar{y}) \)
QBF Proof Systems
Proof Systems (1)

Definition (QBF Proof System (informal))

- Formal system $\mathcal{PS}$ consisting of inference rules.
- Inference rules to derive formulas from a QBF $\psi$.
- If $\bot$ ($\top$) derivable in $\mathcal{PS}$ from $\psi$ then $\psi$ is unsatisfiable (satisfiable).
- Proof of $\psi$ in $\mathcal{PS}$: sequence of inference steps deriving $\bot$ ($\top$).
- $\mathcal{PS}$ stronger than $\mathcal{PS}'$ if the lengths $|P|$ and $|P'|$ with $|P| < |P'|$ of the shortest proofs $P$ and $P'$ of some QBF $\psi$ in $\mathcal{PS}$ and $\mathcal{PS}'$, respectively, differ by an exponential factor.
- Formal definition by Cook and Reckhow: [CR79].

- QBF proof systems underlie implementations of QBF solvers.
- Study QBF proof systems and their strengths to improve QBF solving.
Proof Systems (2): QBF Resolution

**Definition (Q-Resolution Calculus QRES, c.f. [BKF95])**

Let $\psi = \hat{Q}.\phi$ be a PCNF and $C, C_1, C_2$ clauses.

1. **(init)** For all $x \in \hat{Q}$: \{x, \bar{x}\} \not\subseteq C and $C \in \phi$

2. **(red)** For all $x \in \hat{Q}$: \{x, \bar{x}\} \not\subseteq (C \cup \{l\})$, $q(l) = \forall$, and $l' < l$ for all $l' \in C$ with $q(l') = \exists$

3. **(res)** For all $x \in \hat{Q}$: \{x, \bar{x}\} \not\subseteq (C_1 \cup C_2)$, $\bar{p} \not\in C_1$, $p \not\in C_2$, and $q(p) = \exists$

- **Axiom** `init`, universal reduction `red`, resolution `res`.
- **PCNF** $\psi$ is unsatisfiable iff empty clause $\emptyset$ can be derived by QRES.
Proof Systems (3): QBF Resolution

Example

\[ \psi = \exists x \forall u \exists y \forall v \exists z. \]
\[ (y \lor v \lor z) \land \overline{(y \lor v \lor z)} \land (x \lor u \lor \overline{z}) \land \overline{(x \lor u \lor \overline{z})} \land (\overline{x} \lor u \lor \overline{z}) \land \overline{(\overline{x} \lor u \lor \overline{z})} \]

\[ C_1 \]
\[ C_2 \]
\[ C_3 \]
\[ C_4 \]
\[ C_5 \]

F. Lonsing (TU Wien)
Example (continued)

\[ \psi = \exists x \forall u \exists y \forall v \exists z. \]
\[ (y \lor v \lor z) \land (\bar{y} \lor \bar{v} \lor z) \land (x \lor u \lor \bar{z}) \land (\bar{x} \lor u \lor \bar{z}) \land (\bar{x} \lor \bar{u} \lor \bar{z}) \]

\[ C_1 \land C_2 \land C_3 \land C_4 \land C_5 \]

\[ \frac{C_1 \land C_2}{(v \lor \bar{v} \lor z)} \]

**Long-Distance Q-Resolution:** [ZM02a, BJ12]

- Like Q-resolution, but allow certain tautological resolvents.
- Tautological resolvent \( C \) with \( \{x, \bar{x}\} \subseteq C \): 
  - \( q(x) = \forall \)
  - Existential pivot \( p: p < x \).
- Exponentially stronger than traditional Q-resolution.
Example (continued)

\[ \psi = \exists x \forall u \exists y \forall v \exists z. \]

\[ (y \lor v \lor z) \land (\overline{y} \lor \overline{v} \lor z) \land (x \lor u \lor \overline{z}) \land (\overline{x} \lor u \lor \overline{z}) \land (\overline{x} \lor \overline{u} \lor \overline{z}) \]

\[ C_1 \]
\[ C_2 \]
\[ C_3 \]
\[ C_4 \]
\[ C_5 \]

\[ C_4 \land C_5 \]

\[ (\overline{x} \lor \overline{z}) \]

**QU-Resolution:** [VG12]

- Like Q-resolution but additionally allow universal variables as pivots.
- Exponentially stronger than traditional Q-resolution.
Example (continued)

\[ \psi = \exists x \forall u \exists y \forall v \exists z. (y \lor v \lor z) \land (\bar{y} \lor \bar{v} \lor z) \land (x \lor u \lor \bar{z}) \land (\bar{x} \lor u \lor \bar{z}) \land (\bar{x} \lor \bar{u} \lor \bar{z}) \]

\[ C_1 \quad C_2 \quad C_3 \quad C_4 \quad C_5 \]

\[ C_4 \quad C_5 \]

\[ (\bar{x} \lor \bar{z}) \]

Further Variants: [BWJ14]

- Combinations of QU- and long-distance Q-resolution.
- Existential and universal pivots, tautologies due to universal variables.
Proof Systems (5): Expansion and Instantiation

Example

\[ \psi = \exists x \forall u \exists y. (\overline{x} \lor y) \land (x \lor \overline{y}) \land (\overline{u} \lor y) \land (u \lor \overline{y}) \]

- **Expand** \(u\): copy CNF and replace \(y\) by fresh \(z\) in copy of CNF.

- **Obtain** \((\overline{x})\) from \((\overline{x} \lor y)\) and \((\overline{y})\), \((x)\) from \((x \lor \overline{z})\) and \((z)\).

**Universal Expansion:** cf. [AB02, Bie04, JKMSc16]

- Idea: eliminate all universal variables, cf. Shannon expansion [Sha49].
- Finally, apply propositional resolution (no universal reduction).
- If \(x\) innermost: replace \(\hat{Q} \forall x. \phi\) by \(\hat{Q}. (\phi[x/\top] \land \phi[x/\top])\).
- Otherwise, duplicate existential variables inner to \(x\) [Bie04, BK07].
- Based on CNF, NNF, and-inverter graphs [AB02, LB08, PS09].
Definition ($\forall$Exp+RES [JM13, BCJ14, JM15a])

- **Axiom:** \( \frac{C}{C} \) for all \( x \in \hat{Q} \): \( \{x, \overline{x}\} \not\subseteq C \) and \( C \in \phi \)

- **Instantiation:** \( \frac{C}{\{l^A | l \in C, q(l) = \exists\}} \)

  Complete assignment \( A \) to universal variables s.t. literals in \( C \) falsified, \( A_l \subseteq A \) restricted to universal variables \( u \) with \( u < l \).

- **Resolution:** \( \frac{C_1 \cup \{p^A\} \quad C_2 \cup \{\overline{p}^A\} \quad C_1 \cup C_2}{C_1 \cup C_2} \) for all \( x \in \hat{Q} \): \( \{x, \overline{x}\} \not\subseteq (C_1 \cup C_2) \)

- First, instantiate (i.e. replace) all universal variables by constants.
- Existential literals in a clause are annotated by partial assignments.
- Finally, resolve on existential literals with matching annotations.
- Instantiation and annotation mimics universal expansion.
Example (continued)

\[
\psi = \exists x \forall u \exists y. (\bar{x} \lor y) \land (x \lor \bar{y}) \land (\bar{u} \lor y) \land (u \lor \bar{y})
\]

- Complete assignments: \( A = \{\bar{u}\} \) and \( A' = \{u\} \).
- Instantiate: \((\bar{x} \lor y \bar{u}) \land (x \lor \bar{y} u) \land (y u) \land (\bar{y} \bar{u})\)
- Note: cannot resolve \((y u)\) and \((\bar{y} \bar{u})\) due to mismatching annotations.
- Obtain \((x)\) from \((x \lor \bar{y} u)\) and \((y u)\), \((\bar{x})\) from \((\bar{x} \lor y \bar{u})\) and \((\bar{y} \bar{u})\).

Different Power of QBF Proof Systems:

- Q-resolution and expansion/instantiation are incomparable [BCJ15].
- Interpreting QBFs as first-order logic formulas [SLB12, Egl16].
Typical QBF Workflow
Which problems can be modelled as a QBF?
How to encode problems as a QBF?
How to simplify QBF encodings?
How to solve a QBF?
How to obtain the solution to a problem from a solved QBF?
Problems (1)

Definition (Polynomial-Time Hierarchy, cf. [BB09, MS72])

For $k \geq 0$:

- $\Sigma_0^P := \Pi_0^P := P$, $\Sigma_{k+1}^P := NP^{\Sigma_k^P}$, $\Pi_{k+1}^P := co\Sigma_{k+1}^P$

- $\Sigma_{k+1}^P$: problems decidable in non-det. poly-time with $\Sigma_k^P$ oracle.
- $\Pi_{k+1}^P$: class of problems whose complement is in $\Sigma_{k+1}^P$.
- $\Sigma_1^P = NP$, $\Pi_1^P = coNP$, every $\Sigma_i^P$, $\Pi_i^P$ contained in PSPACE [Sto76].

Definition (Prefix Type [BB09])

A propositional formula $\phi$ has prefix type $\Sigma_0 = \Pi_0$. Given a QBF with prefix type $\Sigma_n (\Pi_n)$, the QBF $\forall B.\phi (\exists B.\phi)$ has prefix type $\Pi_{n+1} (\Sigma_{n+1})$.

Proposition (cf. [BB09])

For $k \geq 1$, the satisfiability problem of a QBF $\psi$ with prefix type $\Sigma_k (\Pi_k)$ is $\Sigma_k^P$-complete ($\Pi_k^P$-complete).
### Problems (2)

<table>
<thead>
<tr>
<th>Class</th>
<th>Prefix</th>
<th>Problems (e.g.)</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\Sigma_1^P = NP$</td>
<td>$\exists B_1. \phi$</td>
<td>SAT, checking Herbrand function countermodels of QBFs [BJ12]</td>
</tr>
<tr>
<td>$\Sigma_2^P$</td>
<td>$\exists B_1 \forall B_2. \phi$</td>
<td>MUS membership testing [JS11b, Lib05], encodings of conformant planning [Rin07], ASP-related problems [FR05], abstract argumentation [CDG+15]</td>
</tr>
<tr>
<td>$\Pi_1^P = co-NP$</td>
<td>$\forall B_1. \phi$</td>
<td>Checking Skolem function models of QBFs [BJ12]</td>
</tr>
<tr>
<td>PSPACE</td>
<td>$Q_1 B_1 \ldots Q_n B_n. \phi$ ((n) depending on problem instance)</td>
<td>LTL model checking [SC85], NFA language inclusion, games [Sch78]</td>
</tr>
</tbody>
</table>
Example (Bounded Model Checking (BMC) [BCCZ99])

- System $S$, states of $S$ as a state graph, invariant $P$.
- Goal: search for a counterexample of $P$ of bounded length.

**SAT Encoding:**

- Initial state predicate $I(s)$, transition relation $T(s, s')$.
- “Bad state” predicate $B(s)$: $s$ is a state where $P$ is violated.
- Error trace of length $k$: $I(s_0) \land T(s_0, s_1) \land \ldots \land T(s_{k-1}, s_k) \land B(s_k)$.

**QBF Encoding:** [BM08, JB07]

- $\exists s_0, \ldots, s_k \forall x, x'$. $I(s_0) \land B(s_k) \land (\bigvee_{i=0}^{k-1} ((x = s_i) \land (x' = s_{i+1}))) \rightarrow T(x, x')$.
- Only one copy of $T$ in contrast to $k$ copies in SAT encoding.
How can problems be encoded as a QBF?
Encodings (1)

**QCIR: Quantified CIRcuit**

- Format for QBFs in non-prenex non-CNF.
- Conversion tools, e.g., part of GhostQ solver [Gho16, KSGC10].

### 2 Format Specification

#### 2.1 Syntax

The following BNF grammar specifies the structure of a formula represented in QCIR (Quantified CIRcuit).

```
qcir-file ::= format-id qblock-stmt output-stmt (gate-stmt nl)*
format-id ::= #QCIR-G14 [integer] nl
qblock-stmt ::= [free(var-list) nl] qblock-quant*
qblock-quant ::= quant (var-list) nl
var-list ::= (var,)* var
lit-list ::= (lit,)* lit | ε
output-stmt ::= output(lit) nl
gate-stmt ::= gvar = ngate_type(lit-list)
  | gvar = xor(lit, lit)
  | gvar = ite(lit, lit, lit)
  | gvar = quant(var-list; lit)
quant ::= exists | forall
var ::= (A string of ASCII letters, digits, and underscores)
gvar ::= (A string of ASCII letters, digits, and underscores)
lit ::= var | -var | gvar | -gvar
ngate_type ::= and | or
```

#### 3.2 Formula in Non-Prenex Form

A formula in non-prenex form looks as follows:

```
∀z. z ∨ ∃x_1. ∃x_2. (x_1 ∧ x_2 ∧ z)
```

Definition (Prenexing, cf. [AB02, Egl94, EST+03, ETW02, GNT07])

\[(Qx. \phi) \circ \psi \equiv Qx. (\phi \circ \psi), \psi \text{ a QBF, } Q \in \{\forall, \exists\}, \circ \in \{\land, \lor\}, x \notin \text{Var}(\psi).\]

Definition (CNF transformation, cf. [Tse68, NW01, PG86])

- Given a prenex QBF \(\psi := \hat{Q}.\phi\), subformulas \(\psi_i\) of \(\psi\).
- \(\psi_i = (\psi_{i,l} \circ \psi_{i,r})\), \(\circ \in \{\lor, \land, \rightarrow, \leftrightarrow, \otimes\}\).
- Add equivalences \(t_i \leftrightarrow (\psi_{i,l} \circ \psi_{i,r})\), fresh variable \(t_i\).
- Convert each \(t_i \leftrightarrow (\psi_{i,l} \circ \psi_{i,r})\) to CNF depending on \(\circ\).
- Resulting PCNF \(\psi':\) satisfiability-equivalent to \(\psi\), size linear in \(|\psi|\).
- Safe: quantify each \(t_i\) innermost [GMN09]: \(\psi := \hat{Q}\exists t_i.\phi\).
Encodings (3)

Definition (QBF Extension Rule, cf. [Tse68, JBS+07, BCJ16])

- Let $\psi := Q_1 x_1 \ldots Q_i x_i \ldots Q_j x_j \ldots Q_n x_n. \phi$ be a PCNF.
- Consider variables $x_i, x_j$ with $x_i \leq x_j$ in $\psi$, fresh existential variable $v$.
- Add definition $v \leftrightarrow (\bar{x}_i \vee \bar{x}_j)$ in CNF: $(\bar{v} \lor \bar{x}_i \lor \bar{x}_j) \land (v \lor x_i) \land (v \lor x_j)$.
- Strong variant: quantify $v$ after $x_j$, $Q_1 x_1 \ldots Q_i x_i \ldots Q_j x_j \exists v \ldots Q_n x_n$.
- Weak variant: quantify $v$ innermost, $Q_1 x_1 \ldots Q_i x_i \ldots Q_j x_j \ldots Q_n x_n \exists v$.

Proposition (cf. [JBS+07, BCJ16])

Q-resolution with the strong extension rule is exponentially more powerful than with the weak extension rule with respect to lengths of refutations.

$\Rightarrow$ “bad” placement of Tseitin variables in encoding phase may have negative impact on solving in a later stage.
Encodings (4): QParity

Definition (QParity Function [BCJ15])

\[ Q_{Parity_n} := \exists x_1, \ldots, x_n \forall y. \text{XOR}(\text{XOR}(\ldots \text{XOR}(x_1, x_2), \ldots, x_n), y). \]

CNF \( \phi \) of \( Q_{Parity_n} \) by Tseitin translation:

\[
(t_1 \leftrightarrow \text{XOR}(x_1, x_2)) \land \\
\land_{1<i<n} (t_i \leftrightarrow \text{XOR}(t_{i-1}, x_{i+1})) \land \\
(t_n \leftrightarrow \text{XOR}(t_{n-1}, y)) \land (t_n)
\]

Prefix by weak extension rule: \( \hat{Q}_W := \exists x_1, \ldots, x_n \forall y \exists t_1, \ldots, t_n \)

Prefix by strong extension rule: \( \hat{Q}_S := \exists x_1, \ldots, x_n \exists t_1, \ldots, t_{n-1} \forall y \exists t_n \)

Proposition ([BCJ15, BCJ16])

- The PCNF \( \hat{Q}_W \cdot \phi \) has only exponential Q-resolution refutations.
- The PCNF \( \hat{Q}_S \cdot \phi \) has polynomial Q-resolution refutations.
Encodings (5): QParity

\[ \hat{Q}_W \phi := \exists x_1, x_2, x_3 \forall y \]

\[ . \ \text{XOR}_3(\text{XOR}_2(\text{XOR}_1(x_1, x_2), x_3), y) \]

\[ t_1 : \quad (\bar{t}_1 \lor x_1 \lor x_2) \land \]
\[ \quad (\bar{t}_1 \lor \bar{x}_1 \lor \bar{x}_2) \land \]
\[ \quad (t_1 \lor \bar{x}_1 \lor x_2) \land \]
\[ \quad (t_1 \lor x_1 \lor \bar{x}_2) \land \]

\[ t_2 : \quad (\bar{t}_2 \lor t_1 \lor x_3) \land \]
\[ \quad (\bar{t}_2 \lor \bar{t}_1 \lor \bar{x}_3) \land \]
\[ \quad (t_2 \lor \bar{t}_1 \lor x_3) \land \]
\[ \quad (t_2 \lor t_1 \lor \bar{x}_3) \land \]

\[ t_3 : \quad (\bar{t}_3 \lor t_2 \lor y) \land \]
\[ \quad (\bar{t}_3 \lor \bar{t}_2 \lor \bar{y}) \land \]
\[ \quad (t_3 \lor \bar{t}_2 \lor y) \land \]
\[ \quad (t_3 \lor t_2 \lor \bar{y}) \land \]

\[ \text{out} : \quad (t_3) \]
$\hat{Q}_W.\phi := \exists x_1, x_2, x_3 \forall y \exists t_1, t_2, t_3. \ XOR_3(\ XOR_2(\ XOR_1(x_1, x_2), x_3), y)$

t_1 \iff \ XOR(x_1, x_2)

t_2 \iff \ XOR(t_1, x_3)

t_3 \iff \ XOR(t_2, y)

\begin{align*}
t_1 & : (\overline{t}_1 \lor x_1 \lor x_2) \land \\
& \quad (\overline{t}_1 \lor \overline{x}_1 \lor \overline{x}_2) \land \\
& \quad (t_1 \lor \overline{x}_1 \lor x_2) \land \\
& \quad (t_1 \lor x_1 \lor \overline{x}_2) \land \\
t_2 & : (\overline{t}_2 \lor t_1 \lor x_3) \land \\
& \quad (\overline{t}_2 \lor \overline{t}_1 \lor \overline{x}_3) \land \\
& \quad (t_2 \lor \overline{t}_1 \lor x_3) \land \\
& \quad (t_2 \lor t_1 \lor \overline{x}_3) \land \\
t_3 & : (\overline{t}_3 \lor t_2 \lor y) \land \\
& \quad (\overline{t}_3 \lor \overline{t}_2 \lor \overline{y}) \land \\
& \quad (t_3 \lor \overline{t}_2 \lor y) \land \\
& \quad (t_3 \lor t_2 \lor \overline{y}) \land \\
out & : (t_3)
\end{align*}
$\hat{Q}_S.\phi := \exists x_1, x_2, x_3 \quad \forall y \quad . \text{XOR}_3(\text{XOR}_2(\text{XOR}_1(x_1, x_2), x_3), y)$

\[
\begin{array}{l}
\text{out} : (t_3) \\
\text{t}_3 : (\overline{t}_3 \lor t_2 \lor y) \land \\
(\overline{t}_3 \lor t_2 \lor \overline{y}) \land \\
(t_3 \lor t_2 \lor y) \land \\
(t_3 \lor t_2 \lor \overline{y}) \land \\
\text{t}_2 : (\overline{t}_2 \lor t_1 \lor x_3) \land \\
(\overline{t}_2 \lor \overline{t}_1 \lor \overline{x}_3) \land \\
(t_2 \lor t_1 \lor x_3) \land \\
(t_2 \lor \overline{t}_1 \lor \overline{x}_3) \land \\
\text{t}_1 : (\overline{t}_1 \lor \overline{x}_1 \lor \overline{x}_2) \land \\
(t_1 \lor \overline{x}_1 \lor x_2) \land \\
(t_1 \lor x_1 \lor \overline{x}_2) \land \\
\end{array}
\]

$t_1 \leftrightarrow \text{XOR}(x_1, x_2)$

$t_2 \leftrightarrow \text{XOR}(t_1, x_3)$

$t_3 \leftrightarrow \text{XOR}(t_2, y)$
\( \hat{Q}_S \phi := \exists x_1, x_2, x_3, t_1, t_2 \forall y \exists t_3. \ XOR_3(\ XOR_2(\ XOR_1(x_1, x_2), x_3), y) \)

\[
\begin{array}{c}
\otimes t_3 \\
\downarrow \\
\otimes t_2 \\
\downarrow \\
\otimes t_1 \\
\downarrow \\
x_1 \\
\downarrow \\
x_2
\end{array}
\]

\[
\begin{array}{l}
t_1 \leftrightarrow XOR(x_1, x_2) \\
t_2 \leftrightarrow XOR(t_1, x_3) \\
t_3 \leftrightarrow XOR(t_2, y)
\end{array}
\]

\[
\begin{array}{l}
t_1 : (\bar{t}_1 \lor x_1 \lor x_2) \land \\
(\bar{t}_1 \lor \bar{x}_1 \lor \bar{x}_2) \land \\
(t_1 \lor \bar{x}_1 \lor x_2) \land \\
(t_1 \lor x_1 \lor \bar{x}_2) \land \\
\hline
t_2 : (\bar{t}_2 \lor t_1 \lor x_3) \land \\
(\bar{t}_2 \lor t_1 \lor \bar{x}_3) \land \\
(t_2 \lor \bar{t}_1 \lor x_3) \land \\
(t_2 \lor t_1 \lor \bar{x}_3) \land \\
\hline
t_3 : (\bar{t}_3 \lor t_2 \lor y) \land \\
(\bar{t}_3 \lor \bar{t}_2 \lor \bar{y}) \land \\
(t_3 \lor \bar{t}_2 \lor y) \land \\
(t_3 \lor t_2 \lor \bar{y}) \land \\
\hline
\text{out} : (t_3)
\end{array}
\]
How can QBF encodings be simplified?
Preprocessing (1)

Preprocessing as Incomplete Solving:

- Apply Q-resolution and expansion in restricted and bounded fashion.
- E.g. Bloqquer \([BLS11, HJL^{+15}]\) and sQueueBF\([GMN10b]\).
- Failed literal detection \([LB11, VGWL12]\): find necessary assignments.

Reconstructing Structure:

- Recover non-CNF structure from Tseitin encodings \([GB13, KSGC10]\).
- Move definition variables in prefix outwards, e.g. QParity function.

Effect on Solver Performance: \([LSVG16]\)

- Iterative and incremental preprocessing may be powerful.
- Preprocessing may blur formula structure and thus be harmful.
### Preprocessing (2)

<table>
<thead>
<tr>
<th>Category/ Solvers</th>
<th>Number Solved</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Best</td>
</tr>
<tr>
<td><strong>NO Bloggger (solvers perform better without Bloggger)</strong></td>
<td></td>
</tr>
<tr>
<td>bGhostQ-CEGAR</td>
<td>142</td>
</tr>
<tr>
<td>GhostQ-CEGAR</td>
<td>142</td>
</tr>
<tr>
<td>GhostQ</td>
<td>122</td>
</tr>
<tr>
<td>sDual_Ooq</td>
<td>118</td>
</tr>
<tr>
<td>sDual_Ooq</td>
<td>105</td>
</tr>
<tr>
<td><strong>WANT Bloggger (solvers perform better with Bloggger)</strong></td>
<td></td>
</tr>
<tr>
<td>RAReQS</td>
<td>132</td>
</tr>
<tr>
<td>DepQBF-lazy-qpup</td>
<td>128</td>
</tr>
<tr>
<td>DepQBF</td>
<td>125</td>
</tr>
<tr>
<td>Hiqqer3</td>
<td>117</td>
</tr>
<tr>
<td>Qoq</td>
<td>93</td>
</tr>
<tr>
<td>QuBE</td>
<td>91</td>
</tr>
<tr>
<td>Nenofex</td>
<td>68</td>
</tr>
</tbody>
</table>

- QBF Gallery 2013 [LSVG16]: QBFLIB set (276 formulas).
- Solver performance with and without preprocessing by Bloggger.
- Preprocessing may be harmful to the performance of some solvers.
Preprocessing (3): Prefix Ordering Matters

Definition (Blocking Literal, Blocked Clause [Kul99, BLS11, HJL+15])

Let $\psi = \hat{Q}.\phi$ be a PCNF and $C \in \phi$ a clause.

- **blocking literal** $l$: $l \in C$ with $q(l) = \exists$ such that for all $C' \in \phi$ with $\overline{l} \in C'$, there exists $l'$ with $l' \leq l$ such that $\{l', \overline{l}'\} \subseteq (C \cup (C' \setminus \{\overline{l}\}))$.
- A clause $C$ is **blocked** if it contains a blocking literal.
- Removing blocked clauses preserves satisfiability.

Example

$\psi = \exists x \forall u \exists y. (\overline{x} \lor y) \land (x \lor \overline{y}) \land (\overline{u} \lor y) \land (u \lor \overline{y})$

- No clause in $\psi$ is blocked.
- Informally, inspect all resolvents on potential blocking literals.
- Prefix ordering has to be taken into account in QBF preprocessing.
How can a QBF be solved?
Solving (2): QCDCL

High-Level Workflow:
- **Assign decision variables** starting at left end of prefix of $\psi[A]$.
- **Propagation**: simplify $\psi$ under $A$ and universal reduction.
- **Conflict**: $\psi[A] = \bot$: CNF $\phi$ contains a falsified clause.
- **Solution**: $\psi[A] = \top$: all clauses in CNF of $\psi$ satisfied.
High-Level Workflow:

- Clause (cube) learning based on Q-resolution.
- Asserting clause (cube) $C$: $C[A']$ unit for some $A' \subseteq A$.
- Empty clause (cube) $C = \emptyset$: formula proved UNSAT (SAT).
- QCDCL solvers, e.g., [LB10a, GMN10a, KSGC10, ZM02b]
Solving (3): QCDCL

Result qcdcl (PCNF $\psi$)

Result $R = \text{UNDEF}$;
Assignment $A = \emptyset$;

while (true)

  /* Simplify under $A$. */
  $(R, A) = \text{qbcp}(\psi, A)$;
  if ($R == \text{UNDEF}$)

    /* Decision making. */
    $A = \text{assign\_dec\_var}(\psi, A)$;
  else

    /* Backtracking. */
    /* $R == \text{UNSAT/SAT} */
    $B = \text{analyze}(R, A)$;
    if ($B == \text{INVALID}$)

      return $R$;
    else

      $A = \text{backtrack}(B)$;
Solving (4): QCDCL

**Definition (Unit Literal Detection [CGS98])**

- Given a QBF $\psi$, a clause $C \in \psi$ is *unit* if $C = (l)$ and $q(l) = \exists$.
- *Unit literal detection (UL)* assigns $\text{var}(l)$ to satisfy the unit clause $C = (l)$.
- (If $q(l) = \forall$ then $C$ is effectively empty by universal reduction.)

**Definition (Pure Literal Detection [CGS98])**

- A literal $l$ is *pure* in a QBF $\psi$ if there are clauses which contain $l$ but no clauses which contain $\overline{l}$.
- *Pure literal detection (PL)* assigns $\text{var}(l)$ of an existential (universal) pure literal $l$ so that clauses are satisfied (not satisfied, i.e. shortened).
Solving (5): QCDCL

Definition (Boolean Constraint Propagation for QBF (QBCP))

1. Given a PCNF $\psi$ and the empty assignment $A = \{\}$, i.e. $\psi[A] = \psi$.
2. Apply universal reduction (UR) to $\psi[A]$ to get $\psi'$.
3. Apply UL to $\psi'$.
4. Apply PL to $\psi'$.
5. Add assignments by by UL and PL to $A$, set $\psi := \psi'$, repeat steps 1-3.
6. Stop if $A$ does not change anymore or if $\psi[A] = \top$ or $\psi[A] = \bot$.

Properties of QBCP:

- Result: extended assignment $A'$ and simplified PCNF $\psi' = \psi[A']$ by UL, PL, and UR such that $\psi \equiv_{sat} \psi'$.
- QBCP can assign variables out of prefix ordering.
Solving (6): QBCP and Implication Graphs

Definition (Implication Graph)

- Let $\psi$ be the original QBF.
- Vertices: literals in $A$ (variable assignments), special vertex $\emptyset$ denoting a clause $C \in \psi$ such that $C[A] = \bot$ by UR.
- For assignments $\{l\}$ by UL from a unit clause $C[A]$: the clause $\text{ante}(l) := C$ with $C \in \psi$ is the antecedent clause of assignment $\{l\}$.
- Define $\text{ante}(\emptyset) = C$, for a clause $C \in \psi$ such that $C[A] = \bot$.
- Edges: $(x, y) \in E$ if $y$ assigned by UL and literal $\neg x \in \text{ante}(y)$.

- Antecedent clauses in the original PCNF $\psi$ are recorded.
- Implication graph constructed on the fly during QBCP.
- Conflict graph: implication graph containing empty clause $\emptyset$. 
Example (Clause Learning)

\[ \psi = \exists x_1, x_3, x_4 \forall y_5 \exists x_2. \\
\quad (\bar{x}_1 \lor x_2) \land (x_3 \lor y_5 \lor \bar{x}_2) \land (x_4 \lor \bar{y}_5 \lor \bar{x}_2) \land (\bar{x}_3 \lor \bar{x}_4) \]

- Make decision \( A = \{x_1\} \):
  \[ \psi[\{x_1\}] = \exists x_3, x_4 \forall y_5 \exists x_2. (x_2) \land (x_3 \lor y_5 \lor \bar{x}_2) \land (x_4 \lor \bar{y}_5 \lor \bar{x}_2) \land (\bar{x}_3 \lor \bar{x}_4) \]

- By UL: \[ \psi[\{x_1, x_2\}] = \exists x_3, x_4 \forall y_5. (x_3 \lor y_5) \land (x_4 \lor \bar{y}_5) \land (\bar{x}_3 \lor \bar{x}_4). \]

- By UR: \[ \psi[\{x_1, x_2\}] = \exists x_3, x_4. (x_3) \land (x_4) \land (\bar{x}_3 \lor \bar{x}_4) \]

- By UL: \[ \psi[\{x_1, x_2, x_3, x_4\}] = \bot, \text{ clause } (\bar{x}_3 \lor \bar{x}_4) \text{ conflicting.} \]

Conflict graph \( G \):

\[ \begin{array}{c}
  x_1 \\
  \downarrow \\
  x_2 \\
  \downarrow \\
  x_3 \\
  \downarrow \\
  \emptyset \\
  \downarrow \\
  x_4 \\
\end{array} \]

Antecedent clauses:

\[ \begin{align*}
  x_2 & : (\bar{x}_1 \lor x_2) \\
  x_3 & : (x_3 \lor y_5 \lor \bar{x}_2) \\
  x_4 & : (x_4 \lor \bar{y}_5 \lor \bar{x}_2) \\
  \emptyset & : (\bar{x}_3 \lor \bar{x}_4)
\end{align*} \]
Example (Clause Learning, continued)

Prefix: $\exists x_1, x_3, x_4 \forall y_5 \exists x_2$

Assignment $A = \{x_1, x_2, x_3, x_4\}$

Conflict graph $G$:

- Idea: start at $\emptyset$, select pivots in reverse assignment ordering.
- Resolve antecedents of $x_4, x_3$.
- $Q$-resolution [BKF95] disallows tautologies like $(\bar{y_5} \lor y_5 \lor \bar{x_2})$!
- Pivot selection more complex than in CDCL for SAT.

Antecedent clauses:

- $x_2 : (\bar{x_1} \lor x_2)$
- $x_3 : (x_3 \lor y_5 \lor \bar{x_2})$
- $x_4 : (x_4 \lor \bar{y_5} \lor \bar{x_2})$
- $\emptyset : (\bar{x_3} \lor \bar{x_4})$
Solving (9): QCDCL

Example (Clause Learning, continued)

Prefix: $\exists x_1, x_3, x_4 \land y_5 \exists x_2$

Assignment $A = \{x_1, x_2, x_3, x_4\}$

Conflict graph $G$:

$\begin{array}{c}
\text{x}_1 \\
\text{x}_2 \\
\text{x}_3 \\
\text{x}_4 \\
\emptyset
\end{array}$

- Avoid tautologies: resolve on UR-blocking existentials.
- Select pivots: $x_4, x_2, x_3, x_2$.
- Q-resolution derivation of a learned clause ($\bar{x}_1$) is not regular, i.e. resolve on variables more than once.

Antecedent clauses:

- $x_2: (\bar{x}_1 \lor x_2)$
- $x_3: (x_3 \lor y_5 \lor \bar{x}_2)$
- $x_4: (x_4 \lor \bar{y}_5 \lor \bar{x}_2)$
- $\emptyset: (\bar{x}_3 \lor \bar{x}_4)$
Solving (10): QCDCL

Clause Learning by Traditional Q-Resolution [BKF95]:

- Avoid tautologies by appropriate pivot selection [GNT06].
- Derivation of a learned clause may be exponential [VG12].
- Annotate nodes in conflict graph with intermediate resolvents, resulting in tree-like (instead of linear) Q-resolution derivations of learned clauses [LEG13].

Clause Learning by Long Distance Q-Resolution [ZM02a, BJ12]:

- First implementation in quaffle:
- Select pivots in strict reverse assignment ordering.
- Every resolution step is a valid LDQ-resolution step [ZM02a, ELW13].
Solving (11): QCDCL

Example (Clause Learning, continued)

Prefix: $\exists x_1, x_3, x_4 \forall y_5 \exists x_2$
Assignment $A = \{ x_1, x_2, x_3, x_4 \}$
Conflict graph $G$:

- Start at $\emptyset$, always select pivots in reverse assignment ordering.
- Resolve antecedents of $x_4, x_3, x_2$.
- Pivots obey order restriction of LDQ-resolution.
- Derivation of learned clause is regular, size linear in $|G|$. 

Antecedent clauses:

- $x_2 : (\bar{x}_1 \lor x_2)$
- $x_3 : (x_3 \lor y_5 \lor \bar{x}_2)$
- $x_4 : (x_4 \lor \bar{y}_5 \lor \bar{x}_2)$
- $\emptyset : (\bar{x}_3 \lor \bar{x}_4)$
Solving (12): QCDCL for Satisfiable QBFs

Definition (Model Generation, cf. [GNT06, Let02, ZM02b])

Let $\psi = \hat{Q}.\phi$ be a PCNF.

\[ C = \bigwedge_{i \in A} \] is a cube where $\{x, \bar{x}\} \not\subset C$ and $A$ is an assignment with $\psi[A] = \top$, i.e. every clause of $\psi$ satisfied under $A$.

Cube Learning Dual to Clause Learning:

- Cube $C$ by model generation: $v \in C$ ($\bar{v} \in C$) if $v$ assigned to $\top$ ($\bot$).
- $C$ (also called cover set): implicant of CNF $\phi$, i.e. $C \Rightarrow \phi$.
- Model generation is an axiom of QRES.
- Q-resolution and existential reduction on cubes.
- Learn asserting cubes similar to asserting clauses.
- PCNF $\psi$ is satisfiable iff the empty cube can be derived from $\psi$. 
Solving (13): QCDCL for Satisfiable QBFs

Example

\[ \psi = \exists x \forall u \exists y. (\bar{x} \lor u \lor \bar{y}) \land (\bar{x} \lor \bar{u} \lor y) \land (x \lor u \lor y) \land (x \lor \bar{u} \lor \bar{y}) \]

- By model generation: derive cubes \((\bar{x} \land u \land \bar{y})\) and \((\bar{x} \land \bar{u} \land y)\).
- By existential reduction: reduce trailing \(\bar{y}\) from \((\bar{x} \land u \land \bar{y})\), \(y\) from \((\bar{x} \land \bar{u} \land y)\).
- Resolve \((\bar{x} \land \bar{u})\) and \((\bar{x} \land u)\) on universal \(u\).
- Reduce \((\bar{x})\) to derive \(\emptyset\).
Solving (14): QCDCL for Satisfiable QBFs

QCDCL and Cube Learning in Practice:
- PCNF $\psi := \hat{Q}. \phi$ with quantifier prefix $\hat{Q}$ and CNF $\phi$.
- Original clauses $\phi$, learned clauses $\theta$ and cubes $\gamma$.
- Properties: $\hat{Q}. \phi \equiv_{sat} \hat{Q}. (\phi \land \theta)$ and $\hat{Q}. \phi \equiv_{sat} \hat{Q}. (\phi \lor \gamma)$.

Problem: [RBM97, Let02]
- Easy formula with exponential DNF (and exponential cube proofs):
  $$\psi = \forall u_1 \exists x_1 \ldots \forall u_n \exists x_n. \bigwedge_{i=1}^{n} [(u_i \lor \bar{x}_i) \land (\bar{u}_i \lor x_i)]$$

Generalized Axioms: [LBB+15, LES16]
- Generalize model generation (axiom) to derive shorter cubes $C$ from assignments $A$ in QCDCL where $\psi[A]$ is satisfiable.
- In general, $C \nRightarrow \phi$. 
**Example ([CGJ+03, JS11a, JKMC12, JKMSC16])**

Let $\psi := \exists X \forall Y. \phi$ be a one-alternation QBF, $\phi$ a non-CNF formula.

- $\psi$ is satisfiable iff $\psi' := \bigwedge_{y \in B^{|Y|}} \phi[Y/y]$ is satisfiable.
- $\psi'$: full expansion of $\forall Y$ over all possible assignments $y$ of $Y$.
- Let $U \subseteq B^{|Y|}$ and $\text{Abs}(\psi) := \bigwedge_{y \in U} \phi[Y/y]$ be a partial expansion.
- If abstraction $\text{Abs}(\psi)$ is unsatisfiable, then $\psi$ is unsatisfiable.
- Otherwise, consider a model (candidate solution) $x \in B^{|X|}$ of $\text{Abs}(\psi)$.
- If $x$ is also a model of the full expansion $\psi'$, then $\psi$ is satisfiable.
  - $x$ is a model of $\psi'$ iff $\forall Y. \phi[X/x]$ is satisfiable.
  - $\forall Y. \phi[X/x]$ is satisfiable iff $\exists Y. \neg \phi[X/x]$ is unsatisfiable.
  - Let $y$ be a model of $\exists Y. \neg \phi[X/x]$, if one exists (counterexample to $x$).
- Otherwise, refine $\text{Abs}(\psi)$ by $U := U \cup \{y\}$.

Used in 2QBF solving [RTM04, BJS+16], RAReQS solver (recursive).
Solving (16): The Use of SAT Technology

**Proposition**

Given a PCNF $\psi := \hat{Q}.\phi$. If a clause $C$ can be derived from $\phi$ by a SAT solver, then $C$ can be derived from $\psi$ by QU-resolution.

**Coupling QCDCL with SAT Solving:**

- Clauses learned from $\phi$ by CDCL are shared with QCDCL [SB05].
- Models of $\phi$ found by SAT solver guide search process in QCDCL.
- SAT-based generalizations of Q-resolution axioms in QCDCL [LES16].

**Nested and Levelized SAT Solving:**

- Solve $\exists B_1.\phi_1 \land (\forall B_2.\phi_2)$ by solving $\exists B_1.\phi_1 \land (\exists B_2.\neg\phi_2)$ with nested SAT solvers, applicable to arbitrary nestings [BJT16, JTT16].
- Invoke two SAT solvers $S_\forall$ and $S_\exists$ with respect to quantifier blocks, prefix processed from left to right [THJ15].
How to obtain the solution to a problem from a solved QBF?
Q-Resolution Proofs:
- QCDCL solvers produce derivations $P$ of the empty clause/cube.
- Proof $P$ can be filtered out of derivations of all learned clauses/cubes.

Extracting Skolem/Herbrand Functions from Proofs:
- By inspection of $P$, run time linear in $|P|$ ($|P|$ can be exponential).
- Extraction from long-distance Q-resolution proofs [BJJW15].
- Approaches to compute winning strategies from $P$ [GGB11, ELW13].
Definition (Extracting Herbrand functions [BJ11, BJ12])

Let $P$ be a proof (Q-resolution DAG) of the empty clause $\emptyset$.

- Visit clauses in $P$ in topological ordering.
- Inspect universal reduction steps $C' = UR(C)$.
- Update Herbrand functions of variables $u$ reduced from $C$ by $C'$. 

Proofs and Certificates (2)

Example (Extracting Herbrand Functions [BJ11, BJ12])

\[ \psi = \exists x \forall u \exists y. (x \lor u \lor y) \land (x \lor u \lor \overline{y}) \land (\overline{x} \lor \overline{u} \lor y) \land (\overline{x} \lor \overline{u} \lor \overline{y}) \]

- Literal \( u \) reduced from \( (x \lor u) \), update: \( f_u(x) := (x) \).
- Literal \( \overline{u} \) reduced from \( (\overline{x} \lor \overline{u}) \), update: \( f_u(x) := f_u(x) \lor \neg(\overline{x}) = (x) \).
- Unsatisfiable: \( \psi[u/f_u(x)] = \exists x, y. (x \lor y) \land (x \lor \overline{y}) \land (\overline{x} \lor y) \land (\overline{x} \lor \overline{y}) \)
Proofs and Certificates (3): Special Case

Example

Let $\psi := \exists X \forall Y. \phi$ and $\psi' := \forall Y \exists X. \phi$ be one-alternation QBFs.

- If $\psi$ satisfiable: all Skolem functions are constant.
- If $\psi'$ unsatisfiable: all Herbrand functions are constant.
- No need to produce derivations of the empty clause/cube.
- QBF solvers can directly output values of Skolem/Herbrand functions.
- Useful for modelling and solving problems in $\Sigma^P_2$ and $\Pi^P_2$.
- QDIMACS output format specification.
Outlook and Future Work
QBF in Practice:

- QBF tools are not (yet) a push-button technology.
- Pitfalls: Tseitin encodings, premature preprocessing.
- Goal: integrated workflow without the need for manual intervention.

Challenges:

- Extracting proofs and certificates in workflows including preprocessing [HSB14a, HSB14b] and incremental solving [MMLB12, LE14].
- Integrating dependency schemes [SS09, LB10b, VG11, PSS16] in workflows to relax the linear quantifier ordering.
- Implementations of QCDCL do not harness the full power of Q-resolution [Jan16].
- Combining strengths of orthogonal solving approaches.
Outlook and Future Work (2)

- QBF Gallery 2013 application benchmarks [LSVG16].
- 6 sets, 150 formulas each, 900 sec timeout, 7 GB memory limit.
- Diverse solver performance depending on implemented approaches.
Take Home Messages:

- Assuming that $\text{NP} \neq \text{PSPACE}$, QBF is more difficult than SAT...
- ...which is reflected in the complexity of solver implementations...
- ...but allows for exponentially more succinct encodings than SAT.
- The computational hardness of QBF motivates exploring alternative approaches (e.g. CEGAR, expansion) in addition to QCDCL.
- Number of quantifier alternations vs. observed hardness.
- Document and publish your tools and benchmarks!

QBFEVAL'16: http://www.qbfclub.org/qbfeval16.php
Appendix
[Appendix] Syntax

Definition (QBFs as First-Order Logic Formulas [SLB12])

Mapping $[\cdot] : QBF \to FOL$ with respect to unary FOL predicate $p$:

\[
\begin{align*}
[\exists x. \phi] &= \exists x. [\phi] \\
[\phi \lor \psi] &= [\phi] \lor [\psi] \\
[x] &= p(x) \\
[\top] &= p(true) \\
[\psi] &= [\phi] \wedge p(true) \land \neg p(false)
\end{align*}
\]

It holds that $p(true)$ ($p(false)$) is true (false) in every FOL interpretation.

Proposition ([SLB12])

The QBF $\psi$ is satisfiable iff $[\psi] \land p(true) \land \neg p(false)$ is satisfiable.
Example (Clause Selection and Clausal Abstraction [JM15b, RT15])

Let $\psi := \forall X \exists Y. \phi$ be a one-alternation QBF, $\phi$ a CNF.

- $\psi$ unsatisfiable iff, for some $x \in B^{\left| X \right|}$, $\exists Y. \phi[X/x]$ unsatisfiable.
- Think of $x \in B^{\left| X \right|}$ as a selection $\phi^x_S \subseteq \phi$ of clauses.
- Clause $C \in \phi^x_S$ iff $C$ not satisfied by $x$, i.e. $C[X/x] \neq T$.
- If $\exists Y. \phi^x_S[X/x]$ unsatisfiable then $\exists Y. \phi[X/x]$ and $\psi$ unsatisfiable.
- Otherwise, consider model $y \in B^{\left| Y \right|}$ of $\exists Y. \phi^x_S[X/x]$.
- Find new $x' \in B^{\left| X \right|}$ such that there exists $C \in \phi^{x'}_S$ with $C[Y/y] \neq T$.
- If no such $x'$ exists then $\psi$ is satisfiable.
- CEGAR: find candidate solutions $x$ and counterexamples $y$ by SAT solving, refinement step blocks unsuccessful selections $\phi^x_S$. 
References
Please note: since the duration of this talk is limited, the list of references below is incomplete and does not reflect the history and state of the art in QBF research in full accuracy.


References II

Proof Complexity of Resolution-based QBF Calculi.

Extension Variables in QBF Resolution.
Beyond NP Workshop 2016 at AAAI-16.

[BHvMW09] Armin Biere, Marijn Heule, Hans van Maaren, and Toby Walsh, editors.

[Bie04] Armin Biere.
Resolve and Expand.

Resolution Proofs and Skolem Functions in QBF Evaluation and Applications.

Unified QBF certification and its applications.
References III


<table>
<thead>
<tr>
<th>Reference</th>
<th>Authors</th>
<th>Title</th>
<th>Details</th>
</tr>
</thead>
</table>
References V


References VI

[Egl16] Uwe Egly.  
On Stronger Calculi for QBFs.  
To appear in the proceedings of SAT 2016, LNCS, Springer.  

[ELW13] Uwe Egly, Florian Lonsing, and Magdalena Widl.  
Long-Distance Resolution: Proof Generation and Strategy Extraction in Search-Based QBF Solving.  

[EST+03] Uwe Egly, Martina Seidl, Hans Tompits, Stefan Woltran, and Michael Zolda.  
Comparing Different Prenexing Strategies for Quantified Boolean Formulas.  

[ETW02] Uwe Egly, Hans Tompits, and Stefan Woltran.  
On Quantifier Shifting for Quantified Boolean Formulas.  

[FR05] Wolfgang Faber and Francesco Ricca.  
Solving Hard ASP Programs Efficiently.  
References VII


References VIII

[GMN10a] Enrico Giunchiglia, Paolo Marin, and Massimo Narizzano.
QuBE7.0.

[GMN10b] Enrico Giunchiglia, Paolo Marin, and Massimo Narizzano.
sQueezeBF: An Effective Preprocessor for QBFs Based on Equivalence Reasoning.

[GNT02] Enrico Giunchiglia, Massimo Narizzano, and Armando Tacchella.
Learning for Quantified Boolean Logic Satisfiability.

[GNT06] Enrico Giunchiglia, Massimo Narizzano, and Armando Tacchella.
Clause/Term Resolution and Learning in the Evaluation of Quantified Boolean Formulas.

Quantifier Structure in Search-Based Procedures for QBFs.

A Synthesized Algorithm for Interactive Consistency.
<table>
<thead>
<tr>
<th>Reference</th>
<th>Authors</th>
<th>Title</th>
<th>Conference/Journal</th>
<th>Pages/Publication Details</th>
</tr>
</thead>
</table>
References


References XI


References XII


References XIII

[LES16] Florian Lonsing, Uwe Egly, and Martina Seidl.
Q-Resolution with Generalized Axioms.
To appear in the proceedings of SAT 2016, LNCS, Springer.

[Let02] Reinhold Letz.
Lemma and Model Caching in Decision Procedures for Quantified Boolean Formulas.

[Lib05] Paolo Liberatore.
Redundancy in logic I: CNF propositional formulae.

[LSVG16] Florian Lonsing, Martina Seidl, and Allen Van Gelder.
The QBF Gallery: Behind the scenes.

Verification of partial designs using incremental QBF solving.
References XIV

Chaff: Engineering an Efficient SAT Solver.

The Equivalence Problem for Regular Expressions with Squaring Requires Exponential Space.

Robust QBF Encodings for Sequential Circuits with Applications to Verification, Debug, and Test.

Boolean satisfiability from theoretical hardness to practical success.
[NW01] Andreas Nonnengart and Christoph Weidenbach.
Computing Small Clause Normal Forms.

[PG86] David A. Plaisted and Steven Greenbaum.
A Structure-Preserving Clause Form Translation.

[PS09] Florian Pigorsch and Christoph Scholl.
Exploiting structure in an AIG based QBF solver.

Long Distance Q-Resolution with Dependency Schemes.


CNF and DNF Considered Harmful for Computing Prime Implicants/Implicates.
[Rin07] Jussi Rintanen.
Asymptotically Optimal Encodings of Conformant Planning in QBF.

Incremental Determinization.

CAQE: A Certifying QBF Solver.
In FMCAD, pages 136–143. IEEE, 2015.

A Comparative Study of 2QBF Algorithms.
In SAT, 2004.

[SB05] Horst Samulowitz and Fahiem Bacchus.
Using SAT in QBF.

The Complexity of Propositional Linear Temporal Logics.
References XVII

[Sch78] Thomas J Schaefer.
On the Complexity of Some Two-Person Perfect-Information Games.

[Sha49] Claude Elwood Shannon.
The Synthesis of Two-Terminal Switching Circuits.

qbf2epr: A Tool for Generating EPR Formulas from QBF.

GRASP - a new search algorithm for satisfiability.

GRASP: A Search Algorithm for Propositional Satisfiability.

Backdoor Sets of Quantified Boolean Formulas.
References XVIII


