On CDCL vs Resolution in QBF

Olaf Beyersdorff

Friedrich Schiller University Jena, Germany

joint work with
Joshua Blinkhorn and Meena Mahajan (LICS'20),
Benjamin Böhm (ITCS'21, SAT'21), and
Benjamin Böhm and Tomáš Peitl (IJCAI'22, SAT'22)

Quantified Boolean Formulas (QBF)

- propositional logic + quantification
- Boolean quantifiers ranging over 0/1

Why QBF proof complexity?

- interesting theory
- driven by QBF solving – and (hopefully) informs QBF solving
- shows different effects from propositional proof complexity
- connects to circuit complexity, bounded arithmetic, ...

Interesting test case for algorithmic progress

SAT revolution

SAT	NP	main breakthrough late 90s
QBF	PSPACE	reaching industrial applicability now
DQBF	NEXPTIME	very early stage

A core QBF system: QU-Resolution

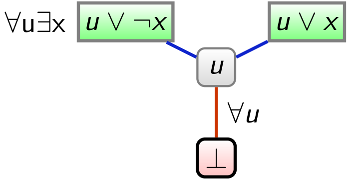
$=$ Resolution $+$ $\forall$-reduction [Kleine Büning et al. 95, V. Gelder 12]

Rules

- Resolution: $\dfrac{x \vee C \qquad \neg x \vee D}{C \vee D}$ ($C \vee D$ not tautological)
- $\forall$-Reduction: $\dfrac{C \vee u}{C}$ (u universally quantified)

  C does not contain variables right of u in the quantifier prefix.

Example



Proof complexity of QU-Resolution

By now quite good theoretical understanding of QU-Resolution

- lower bounds for
  - various handcrafted QBFs
  - random QBFs
- different lower bound techniques:
  - semantic size-cost-capacity technique (deriving proof-size lower bounds from the size of countermodels)
  - transfer of circuit lower bounds to proof-size bounds
  - size-width technique (different from [Ben-Sasson & Wigderson 2001])
  - characterisation of QU-Resolution size by a simple circuit model (UDL = unified decision lists)

Unified decision lists

Our circuit model

- natural multi-output generalisation of decision lists [Rivest 87]
- computes functions $\{0,1\}^n \to \{0,1\}^m$
- input variables $x_1, \ldots, x_n$
- output variables $u_1, \ldots, u_m$

$$
\begin{array}{l}
\textsc{If } t_1 \textsc{ Then } \vec{u} = \vec{b}_1 \\
\textsc{Else If } t_2 \textsc{ Then } \vec{u} = \vec{b}_2 \\
\qquad\qquad\vdots \\
\textsc{Else If } t_k \textsc{ Then } \vec{u} = \vec{b}_k \\
\textsc{Else } \vec{u} = \vec{b}_{k+1}
\end{array}
$$

- t_i are terms in $x_1, \ldots, x_n$
- $\vec{b}_i$ are total assignments to $u_1, \ldots, u_m$

We call this model unified decision lists (UDL).

Unified decision lists

Unified decision lists (UDLs)

- naturally compute countermodels for false QBFs.
- Let $\Phi(\vec{x}, \vec{u})$ be a QBF with existential variables $\vec{x}$ and universal variables $\vec{u}$.
- Let T be a UDL with inputs $\vec{x}$ and outputs $\vec{u}$.
- We call T a UDL for Φ if for each assignment α to $\vec{x}$, the UDL T computes an assignment $T(\alpha)$ such that $\alpha \cup T(\alpha)$ falsifies Φ.
- The UDL needs to respect the quantifier dependencies of Φ, e.g. in $\exists x_1 \forall u_1 \exists x_2$ the value of u_1 must only depend on x_1.

Hardness characterisation

Result (informally)

If each countermodel of Φ is hard to compute for UDLs,
then Φ requires long proofs in *QU-Res*.

Theorem (more formally)

- Let Φ be a false QBF of bounded quantifier complexity.
- Then the size of the smallest $\mathit{QU\text{-}Res}^{NP}$ refutation of Φ is polynomially related to the size of the smallest UDL for Φ.

Alternative characterisation

A sequence Φ_n of bounded quantification is hard for *QU-Res* iff

1. Φ_n require large UDLs, or
2. Φ_n contain propositional resolution hardness.

The propositional hardness in 2. can be precisely identified.

Hard QBFs: first example

Parity formulas

$$\mathrm{QP\small{ARITY}}_n = \exists x_1 \cdots x_n \forall u \exists t_1 \cdots t_n$$
$$\{x_1 \leftrightarrow t_1\} \cup \bigcup_{i=2}^{n} \{(t_{i-1} \oplus x_i) \leftrightarrow t_i\} \cup \{u \not\leftrightarrow t_n\}$$

- The only winning strategy is to compute $u = x_1 \oplus \ldots \oplus x_n$.

Hardness for QU-Res

- easy to see: the first line of each UDL for $\mathrm{QP\small{ARITY}}_n$ requires all existential variables $x_1, \ldots, x_n$
- size-width result immediately yields a lower bound of $2^{\Omega(n)}$

Hard QBFs: second example

Equality formulas

$$EQ_n = \exists x_1 \cdots x_n \forall u_1 \cdots u_n \exists t_1 \cdots t_n \left(\bigwedge_{i=1}^{n} (x_i \vee u_i \vee \neg t_i) \wedge (\neg x_i \vee \neg u_i \vee \neg t_i) \right) \wedge \left(\bigvee_{i=1}^{n} t_i \right)$$

- The only winning strategy is to compute $u_i = x_i$ for $i \in [n]$.

Hardness for QU-Res

- easy to see: the first line of each UDL for EQ_n requires all existential variables $x_1, \ldots, x_n$
- yields exponential lower bound

Intermediate summary: QBF resolution

- Tight characterisation of QBF resolution hardness by circuit complexity (UDLs)
- UDLs are a natural computational model to compute QBF countermodels.
- yields size-width relation for QBF, but different dependence than in [Ben-Sasson & Wigderson 2001]
- allows to elegantly (re)prove many lower bounds

Now:

- Relation between QBF resolution and QCDCL solving

CDCL Pseudocode

```
CDCL(F)
L ← 0; α ← empty assignment
Loop
   extend α by unit propagation as long as possible
   IF α satisfies F THEN return α
   IF α falsifies a clause of F THEN
      IF L = 0 THEN return unsatisfiable
      learn one or more clauses and add them to F
      choose backjumping level L′ < L
      delete all assignments of literals on levels > L′
      L ← L′
   ELSE
      choose an unassigned literal x
      extend α by x = 0 (or x = 1)
      L ← L + 1
   ENDIF
ENDLOOP
```

QCDCL

CDCL can be lifted to QBF [Zhang & Malik 2002]

CDCL $\Rightarrow$ QCDCL: crucial differences

- selection of decision variables follows the order of the prefix
- unit propagation also incorporates universal reduction

Unit propagation in CDCL

- $x \vee \bar{y} \vee z$ becomes unit clause z under $x = 0, y = 1$

Unit propagation in QCDCL

- assume prefix $\exists x \forall u \exists y$
- $x \vee u \vee y$ becomes unit clause x under $y = 0$

(Q)CDCL trails

Partial assignments in (Q)CDCL are represented as trails.

A CDCL trail

- is a sequence of literals that represents a CDCL run between two backtracking steps.
- takes the form

$$(p_{(0,1)}, \ldots, p_{(0,g_0)}; \boldsymbol{d}_1, p_{(1,1)}, \ldots, p_{(1,g_1)}; \ldots; \boldsymbol{d}_{\boldsymbol{r}}, p_{(r,1)}, \ldots, p_{(r,g_r)})$$

- $d_1, \ldots, d_r$ are the decision literals.
- $p_{(i,j)}$ are literals propagated by unit propagation.
- works analogously for QCDCL

(Q)CDCL proofs

- In CDCL learned clauses are derived by resolution.
- In QCDCL learned clauses are derived by long-distance Q-resolution (LDQ-Resolution), an extension of Q-Resolution.

(Q)CDCL as a formal proof system

- A CDCL proof of F has the form

$$((\mathcal{T}_1, \ldots, \mathcal{T}_m), (C_1, \ldots, C_m), (\pi_1, \ldots, \pi_m)).$$

- $\mathcal{T}_1, \ldots, \mathcal{T}_m$ are CDCL trails.
- C_i is the clause learned after the conflict in trail $\mathcal{T}_i$.
- π_i is a resolution derivation of C_i from $F \cup \{C_1, \ldots, C_{i-1}\}$.
- In QCDCL, π_i is a LDQ-Resolution proof.

SAT/QBF solvers and proof systems

- Construct resolution refutations from CDCL runs on unsatisfiable formulas

$$\begin{tikzcd} \boxed{\text{CDCL}} \arrow[r, "\text{extract}"] & \boxed{\text{Resolution}} \\ \boxed{\text{QCDCL}} \arrow[r, "\text{extract}"] & \boxed{\text{LDQ-Resolution}} \end{tikzcd}$$

- From a CDCL proof

$$((\mathcal{T}_1, \ldots, \mathcal{T}_m), (C_1, \ldots, C_m), (\pi_1, \ldots, \pi_m))$$

extract a Resolution proof of C_m by sticking together the subproofs $\pi_1, \ldots, \pi_m$.

- analogously for QCDCL and LDQ-Resolution

Solvers and proof systems

$$\text{CDCL} \xrightarrow{\text{extract}} \text{Resolution}$$

$$\text{QCDCL} \xrightarrow{\text{extract}} \text{LDQ-Resolution}$$

Question

What about the converse directions?

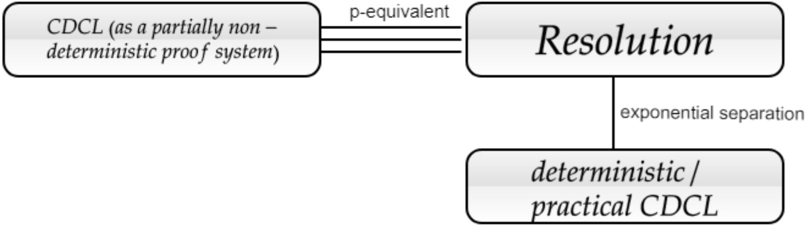
Theorem [Pipatsrisawat & Darwiche 2011][Atserias, Fichte & Thurley 2011]

For each Resolution refutation π of a formula ϕ in n variables there is a CDCL run of size $\mathcal{O}(n^4|\pi|)$ that refutes ϕ.

- Hence CDCL and Resolution are p-equivalent.
- But: The CDCL model contains non-deterministic elements (e.g., decisions depend on refutation).

SAT solvers and proof systems

- In practise, CDCL uses deterministic procedures for decision making, clause learning, etc.
- Practical CDCL is exponentially weaker than Resolution. [Vinyals 2020]



- What happens for QBF?

QBF Solvers and proof systems

Theorem [Janota 2016]

Deterministic/practical QCDCL is exponentially weaker than Q-Resolution (demonstrated for QBFs CR_n).

Question

Does QCDCL (as a non-deterministic proof system) simulate Q-Resolution?

Theorem

QCDCL and Q-Resolution are incomparable.
There exist exponential separations in both directions.

Separating QCDCL and Q-Resolution (1)

Parity formulas

$$\begin{aligned}\textsc{QParity}_n &= \exists x_1 \cdots x_n \forall u \exists t_1 \cdots t_n \\ &\quad \{x_1 \leftrightarrow t_1\} \cup \bigcup_{i=2}^{n} \{(t_{i-1} \oplus x_i) \leftrightarrow t_i\} \cup \{u \not\leftrightarrow t_n\}\end{aligned}$$

Theorem

- $\textsc{QParity}_n$ is exponentially hard for Q-Resolution.
- There exist polynomial-size QCDCL refutations of $\textsc{QParity}_n$.

Separating QCDCL and Q-Resolution (2)

Trapdoor formulas

Let $\texttt{Trapdoor}_n$ be the QBF

$$\exists y_1, \ldots, y_{s_n} \forall w \exists t, x_1, \ldots, x_{s_n} \forall u$$

$$\mathrm{PHP}_n^{n+1}(x_1, \ldots, x_{s_n}) \wedge$$

$$\bigwedge_{i \in [s_n]} ((\bar{y}_i \vee x_i \vee u) \wedge (y_i \vee \bar{x}_i \vee u)$$

$$(y_i \vee w \vee t) \wedge (y_i \vee w \vee \bar{t}) \wedge (\bar{y}_i \vee w \vee t) \wedge (\bar{y}_i \vee w \vee \bar{t}))$$

- $\texttt{Trapdoor}_n$ needs exponential-size QCDCL refutations.
- There are constant-size Q-Resolution refutations of $\texttt{Trapdoor}_n$.

Trapdoor is hard for QCDCL

$$\exists y_1, \dots, y_{s_n} \forall w \exists t, x_1, \dots, x_{s_n} \forall u$$
$$\mathrm{PHP}_n^{n+1}(x_1, \dots, x_{s_n}) \wedge$$
$$\bigwedge_{i \in [s_n]} ((\bar{y}_i \vee x_i \vee u) \wedge (y_i \vee \bar{x}_i \vee u)$$
$$(y_i \vee w \vee t) \wedge (y_i \vee w \vee \bar{t}) \wedge (\bar{y}_i \vee w \vee t) \wedge (\bar{y}_i \vee w \vee \bar{t}))$$

- In QCDCL, variables are decided in prefix order.
- Hence each trail starts with the y variables.
- Unit propagation (with universal reduction) enforces $x_i = y_i$.
- Therefore the trail runs into a conflict on the PHP clauses.
- This happens repeatedly, generating a refutation of PHP.
- Clauses in the last line have trivial Q-Resolution refutations.
- $\Rightarrow$ constant-size Q-Resolution refutations of $\texttt{Trapdoor}_n$

Trapdoor is hard for QCDCL

$$\exists y_1, \ldots, y_{s_n} \forall w \exists t, x_1, \ldots, x_{s_n} \forall u$$
$$\mathrm{PHP}_n^{n+1}(x_1, \ldots, x_{s_n}) \wedge$$
$$\bigwedge_{i \in [s_n]} ((\bar{y}_i \vee x_i \vee u) \wedge (y_i \vee \bar{x}_i \vee u)$$
$$(y_i \vee w \vee t) \wedge (y_i \vee w \vee \bar{t}) \wedge (\bar{y}_i \vee w \vee t) \wedge (\bar{y}_i \vee w \vee \bar{t}))$$

- Separation can also be shown on other formulas, where no propositional hardness is present, e.g. CR_n.

What else is hard in QCDCL?

Equality formulas

$$
\begin{aligned}
EQ_n \quad = \quad & \exists x_1 \cdots x_n \forall u_1 \cdots u_n \exists t_1 \cdots t_n \\
& \left(\bigwedge_{i=1}^{n} (x_i \vee u_i \vee \neg t_i) \wedge (\neg x_i \vee \neg u_i \vee \neg t_i) \right) \wedge \left(\bigvee_{i=1}^{n} t_i \right)
\end{aligned}
$$

- The only winning strategy is to compute $u_i = x_i$ for $i \in [n]$.

Theorem

- EQ_n is hard for Q-Resolution.
- Hardness lifts to QCDCL.

Different policies in QCDCL

Consider different policies for

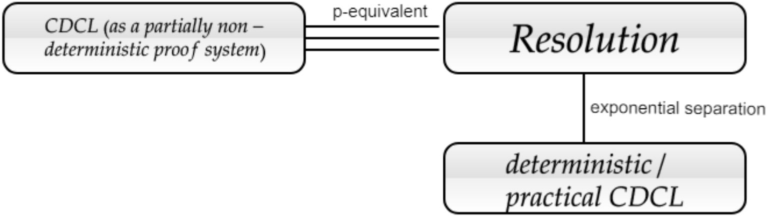
- unit propagation
  - use universal reduction in unit propagation (practical QCDCL)
  - just use plain unit propagation as in CDCL
- selection of decision literals
  - follow the order of the prefix (practical QCDCL)
  - relax this requirement (but still learn asserting clauses)
  - use arbitrary order

Theorem

- All combination of policies yield sound algorithms and QCDCL proof systems.
- Q-Resolution and $\text{QCDCL}^{\textsc{Any-Ord}}_{\textsc{No-Red}}$ are p-equivalent.
- For each Q-Resolution refutation π of a QBF Φ in n variables there is a $\text{QCDCL}^{\textsc{Any-Ord}}_{\textsc{No-Red}}$ refutation of size $\mathcal{O}(n^3|\pi|)$.

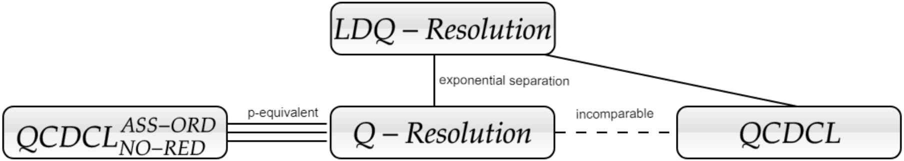
Summary: The SAT case

- Practical/deterministic CDCL is weaker than the underlying system Resolution.
- The non-deterministic CDCL model is equivalent to Resolution.



Summary: The QBF case

- more complex picture in QBF
- QCDCL (even in non-deterministic model) is incomparable to Q-Resolution.
- We design a new QCDCL model that is equivalent to Q-resolution.



Analysis of further QCDCL ingredients

Cube learning

- cube = term = conjunction of literals
- QCDCL not only learns clauses, but also cubes
- is needed for completeness: in case of a true QBF a cube refutation is computed

Cube refutation

- cube refutation = resolution on cubes
- two rules:
  - Resolution: $$\frac{x \wedge C \qquad \neg x \wedge D}{C \wedge D}$$
  - $\exists$-Reduction: $$\frac{C \wedge x}{C}$$ (x existentially quantified)

    C does not contain variables right of x in the quantifier prefix.

Analysis of further QCDCL ingredients

Cube learning - how is it done?

- Input: QBF with CNF matrix
- Start with the empty set of cubes.
- When the current trail satisfies the matrix, a cube is learned (consisting of a subset of literals on the trail that satisfy all clauses).
- Cubes are also used for unit propagation: a unit cube must be falsified.
- Cubes can also generate conflicts: if a cube is satisfied.
- In this case a cube is learned from the conflict (by cube resolution).

Is cube learning advantageous?

- It is needed for true QBFs.
- But can also affect the run time for false QBFs (because of additional unit propagations).
- Define $\mathsf{QCDCL}^{\mathsf{Cube}}$ as the (non-deterministic) proof system for false QBFs where prefix order is obeyed, but cube learning and propagation are enabled.

Observation

$\mathsf{QCDCL}^{\mathsf{Cube}}$ simulates QCDCL.

Is cube learning advantageous?

Observation

$\mathsf{QCDCL}^{\mathsf{Cube}}$ simulates QCDCL.

Theorem

$\mathsf{QCDCL}^{\mathsf{Cube}}$ is exponentially stronger than QCDCL.

Proof sketch

- EQ_n is exponentially hard for QCDCL.
- But has short refutations in $\mathsf{QCDCL}^{\mathsf{Cube}}$.
- Learning the right cubes enables out-of-order 'decisions'. $\square$

Another QCDCL technique: pure-literal elimination

- A variable is pure in Φ if it only occurs in one polarity.
- In QCDCL, if a variable becomes pure, then the corresponding literal is set to
  - true, if the variable is existential;
  - false, if the variable is universal.
- Pure-literal elimination (PLE) is included in some QCDCL solvers, e.g. DepQBF.

Question

Does pure-literal elimination help?

Answer

Sometimes.

Include PLE into proof systems

- Let $\text{QCDCL}^{\text{PLE}}$ be the model with pure-literal elimination enabled, but without Cube Learning.
- Let $\text{QCDCL}^{\text{Cube+PLE}}$ be the model with both enabled.

Theorem

QCDCL and $\text{QCDCL}^{\text{PLE}}$ are incomparable.

Proof

- EQ_n is exponentially hard for QCDCL.
- But has short refutations in $\text{QCDCL}^{\text{PLE}}$.
- Intuition: PLE can enable useful out-of-order 'decisions'.
- Construct other QBFs $PLE\text{-}trap_n$ (based on CR_n), which are easy for QCDCL, but hard for $\text{QCDCL}^{\text{PLE}}$.
- Intuition: PLE can force bad out-of-order decisions, leading to the hard trap. □

Include PLE into proof systems

- Let $\text{QCDCL}^{\text{PLE}}$ be the model with pure-literal elimination enabled, but without Cube Learning.
- Let $\text{QCDCL}^{\text{Cube+PLE}}$ be the model with both enabled.

Theorem
QCDCL and $\text{QCDCL}^{\text{PLE}}$ are incomparable.

Theorem
$\text{QCDCL}^{\text{Cube}}$ and $\text{QCDCL}^{\text{Cube+PLE}}$ are incomparable.

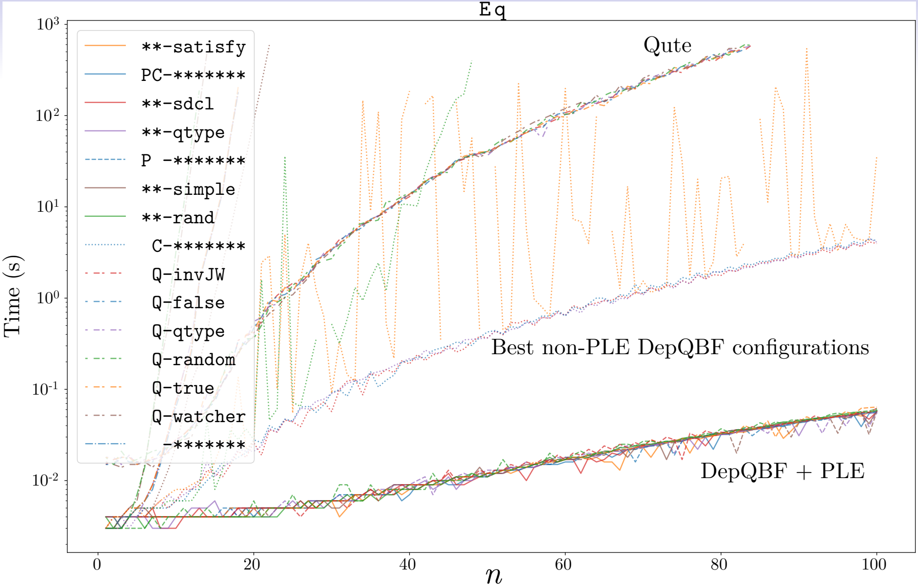
Different ingredients in (Q)CDCL

General question

- Which (Q)CDCL components are most influential for performance?
- important ingredients: decision heuristics, restarts, clause-learning schemes ...
- test case here: cube learning, pure-literal elimination
- not well understood from a theoretical perspective

Comparing CDCL and QCDCL

- almost no theoretical results known for CDCL ingredients
- analysis appears easier in QCDCL, because prefix imposes decision order in the most common QCDCL model.



Eq_n: easy for QCDCL$^{\mathrm{PLE}}$ and for QCDCL$^{\mathrm{Cube}}$, but hard for QCDCL (in proof complexity)

Do we need to follow prefix order in QCDCL?

Answer

- no, decisions can be made in any order
- prefix needs to be obeyed during clause learning
- It is no longer guaranteed that asserting clauses/cubes can be learnt.

We introduce two new QCDCL models

- $\text{QCDCL}^{\textit{uni-any}}$:
  - arbitrary universal decisions
  - an existential var x can be decided when all universal vars left of x are assigned
  - guarantees that asserting clauses can always be learnt
- $\text{QCDCL}^{\textit{exi-any}}$: dual model with arbitrary existential decisions

Separations

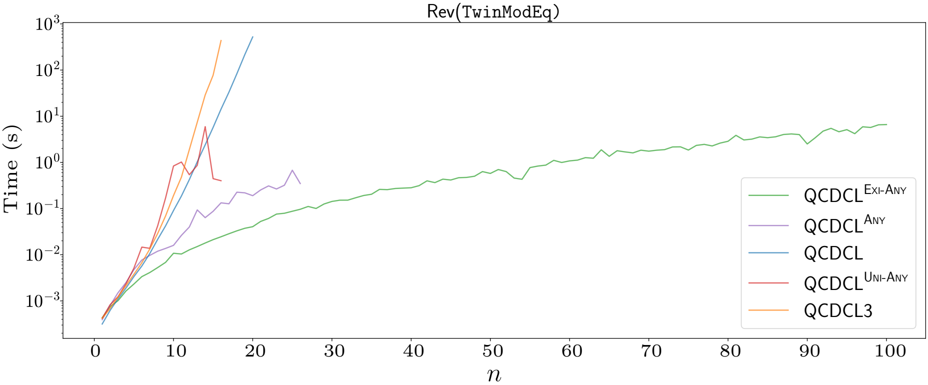
Theorem

- There exist true QBFs (variations of Eq_n) that are exponentially hard for QCDCL, but easy for $\mathrm{QCDCL}^{\textit{exi-any}}$.
- There exist false QBFs (variations of CR_n) that are exponentially hard for QCDCL, but easy for $\mathrm{QCDCL}^{\textit{uni-any}}$.

Interesting model

- should be further explored in practice
- no dedicated lower bounds known
  (except those existing for long-distance resolution)
- Difficulty: have to argue against more complex decision heuristics (as in SAT)

Some initial experiments

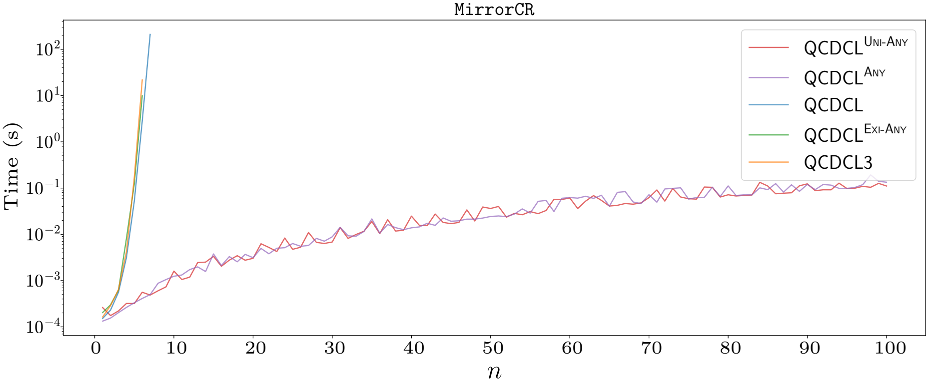


Running times on the true QBFs (variations of Eq_n) that are

- exponentially hard for QCDCL,
- but easy for QCDCL$^{exi\text{-}any}$

as shown with proof complexity.

Some initial experiments



Running times on the false QBFs (variations of CR_n) that are

- exponentially hard for QCDCL,
- but easy for QCDCL$^{\textit{uni-any}}$

as shown with proof complexity.

Conclusion

QCDCL vs Q-Resolution

- complex picture
- lower bounds somewhat more accessible than in SAT
- incomparable heuristics

What is the best QCDCL model?

- promising models:
  - QCDCL$^{\textit{exi-any}}$ (better for true QBFs)
  - QCDCL$^{\textit{uni-any}}$ (better for false QBFs)
- more theoretical + experimental research needed