


Understanding the Relative Strength of QBF CDCL Solvers and QBF Resolution

Olaf Beyersdorff 

Friedrich Schiller Universität Jena, Germany
olaf.beyersdorff@uni-jena.de

Benjamin Böhm 

Friedrich Schiller Universität Jena, Germany
benjamin.boehm@uni-jena.de

Abstract

QBF solvers implementing the QCDCL paradigm are powerful algorithms that successfully tackle many computationally complex applications. However, our theoretical understanding of the strength and limitations of these QCDCL solvers is very limited.

In this paper we suggest to formally model QCDCL solvers as proof systems. We define different policies that can be used for decision heuristics and unit propagation and give rise to a number of sound and complete QBF proof systems (and hence new QCDCL algorithms). With respect to the standard policies used in practical QCDCL solving, we show that the corresponding QCDCL proof system is incomparable (via exponential separations) to Q-resolution, the classical QBF resolution system used in the literature. This is in stark contrast to the propositional setting where CDCL and resolution are known to be p-equivalent.

This raises the question what formulas are hard for standard QCDCL, since Q-resolution lower bounds do not necessarily apply to QCDCL as we show here. In answer to this question we prove several lower bounds for QCDCL, including exponential lower bounds for a large class of random QBFs.

We also introduce a strengthening of the decision heuristic used in classical QCDCL, which does not necessarily decide variables in order of the prefix, but still allows to learn asserting clauses. We show that with this decision policy, QCDCL can be exponentially faster on some formulas.

We further exhibit a QCDCL proof system that is p-equivalent to Q-resolution. In comparison to classical QCDCL, this new QCDCL version adapts both decision and unit propagation policies.

2012 ACM Subject Classification Theory of computation → Proof complexity

Keywords and phrases CDCL, QBF, QCDCL, proof complexity, resolution, Q-resolution

Digital Object Identifier 10.4230/LIPIcs.ITCS.2021.12

Related Version This is an extended abstract. A full version of the paper is available at <https://eccc.weizmann.ac.il/report/2020/053/>.

Funding *Olaf Beyersdorff*: John Templeton Foundation (grant no. 60842), Carl Zeiss Foundation.

1 Introduction

SAT solving has revolutionised the way we perceive and approach computationally complex problems. While traditionally, NP-hard problems were considered computationally intractable, today SAT solvers routinely and successfully solve instances of NP-hard problems from virtually all application domains, and in particular problem instances of industrial relevance [53]. Starting with the classic DPLL algorithm from the 1960s [25, 26], there have been a number of milestones in the evolution of SAT solving, but clearly one of the breakthrough achievements was the introduction of clause learning in the late 1990s, leading to the paradigm of *conflict-driven clause learning* (CDCL) [43, 55], the predominant technique



© Olaf Beyersdorff and Benjamin Böhm;
licensed under Creative Commons License CC-BY
12th Innovations in Theoretical Computer Science Conference (ITCS 2021).

Editor: James R. Lee; Article No. 12; pp. 12:1–12:20



Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

of modern SAT solving. CDCL ingeniously combines a number of crucial ingredients, among them variable decision heuristics, unit propagation, clause learning from conflicts, and restarts (cf. [42] for an overview).

Inspired by the success of SAT solving, many researchers have concentrated on the task to extend the reach of these technologies to computationally even more challenging settings with *quantified Boolean formulas* (QBF) receiving key attention. As a PSPACE-complete problem, the satisfiability problem for QBFs encompasses all problems from the polynomial hierarchy and allows to encode many problems far more succinctly than in propositional logic (cf. [51] for applications).

One of the main techniques in QBF solving is the propositional CDCL technique, lifted to QBF in the form of QCDCL [56]. However, solving QBFs presents additional challenges as the quantifier type of variables (existential and universal) needs to be taken into account as well as the variable dependencies stemming from the quantifier prefix.¹ This particularly impacts the variable selection heuristics and details of the unit propagation within QCDCL. In addition to QCDCL there are further QBF solving techniques, exploiting QBF features absent in SAT, such as expanding universal variables in expansion solving [36] and dependency schemes in dependency-aware solving [40, 47, 52]. Compared to SAT solving, QBF solving is still at an earlier stage. However, QBF solving has seen huge improvements during the past 15 years [49], and there are problems of practical relevance where QBF solvers outperform SAT solvers [28].

The enormous success of SAT and QBF solving of course raises theoretical questions of utmost importance: why are these solvers so successful and what are their limitations? The main approach through understanding these questions comes from proof complexity [20, 46]. The central problem in proof complexity is to determine the size of the smallest proof for a given formula in a specified proof system, typically defined through a set of axioms and inference rules. Traces of runs of SAT/QBF solvers on unsatisfiable instances yield proofs of unsatisfiability, whereby each solver implicitly defines a proof system. In particular, SAT solvers implementing the DPLL and CDCL paradigms are based on resolution [46], which is arguably the most studied proof system in proof complexity.

Propositional resolution operates on clauses and uses the resolution rule

$$\frac{C \vee x \quad D \vee \bar{x}}{C \vee D} \quad (1)$$

as its only inference rule to derive a new clause $C \vee D$ from the two parent clauses $C \vee x$ and $D \vee \bar{x}$.² There is a host of lower bounds and lower bound techniques available for propositional resolution (cf. [5, 38, 50] for surveys).

While it is relatively easy to see that the classic DPLL branching algorithm [25, 26] exactly corresponds to tree-like resolution (where resolution derivations are in form of a tree), the *relation between CDCL and resolution* is far more complex. On the one hand, resolution proofs can be generated efficiently from traces of CDCL runs on unsatisfiable formulas [4], a crucial observation being that learned clauses are derivable by resolution [4, 43]. The opposite simulation is considerably more difficult, with a series of works [1, 4, 33, 48] culminating in the result that CDCL can efficiently simulate arbitrary resolution proofs, i.e., resolution and CDCL are equivalent. This directly implies that all known lower bounds for proof size in resolution translate into lower bounds for CDCL running time. In addition, other measures such as proof space model memory requirements of SAT solvers, thereby implying lower bounds on memory consumption, in particular when considering time-space tradeoffs [45].

¹ In this paper we focus on prenex QBFs with a CNF matrix.

² We denote such a resolution inference with pivot x by $(C \vee x) \stackrel{x}{\bowtie} (D \vee \bar{x})$ throughout the paper.

Exciting as this equivalence between CDCL and resolution is from a theoretical point of view, it has to be interpreted with care. Proof systems are inherently non-deterministic procedures, while CDCL algorithms are largely deterministic (some randomisation might occasionally be used). To overcome this discrepancy, the simulations of resolution by CDCL [4, 48] use arbitrary decision heuristics and perform excessive restarts, both of which diverge from practical CDCL policies. Indeed, in very recent work [54] it was shown that CDCL with practical decision heuristics such as VSIDS [55] is exponentially weaker than resolution, and similar results have been obtained for further decision heuristics [44]. Regarding restarts there is intense research aiming to determine the power of CDCL without restarts from a proof complexity perspective (cf. [19, 21]).

On the QBF level, this naturally raises the question *what proof system corresponds to QCDCL*. As in propositional proof complexity, QBF resolution systems take a prominent place in the QBF proof system landscape, with the basic and historically first Q-resolution system [37] receiving key attention. Q-resolution is a refutational system that proves the falsity of fully quantified prenex QBFs with a CNF matrix (QCNFs). The system allows to use the propositional resolution rule (1) under the conditions that the pivot x is an existential variable and the resolvent $C \vee D$ is non-tautological. In addition, Q-resolution uses a *universal reduction rule*

$$\frac{C \vee u}{C}, \quad (2)$$

where u is a universal literal that in the quantifier prefix is quantified right of all variables in C , i.e., none of the literals in C depends on u . For Q-resolution we have a number of lower bounds [3, 8, 12] as well as lower bound techniques, some of them lifted from propositional proof complexity [13, 15], but more interestingly some of them genuine to the QBF domain [8, 10] that unveil deep connections between proof size and circuit complexity [11, 16], unparalleled in the propositional domain.

Unlike in the relation between SAT and CDCL, it has been open whether QCDCL runs can be efficiently translated into Q-resolution. Instead, QCDCL runs can be simulated by the stronger QBF resolution system of long-distance Q-resolution [2, 56]. In fact, this system originates from solving, where it was noted that clauses learned from QCDCL conflicts can be derived in long-distance Q-resolution [56]. Long-distance Q-resolution implements a more liberal use of the resolution rule (1), which allows to derive certain tautologies. In general, allowing to derive tautologies with (1) is unsound. However, the tautologies allowed in long-distance Q-resolution do not present problems for soundness and are exactly those clauses needed when learning clauses in QCDCL. Hence long-distance Q-resolution simulates QCDCL [2, 56]. However, it is known that long-distance Q-resolution allows exponentially shorter proofs than Q-resolution for some QBFs [8, 9, 27].

We also remark that there are further QBF resolution systems (cf. [18] for an overview) and even stronger QBF calculi [11, 14, 23, 34]. Some of these correspond to other solving approaches in QBF, such as the system $\forall\text{Exp}+\text{Res}$ [36] that captures expansion QBF solving [6].

In summary, it is fair to say that the relations between QCDCL solving and QBF resolution (either Q-resolution or long-distance Q-resolution) are *currently not well understood*. In particular, an analogue of the equivalence of CDCL SAT solving and propositional resolution [1, 4, 48] is currently absent in the QBF domain. This brings us to the topic of this paper. However, rather than giving an overview of our results in this introduction, we will describe our results in Sections 3 to 7, after stating some preliminaries in Section 2. Most proofs will be omitted in this extended abstract due to space constraints.

2 Preliminaries

2.1 Propositional and quantified formulas

We will consider propositional and quantified formulas over a countable set of variables. Variables and negations of variables are called *literals*, i.e., for a variable x we can form two literals: x and its negation \bar{x} . Sometimes we write x^1 instead of x and x^0 instead of \bar{x} . We denote the corresponding variable as $\text{var}(x) := \text{var}(\bar{x}) := x$.

A *clause* is a disjunction $\ell_1 \vee \dots \vee \ell_m$ of some literals ℓ_1, \dots, ℓ_m . We will sometimes view a clause as a set of literals, i.e., we will use the notation $\ell \in C$ if the literal ℓ is one of the literals in the clause C . If $m = 1$, we will often write (ℓ_1) to emphasize the difference between literals and clauses. The *empty clause* is the clause consisting of zero literals, denoted by (\perp) . For reasons of consistency it is helpful to define an *empty literal*, denoted by \perp in our case. As a consequence, we have $\perp \in (\perp)$, although we define the empty clause as a clause with zero literals.

The negation of a clause $C = \ell_1 \vee \dots \vee \ell_m$ is called a *term*, i.e., terms are conjunctions $\bar{\ell}_1 \wedge \dots \wedge \bar{\ell}_m$ of literals. Similarly terms can be considered as sets of literals. A *CNF* (*conjunctive normal form*) is a conjunction of clauses.

Let $C = \ell_1 \vee \dots \vee \ell_m$. We define $\text{var}(C) := \{\text{var}(\ell_1), \dots, \text{var}(\ell_m)\}$. For a CNF $\phi = C_1 \wedge \dots \wedge C_n$ we define $\text{var}(\phi) := \bigcup_{i=1}^n \text{var}(C_i)$.

A clause or a set C of literals is called *tautological*, if there is a variable x with $x, \bar{x} \in C$.

An *assignment* σ of a set of variables X is a non-tautological set of literals, such that for all $x \in X$ there is $\ell \in \sigma$ with $\text{var}(\ell) = x$. The restriction of a clause C by an assignment σ is defined as

$$C|_\sigma := \begin{cases} \top \text{ (true)} & \text{if } C \cap \sigma \neq \emptyset, \\ \bigvee_{\substack{\ell \in C \\ \ell \notin \sigma}} \ell & \text{otherwise.} \end{cases}$$

For example, let $C = t \vee x \vee y \vee \bar{z}$ and define the assignment $\sigma := \{\bar{x}, z, w\}$. Then we have $C|_\sigma = t \vee y$. Note that the set of assigned variables might differ from $\text{var}(C)$. In our case, σ is an assignment of the set $X := \{x, z, w\}$.

One can interpret σ as an operator that sets all literals from σ to the Boolean constant 1. We denote the set of assignments of X by $\langle X \rangle$. A CNF ϕ *entails* another CNF ψ if each assignment that satisfies ϕ also satisfies ψ (denoted by $\phi \models \psi$).

A *QBF* (*quantified Boolean formula*) $\Phi = \mathcal{Q} \cdot \phi$ is a propositional formula ϕ (also called *matrix*) together with a *prefix* \mathcal{Q} . A prefix $Q_1 x_1 Q_2 x_2 \dots Q_k x_k$ consists of variables x_1, \dots, x_k and quantifiers $Q_1, \dots, Q_k \in \{\exists, \forall\}$. We obtain an equivalent formula if we unite adjacent quantifiers of the same type. Therefore we can always assume the prefix to be in the form

$$\mathcal{Q} = Q'_1 X_1 Q'_2 X_2 \dots Q'_s X_s$$

with nonempty sets of variables X_1, \dots, X_s and quantifiers $Q'_1, \dots, Q'_s \in \{\exists, \forall\}$ such that $Q'_i \neq Q'_{i+1}$ for $i \in [s-1]$. For a variable x in \mathcal{Q} we denote the *quantifier level* with respect to \mathcal{Q} by $\text{lv}(x) = \text{lv}_\Phi(x) = i$, if $x \in X_i$. Variables from Φ are called *existential*, if the corresponding quantifier is \exists , and *universal* if the quantifier is \forall . We denote the set of existential variables from Φ by $\text{var}_\exists(\Phi)$, and the set of universal variables by $\text{var}_\forall(\Phi)$.

A QBF with CNF matrix is called a *QCNF*. We require that all clauses from a matrix of a QCNF are non-tautological, otherwise we just delete these clauses. This requirement is crucial for the correctness of the derivation rules we define later for QBF proof systems. Since we will only discuss refutational proof systems, we will always assume that all QCNFs we consider are false.

A QBF can be interpreted as a game between two players: The \exists -player and the \forall -player. These players have to assign the respective variables one by one along the quantifier order from left to right. The \forall -player wins the game if and only if the matrix of the QBF gets falsified by this assignment. It is well known that for every false QBF $\Phi = \mathcal{Q} \cdot \phi$ there exists a winning strategy for the \forall -player.

2.2 Q-resolution and long-distance Q-resolution

Let C_1 and C_2 be two clauses of a QCNF Φ and let ℓ be an existential literal with $\text{var}(\ell) \notin \text{var}(C_1) \cup \text{var}(C_2)$. The *resolvent* of $C_1 \vee \ell$ and $C_2 \vee \bar{\ell}$ over ℓ is defined as

$$(C_1 \vee \ell) \overset{\ell}{\bowtie} (C_2 \vee \bar{\ell}) := C_1 \vee C_2.$$

Let $C := u_1 \vee \dots \vee u_m \vee x_1 \vee \dots \vee x_n \vee v_1 \vee \dots \vee v_s$ be a clause from Φ , where $u_1, \dots, u_m, v_1, \dots, v_s$ are universal literals, x_1, \dots, x_n are existential literals and

$$\{v \in C : v \text{ is universal and } \text{lv}(v) > \text{lv}(x_i) \text{ for all } i \in [n]\} = \{v_1, \dots, v_s\}.$$

Then we can perform a *reduction* step and obtain

$$\text{red}(C) := u_1 \vee \dots \vee u_m \vee x_1 \vee \dots \vee x_n.$$

For a CNF $\phi = \{C_1, \dots, C_k\}$ we define $\text{red}(\phi) := \{\text{red}(C_1), \dots, \text{red}(C_k)\}$.

Q-resolution [37] is a refutational proof system for false QCNFs. A Q-resolution proof π of a clause C from a QCNF $\Phi = \mathcal{Q} \cdot \phi$ is a sequence of clauses $\pi = C_1, \dots, C_m$ with $C_m = C$. Each C_i has to be derived by one of the following three rules:

- *Axiom*: $C_i \in \phi$;
- *Resolution*: $C_i = C_j \overset{x}{\bowtie} C_k$ for some $j, k < i$ and $x \in \text{var}_{\exists}(\Phi)$, and C_i is non-tautological;
- *Reduction*: $C_i = \text{red}(C_j)$ for some $j < i$.

Note that none of our axioms are tautological by definition. A *refutation* of a QCNF Φ is a proof of the empty clause (\perp) .

For the simulating QCDCL runs, long-distance Q-resolution was introduced in [2, 56]. This extension of Q-resolution allows to derive universal tautologies under certain conditions. As in Q-resolution, there are three rules by which a clause C_i can be derived. The axiom and reduction rules are identical to Q-resolution, but the resolution rule is changed to

- *Resolution (long-distance)*: $C_i = C_j \overset{x}{\bowtie} C_k$ for some $j, k < i$ and $x \in \text{var}_{\exists}(\Phi)$. The resolvent C_i is allowed to contain a tautology $u \vee \bar{u}$ if u is a universal variable. If $u \in \text{var}(C_j) \cap \text{var}(C_k)$, then we additionally require $\text{lv}(u) > \text{lv}(x)$.

Note that a long-distance Q-resolution proof without tautologies is just a Q-resolution proof.

Creating universal tautologies without any assumptions is unsound in general. For example, consider the true QCNF $\Psi := \forall u \exists x \cdot (u \vee \bar{x}) \wedge (\bar{u} \vee x)$. There is a winning strategy for the \exists -player by assigning x equal to u . Hence, the step $\text{red}\left((u \vee \bar{x}) \overset{x}{\bowtie} (\bar{u} \vee x)\right) = (\perp)$ is unsound since we resolved over an existential literal x with $\text{lv}_{\Psi}(x) > \text{lv}_{\Psi}(u)$ while generating $u \vee \bar{u}$.

3 Our framework: versions of QCDCL as formal proof systems

We now start to describe the framework for our results. Technically, this paper hinges on the formalisation of QCDCL solving as precisely defined proof systems, which can subsequently be analysed from a proof-complexity perspective. For this we need to formally define central ingredients of QCDCL solving, including trails, decision policies, unit propagation, and clause learning (cf. [18] for background). For decisions and unit propagation we will consider different policies: those corresponding to QCDCL solving in practice and new policies, yet unexplored. We will show that the corresponding QCDCL proof systems are all sound and complete.

We start with defining trails, decisions, unit propagations and our collection of policies.

► **Definition 1** (trails and policies for decision/unit propagation). *Let $\Phi = \mathcal{Q} \cdot \phi$ be a QCNF in n variables. A trail \mathcal{T} for Φ is a sequence of literals (or \perp) of variables from Φ with specific properties. We distinguish two types of literals in \mathcal{T} : decision literals, that can be both existential and universal, and propagated literals, that are either existential or \perp . Most of the time we write a trail \mathcal{T} as*

$$\mathcal{T} = (p_{(0,1)}, \dots, p_{(0,g_0)}; \mathbf{d}_1, p_{(1,1)}, \dots, p_{(1,g_1)}; \dots; \mathbf{d}_r, p_{(r,1)}, \dots, p_{(r,g_r)}).$$

We typically denote decision literals by d_i and propagated literals by $p_{(i,j)}$. To emphasize decisions, we will set decision literals in the trail in **boldface** and put a semicolon at the end of each decision level. The literal $p_{(i,j)}$ represents the j^{th} propagated literal at the i^{th} decision level, determined by the corresponding decision d_i . The decision level 0 is the only level without a decision literal. Similarly as with clauses, we can view \mathcal{T} as a set of literals or as an assignment and use the notation $x \in \mathcal{T}$ if the literal x is contained in \mathcal{T} .

Let $s \in \{0, \dots, r\}$ and $t \in \{0, \dots, g_s\}$. The subtrail of \mathcal{T} at time (s, t) is the trail consisting of all literals from the leftmost literal in \mathcal{T} up to (and including) $p_{(s,t)}$, if $t \neq 0$, or d_s otherwise. We denote this subtrail by $\mathcal{T}[s, t]$. The subtrail $\mathcal{T}[0, 0]$ is the empty trail.

We impose some further requirements for \mathcal{T} to be a trail for a QCNF Φ . The decisions have to be non-tautological and non-repeating, i.e., we require $\text{var}(d_i) \neq \text{var}(d_k)$ for each $i \neq k \in \{0, \dots, r\}$. If $\perp \in \mathcal{T}$, then this must be the last (rightmost) literal in \mathcal{T} . In this case we say that \mathcal{T} has run into a conflict.

We define four policies, concerning the decision of literals, from which we can choose exactly one at a time:

- **LEV-ORD** - For each $d_i \in \mathcal{T}$ we have $lv(d_i) \leq lv(x)$ for all $x \in \text{var}(\phi) \setminus \text{var}(\mathcal{T}[i-1, g_{i-1}])$. This means that we have to decide the variables along the quantification order.
- **ASS-ORD** - We can decide a literal d_k if it is existential, or if it is universal and $lv(d_1) \leq \dots \leq lv(d_k)$.
- **ASS-R-ORD** - We can only decide an existential variable x next, if and only if we already decided all universal variables u with $lv(u) < lv(x)$ before.
- **ANY-ORD** - We can choose any remaining literal as the next decision.

We define two more policies concerning unit propagation. Again, we have to choose exactly one:

- **RED** - For each $p_{(i,j)} \in \mathcal{T}$ there has to be a clause $C \in \phi$ such that $\text{red}(C|_{\mathcal{T}[i,j-1]}) = (p_{(i,j)})$.
- **NO-RED** - For each $p_{(i,j)} \in \mathcal{T}$ there has to be a clause $C \in \phi$ with $C|_{\mathcal{T}[i,j-1]} = (p_{(i,j)})$.

These clauses C as described in the unit-propagation policies are called antecedent clauses and will be denoted by $\text{ante}_{\mathcal{T}}(p_{(i,j)}) := C$. There could be more than one such suitable clause, in which case we will just choose one of them arbitrarily. The antecedent clauses clearly depend on the unit propagation policy we use.

The size of a trail \mathcal{T} is measured by $|\mathcal{T}|$ (i.e., the cardinality of \mathcal{T} as a set). Because each trail can at most contain all variables, we have $|\mathcal{T}| \in \mathcal{O}(n)$.

We remark that QCDCL as used in practice employs the policies LEV-ORD and RED, and the decision policy ANY-ORD originates from CDCL.

The policies RED and NO-RED determine the notion of unit clauses, which are important for unit propagation.

► **Definition 2** (unit clauses). *Let C be a clause. In the policy RED, we call C a unit clause if $\text{red}(C) = (x)$ for an existential literal x or $x = \perp$. Otherwise, for NO-RED, we call C a unit clause if $C = (x)$ for an existential literal x or $x = \perp$.*

Note that (u) is not a unit clause under the policy NO-RED for a universal literal u .

In (Q)CDCL, whenever a trail \mathcal{T} runs into a conflict, i.e., a clause C from Φ is falsified, we perform conflict analysis in the form of *clause learning*. This results in a clause D that follows from Φ and describes a reason for the conflict. Such conflict clauses are obtained by performing resolution (for CDCL) and long-distance Q-resolution (for QCDCL), starting from the conflict clause C and resolving along the propagated variables in \mathcal{T} in reverse order (skipping resolution steps when the pivot is missing).

► **Definition 3** (learnable clauses). *Let $\Phi = \mathcal{Q} \cdot \phi$ be a QCNF and let*

$$\mathcal{T} = (p_{(0,1)}, \dots, p_{(0,g_0)}; \mathbf{d}_1, p_{(1,1)}, \dots, p_{(1,g_1)}; \dots; \mathbf{d}_r, p_{(r,1)}, \dots, p_{(r,g_r)})$$

be a trail with $p_{(r,g_r)} = \perp$ that follows policies $P \in \{\text{LEV-ORD}, \text{ASS-ORD}, \text{ASS-R-ORD}, \text{ANY-ORD}\}$ and $R \in \{\text{RED}, \text{NO-RED}\}$. We call a clause learnable from \mathcal{T} if it appears in the sequence

$$\mathcal{L}_{\mathcal{T}} := (C_{(r,g_r)}, \dots, C_{(r,1)}, \dots, C_{(1,g_1)}, \dots, C_{(1,1)}, C_{(0,g_0)}, \dots, C_{(0,1)})$$

where $C_{(r,g_r)} := \text{red}(\text{ante}(p_{(r,g_r)}))$,

$$C_{(i,j)} := \begin{cases} \text{red}\left(C_{(i,j+1)} \stackrel{p_{(i,j)}}{\boxtimes} \text{red}(\text{ante}(p_{(i,j)}))\right) & \text{if } \bar{p}_{(i,j)} \in C_{(i,j+1)}, \\ C_{(i,j+1)} & \text{otherwise} \end{cases}$$

for $i \in \{0, \dots, r\}$, $j \in [g_i - 1]$, and

$$C_{(i,g_i)} := \begin{cases} \text{red}\left(C_{(i+1,1)} \stackrel{p_{(i,g_i)}}{\boxtimes} \text{red}(\text{ante}(p_{(i,g_i)}))\right) & \text{if } \bar{p}_{(i,g_i)} \in C_{(i+1,1)}, \\ C_{(i+1,1)} & \text{otherwise} \end{cases}$$

for $i \in \{0, \dots, r-1\}$.

Note that clause learning works independently from the used policy. Even if we choose the policy NO-RED, we might have to make reduction steps in this process. After the construction of each trail \mathcal{T} we will choose to learn exactly one clause from $\mathcal{L}_{\mathcal{T}}$. The actual choice represents a kind of nondeterminism in the learning process.

Next we formalise natural trails, where we are not allowed to skip unit propagations.

► **Definition 4** (natural trails). *We call a trail \mathcal{T} natural, if the following holds: For any time (s, t) , $s \in \{0, \dots, r\}$ and $t \in [g_s]$, if $\{D_1, \dots, D_h\}$ are all clauses from the corresponding QCNF that become unit clauses $(\ell_1), \dots, (\ell_h)$ under the assignment $\mathcal{T}[s, t-1]$, then the next propagated literal has to be one of the ℓ_i together with D_i as antecedent clause. If one of the ℓ_i is \perp , then we have to choose this ℓ_i . I.e., conflicts have higher priority.*

The next definition presents the main framework for this paper. Having defined trails in a general sense, we specify how a trail can be generated during a QCDCL run. We introduce the notion of QCDCL-based proofs consisting of three components: the naturally created trails, the clauses learned from each trail, and the proof of each learned clause.

► **Definition 5** (QCDCL proof systems). *Let $\Phi = \mathcal{Q} \cdot \phi$ be a QCNF in n variables. We call a triple of sequences*

$$\iota = \left(\underbrace{(\mathcal{T}_1, \dots, \mathcal{T}_m)}_{=: \theta(\iota)}, \underbrace{(C_1, \dots, C_m)}_{=: \lambda(\iota)}, \underbrace{(\pi_1, \dots, \pi_m)}_{=: \rho(\iota)} \right)$$

a QCDCL $_R^P$ proof from Φ of a clause C for $P \in \{\text{LEV-ORD}, \text{ASS-ORD}, \text{ASS-R-ORD}, \text{ANY-ORD}\}$ and $R \in \{\text{RED}, \text{NO-RED}\}$, if for all $i \in [m]$ the trail \mathcal{T}_i follows the policies P and R and uses the QCNF $\mathcal{Q} \cdot (\phi \cup \{C_1, \dots, C_{i-1}\})$, where $C_j \in \mathcal{L}_{\mathcal{T}_j}$ is a clause learnable from \mathcal{T}_j and $C_m = C$. Each π_i is the derivation of the clause C_i from $\mathcal{Q} \cdot (\phi \cup \{C_1, \dots, C_{i-1}\})$ as defined recursively in Definition 3. We will denote $(\mathcal{T}_1, \dots, \mathcal{T}_m)$ by $\theta(\iota)$, (C_1, \dots, C_m) by $\lambda(\iota)$ and (π_1, \dots, π_m) by $\rho(\iota)$. Note that all these trails need to run into a conflict in order to start clause learning. If $C = (\perp)$ we call ι a refutation.

We also require that \mathcal{T}_1 is natural and for each $i \in \{2, \dots, m\}$ there exist indices (s, t) such that the following holds:

- $\mathcal{T}_i[s, t] = \mathcal{T}_{i-1}[s, t]$.
- For each subtrail $\mathcal{T}_i[a, b]$ with $\mathcal{T}_i[s, t] \subseteq \mathcal{T}_i[a, b]$ and $\perp \notin \mathcal{T}_i[a, b]$ let D_1, \dots, D_h be all the clauses in $\phi \cup \{C_1, \dots, C_{i-1}\}$ such that under the assignment $\mathcal{T}_i[a, b]$ these clauses get unit (under the policy R) with corresponding literals ℓ_1, \dots, ℓ_h . Then we have to propagate one of these literals next, i.e., $\ell_j \in \mathcal{T}_i[a, b+1]$ for some $j \in [h]$, and take the corresponding clause D_j as antecedent.
- In the situation above, if $\perp \in \{\ell_1, \dots, \ell_h\}$, then $\perp \in \mathcal{T}_i[a, b+1]$. I.e., we have to run into a conflict as soon as we find one.

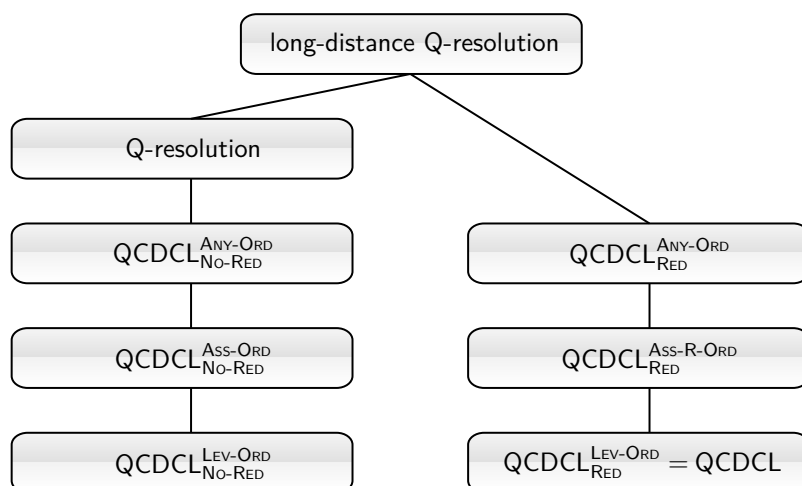
We call this process backtracking to $\mathcal{T}_i[s, t]$. Backtracking to $\mathcal{T}_i[0, 0]$ is called restarting. The size of a proof ι is measured by $|\iota| := \sum_{j=1}^m |\mathcal{T}_j| \in \mathcal{O}(mn)$.

The corresponding (refutational) proof system for false QCNFs is denoted QCDCL $_R^P$. We will refer to these systems as QCDCL proof systems. A trail \mathcal{T} that follows the policies P and R is a QCDCL $_R^P$ trail.

Note that the first trail \mathcal{T}_1 of each proof ι is always natural.

Combining the two policies RED and NO-RED for unit propagation and the four policies ANY-ORD, LEV-ORD, ASS-ORD, and ASS-R-ORD, we obtain six QCDCL systems. These are depicted in Figure 1 (we are not interested in the systems QCDCL $_{\text{RED}}^{\text{ASS-ORD}}$ and QCDCL $_{\text{NO-RED}}^{\text{ASS-R-ORD}}$ since ASS-ORD and ASS-R-ORD would not be beneficial in these combinations). As mentioned, combining LEV-ORD with RED yields the standard QCDCL system, and we will also write QCDCL for QCDCL $_{\text{RED}}^{\text{LEV-ORD}}$. The other five variants are introduced here for the first time.

The decision policies ASS-ORD and ASS-R-ORD might seem slightly unintuitive at first sight. We show that these policies guarantee the learning of so-called asserting clauses (Definition 6) in association with NO-RED resp. RED.



■ **Figure 1** Overview of the defined QCDCL proof systems. Lines denote p-simulations and follow by definition and Theorem 8.

A proof system P p-simulates a proof system S if each S proof can be efficiently transformed into a P proof of the same formula [24]. If the systems p-simulate each other, they are p-equivalent.

It will turn out that π_1, \dots, π_m in Definition 5 are in fact valid long-distance Q-resolution proofs. To prove this, we will argue that in proof systems with NO-RED we cannot derive any tautologies, while with RED we can at most derive universal tautologies.

Next we introduce *asserting learning schemes*. These are commonly used in practice since they guarantee a kind of progression in a run. These learning schemes are important to prevent a trail from backtracking too often.

► **Definition 6** (asserting clauses and asserting learning schemes). *Let $\Phi := \mathcal{Q} \cdot \phi$ be a QCNF in any of the defined QCDCL systems. Let*

$$\mathcal{T} = (p_{(0,1)}, \dots, p_{(0,g_0)}; \mathbf{d}_1, p_{(1,1)}, \dots, p_{(1,g_1)}; \dots; \mathbf{d}_r, p_{(r,1)}, \dots, p_{(r,g_r)} = \perp)$$

be a trail which follows the corresponding policies and $\mathcal{L}_{\mathcal{T}}$ the sequence of learnable clauses. A nonempty clause $C \in \mathcal{L}_{\mathcal{T}}$ is called an asserting clause, if it becomes unit after backtracking, i.e., there exists a time (s, t) with $s \in \{0, \dots, r-1\}$ and $t \in [g_s]$ such that $C|_{\mathcal{T}[s,t]}$ is a unit clause under the corresponding system.

Let \mathbb{T} be the set of trails \mathcal{T} for Φ such that $\perp \in \mathcal{T}$. A learning scheme ξ is a map with domain \mathbb{T} , which maps each \mathcal{T} to a clause $\xi(\mathcal{T}) \in \mathcal{L}_{\mathcal{T}}$.

A learning scheme ξ is called asserting if it maps to asserting clauses or (\perp) as long as $\mathcal{L}_{\mathcal{T}}$ contains such.

It is not guaranteed that we will always find asserting clauses for trails. For example consider the false QCNF $\forall u \exists x \cdot (u \vee x) \wedge (u \vee \bar{x}) \wedge (\bar{u} \vee x) \wedge (\bar{u} \vee \bar{x})$ and the trail $\mathcal{T} = (\mathbf{x}; \mathbf{u}, \perp)$ under the system $\text{QCDCL}_{\text{NO-RED}}^{\text{ANY-ORD}}$. We can only learn the clause $(\bar{u} \vee \bar{x})$, which is non-unit under $\mathcal{T}[0,0] = \emptyset$.

However, we can always learn asserting clauses when using one of the policies ASS-ORD or ASS-R-ORD, which is the reason why we introduced these policies.

► **Lemma 7.**

- Let \mathcal{T} be a trail under the policies ASS-ORD and NO-RED. If $(\perp) \notin \mathcal{L}_{\mathcal{T}}$, then there exists an asserting clause $D \in \mathcal{L}_{\mathcal{T}}$.
- Let \mathcal{T} be a trail under the policies ASS-R-ORD and RED. If $(\perp) \notin \mathcal{L}_{\mathcal{T}}$, then there exists an asserting clause $D \in \mathcal{L}_{\mathcal{T}}$.

We establish that all systems depicted in Figure 1 are sound and complete.

► **Theorem 8.** *All defined QCDCL proof systems are sound and complete QBF proof systems. In particular, all QCDCL calculi are p -simulated by long-distance Q-resolution and the proof systems with NO-RED are even p -simulated by Q-resolution.*

Soundness is shown via efficiently constructing long-distance Q-resolution proofs from QCDCL proofs. Crucially, when using the unit-propagation policy NO-RED, then no long-distance steps are actually needed and we just construct Q-resolution proofs. The resulting simulations are depicted in Figure 1. Simulations between the QCDCL calculi follow by definition. We remark already here that this simulation order simplifies further due to our results in the following sections (cf. Figure 2).

Proving that QCDCL decisions do not necessarily need to follow the order of quantification (as is done in practical QCDCL with policy LEV-ORD), might be a somewhat surprising discovery. It seems to us that inside the QBF community there is the wide-spread belief that following the quantification order in decisions is needed for soundness (cf. e.g. [30, 41, 56]).³ While this is true for QDPLL [22, 30],⁴ it is actually not needed in QCDCL: the quantification order is immaterial for the decisions as long as the quantification order is correctly taken into account when deriving learned clauses (Theorem 8).⁵ Hence our theoretical work also opens the door towards *new solving approaches in practice* (cf. the discussion in Section 8).

From a theoretical point of view, formalising the QCDCL ingredients into proof systems enables a precise proof-theoretic analysis of the QCDCL systems and their comparison to Q-resolution. This will be the underlying feature of our results in the following two sections, showing the incomparability of Q-resolution and QCDCL (Section 4) and the lower bounds for QCDCL (Section 5). We will use it further to obtain a version of QCDCL that is even p -equivalent to Q-resolution (Section 6).

4 QCDCL and Q-resolution are incomparable

This section establishes that QCDCL and Q-resolution are incomparable by exponential separations, i.e., there exist QBFs that are easy for QCDCL, but require exponential-size Q-resolution refutations, and vice versa. As explained above, this is in stark contrast to the propositional setting, where CDCL and resolution are equivalent.

► **Theorem 9.** *The systems Q-resolution and QCDCL are incomparable.*

Proving Theorem 9 requires two families of QBFs. For the first we take the parity formulas.

► **Definition 10** ([12]). *The QCNF QParity_n consists of the prefix $\exists x_1 \dots x_n \forall z \exists t_2 \dots t_n$ and the matrix*

$$\begin{aligned} & x_1 \vee x_2 \vee \bar{t}_2, \quad x_1 \vee \bar{x}_2 \vee t_2, \quad \bar{x}_1 \vee x_2 \vee t_2, \quad \bar{x}_1 \vee \bar{x}_2 \vee \bar{t}_2, \\ & x_i \vee t_{i-1} \vee \bar{t}_i, \quad x_i \vee \bar{t}_{i-1} \vee t_i, \quad \bar{x}_i \vee t_{i-1} \vee t_i, \quad \bar{x}_i \vee \bar{t}_{i-1} \vee \bar{t}_i, \\ & t_n \vee z, \quad \bar{t}_n \vee \bar{z} \end{aligned}$$

for $i \in \{2, \dots, n\}$.

³ In fact we thought so too, prior to this paper.

⁴ The fact that the earlier QDPLL algorithm [22] needs to obey the quantifier order might have been the reason why this policy was adopted in QCDCL as well [56].

⁵ We note, however, that the approach of *dependency learning* [47] starts with an empty set of dependency conditions (cf. [7, 52] for background on dependencies) and incrementally learns new dependencies. As decisions only need to respect the learned dependencies, they can initially be made out of order [47].

The formulas assert that there is an input x_1, \dots, x_n such that the parity $\bigoplus_{i \in [n]} x_i$ is not equal to z . Since z is universally quantified, this means that $\bigoplus_{i \in [n]} x_i$ should be neither 0 nor 1, an obvious contradiction. The parity computation is encoded by using variables t_i for the prefix sums $\bigoplus_{j \in [i]} x_j$. Using strategy extraction for Q-resolution [2, 12] and the result that the parity function is hard for bounded-depth circuits [29, 32], one can show that the QParity_n formulas require exponential-size Q-resolution refutations [12].

Here we show that QParity_n is easy for QCDCL.

► **Proposition 11.** *QParity_n has polynomial-size proofs in QCDCL.*

This requires to construct specific trails and clauses learned from these trails that together comprise a short QCDCL proof of the formulas.

For the opposite separation we consider the following QBFs:

► **Definition 12.** *Let PHP_n^{n+1} be the set of clauses for the pigeonhole principle with n holes and $n + 1$ pigeons using variables x_1, \dots, x_{s_n} . Let Trapdoor_n be the QCNF with the prefix $\exists y_1, \dots, y_{s_n} \forall w \exists t, x_1, \dots, x_{s_n} \forall u$ and the matrix*

$$\text{PHP}_n^{n+1}(x_1, \dots, x_{s_n}) \tag{3}$$

$$\bar{y}_i \vee x_i \vee u, y_i \vee \bar{x}_i \vee u \tag{4}$$

$$y_i \vee w \vee t, y_i \vee w \vee \bar{t}, \bar{y}_i \vee w \vee t, \bar{y}_i \vee w \vee \bar{t} \tag{5}$$

for $i = 1, \dots, s_n$.

We show that these formulas Trapdoor_n require exponential-size QCDCL refutations. In QCDCL, variables have to be decided in order of the quantifier prefix, hence each QCDCL trail for Trapdoor_n has to start with the y variables, which by unit propagation (used together with universal reduction) propagates $x_i = y_i$ for $i \in [s_n]$ by clauses (4). Therefore the trail runs into a conflict on the PHP clauses (3). This happens repeatedly, forcing QCDCL to produce a resolution refutation of the clauses (3), which by the propositional resolution lower bound by Haken [31] has to be of exponential size.

► **Proposition 13.** *The QCNFs Trapdoor_n require exponential-size QCDCL_{RED}^{LEV-ORD} refutations.*

On the other hand, it is easy to obtain short Q-resolution refutations of Trapdoor_n by just using the clauses (5).

► **Proposition 14.** *The QCNFs Trapdoor_n have constant-size Q-resolution refutations.*

This establishes the separation of QCDCL and Q-resolution. We remark that in earlier work, Janota [35] showed that QCDCL with a specific asserting learning scheme requires large running time on some class of QBFs, whereas the same formulas are easy for Q-resolution. Of course, this raises the question whether another learning scheme might produce short QCDCL runs. In contrast, our Theorem 9 rules out any simulation of Q-resolution by QCDCL (or vice versa), regardless of the learning scheme used.

5 Lower bounds for QCDCL

The incomparability of Q-resolution and QCDCL raises the immediate question of what formulas are hard for QCDCL. Previous research has largely concentrated on showing lower bounds for Q-resolution (e.g. [8, 12, 37]). However, by our results from the last section, these lower bounds do not necessarily apply to QCDCL, and prior to this paper no dedicated lower bounds for QCDCL (with arbitrary learning schemes) were known.

12:12 Understanding the Relative Strength of QBF CDCL Solvers and QBF Resolution

Here we show that several formulas from the QBF literature, including the equality formulas and a large class of random QBFs [8] are indeed hard for QCDCL.

We start by defining a proof system in which we can analyse hardness in classical QCDCL.

► **Definition 15.** *We call a long-distance Q-resolution proof π of a clause C from a QCNF Φ a long-distance QCDCL resolution proof of C from Φ , if there exists a $\text{QCDCL}_{\text{RED}}^{\text{LEV-ORD}}$ proof ι of C from Φ such that the long-distance Q-resolution proof π is obtained by pasting together the sub-proofs (π_1, \dots, π_m) from ι (cf. Definition 5).*

The system long-distance QCDCL resolution identifies a fragment of long-distance Q-resolution, which collects all long-distance Q-resolution proofs that appear in $\text{QCDCL}_{\text{RED}}^{\text{LEV-ORD}}$ derivations. By definition therefore, long-distance QCDCL resolution and $\text{QCDCL}_{\text{RED}}^{\text{LEV-ORD}}$ are p-equivalent proof systems.

Our next goal is to identify a whole class of QCNFs that witness the hardness of QCDCL.

The equality formulas from [8] are arguably one of the simplest families of QBFs that are interesting from a proof complexity perspective. The formula Equality_n is defined as the QCNF

$$\exists x_1 \dots x_n \forall u_1 \dots u_n \exists t_1 \dots t_n \cdot (\bar{t}_1 \vee \dots \vee \bar{t}_n) \wedge \bigwedge_{i=1}^n ((\bar{x}_i \vee \bar{u}_i \vee t_i) \wedge (x_i \vee u_i \vee t_i)).$$

These formulas are of the type Σ_3^b , i.e., they have two quantifier alternations starting with \exists .

Inspired by this construction, [8] considered a class of randomly generated QCNFs, again of type Σ_3^b .

► **Definition 16** ([8]). *For each $1 \leq i \leq n$ let $C_i^{(1)}, \dots, C_i^{(cn)}$ be clauses picked uniformly at random from the set of clauses containing 1 literal from the set $U_i = \{u_i^{(1)}, \dots, u_i^{(m)}\}$ and 2 literals from $X_i = \{x_i^{(1)}, \dots, x_i^{(n)}\}$. Define the randomly generated QCNF $Q(n, m, c)$ as:*

$$Q(n, m, c) := \exists X_1, \dots, X_n \forall U_1, \dots, U_n \exists t_1, \dots, t_n \cdot \bigwedge_{i=1}^n \bigwedge_{j=1}^{cn} (\bar{t}_i \vee C_i^{(j)}) \wedge (t_1 \vee \dots \vee t_n).$$

Suitably choosing the parameters c and m , we obtain false QBFs with high probability.

Both the equality and the random formulas require exponential-size proofs in Q-resolution (the random formulas whp) [8]. This is shown in [8] via the size-cost-capacity technique, a semantically grounded QBF lower-bound technique that infers Q-resolution hardness for formulas Φ_n (and in fact hardness for even stronger systems) from lower bounds for the size of countermodels for Φ_n .

It is not clear how to directly apply this technique to QCDCL. Instead, we identify a property, which we term the *XT-property*, that we can use to lift hardness from Q-resolution to QCDCL.

► **Definition 17.** *Let Φ be a QCNF of the form $\exists X \forall U \exists T \cdot \phi$ with sets of variables $X = \{x_1, \dots, x_a\}$, $U = \{u_1, \dots, u_b\}$ and $T = \{t_1, \dots, t_c\}$.*

We call a clause C in the variables of Φ

- *T-clause, if $\text{var}(C) \cap X = \emptyset$, $\text{var}(C) \cap U = \emptyset$ and $\text{var}(C) \cap T \neq \emptyset$,*
- *XT-clause, if $\text{var}(C) \cap X \neq \emptyset$, $\text{var}(C) \cap U = \emptyset$ and $\text{var}(C) \cap T \neq \emptyset$,*
- *XUT-clause, if $\text{var}(C) \cap X \neq \emptyset$, $\text{var}(C) \cap U \neq \emptyset$ and $\text{var}(C) \cap T \neq \emptyset$.*

We say that Φ fulfils the XT-property if ϕ contains no XT-clauses as well as no unit T-clauses and there do not exist two T-clauses $C_1, C_2 \in \phi$ that are resolvable.

Intuitively, this says that in a Σ_3^b formula Φ with quantifier prefix of the form $\exists X \forall U \exists T$ with blocks of variables X, U, T , there is no direct connection between the X and T variables, i.e., Φ does not contain clauses with X and T variables, but no U variables.

We can then prove that QCDCL runs on formulas with this XT -property can be efficiently transformed into Q-resolution refutations, not only into long-distance Q-resolution refutations.

We first show that under the XT -property we cannot derive any XT -clauses.

► **Lemma 18.** *It is not possible to derive XT -clauses by long-distance Q-resolution from a QCNF Φ that fulfils the XT -property.*

Proof. Assume that we can derive an XT -clause C by a long-distance Q-resolution proof π from Φ . Let D be the first XT -clause in π (D might be equal to C). Since Φ contains no XT -clauses as axioms, the last step before D has to be a resolution or reduction. A reduction is not possible since the reduced universal literal would have been blocked by a T -literal in D .

Therefore D is the resolvent of two preceding clauses D_1 and D_2 . If we resolve over an X -literal, then one of these clauses has to be an XT -clause. The same is true for a resolution over a T -literal. However, this contradicts the fact that D was the first XT -clause in π . ◀

The next lemma shows that under the XT -property it is also not possible to derive any non-axiomatic T -clauses.

► **Lemma 19.** *Let Φ be a QCNF with the XT -property and let C be a T -clause derived by long-distance Q-resolution from Φ . Then C is an axiom from Φ .*

We will show later that we need to resolve two XUT -clauses over an X -literal in order to introduce tautologies. Now we prove that this is not possible in long-distance QCDCL resolution under the XT -property.

► **Lemma 20.** *It is not possible to resolve two XUT -clauses over an X -literal in a long-distance QCDCL resolution proof of a QCNF Φ that fulfils the XT -property.*

Proof. Assume there is a long-distance QCDCL resolution proof π that contains such a resolution step over an X -literal x . Let C_1 and C_2 be the corresponding XUT -clauses. One of these clauses, say C_1 , had to be an antecedent clause in a $\text{QCDCL}_{\text{RED}}^{\text{LEV-ORD}}$ trail \mathcal{T} that implied x . Since our decisions in the trail are level-ordered and we did not skip any decisions, either x was propagated at decision level 0, or at a decision level in which we decided another X -literal.

Because C_1 is an XUT -clause, we can find a T -literal $t \in C_1$. The literal \bar{t} must have been propagated before we implied x (\bar{t} could not have been decided because the decisions are level-ordered). That means that for the same trail we can find $E := \text{ante}_{\mathcal{T}}(\bar{t})$. Now, E cannot be a unit T -clause by the XT -property and Lemma 19. Hence E must contain further X -, U -, or T -literals. If E contains a U -literal, then we would have had to decide this U -literal before we use E as an antecedent clause, contradicting the level-order of our decisions. Also, this U -literal cannot be reduced since we want to imply a T -literal with the help of E . Therefore we conclude that E contains an X -literal or a T -literal. If E contains an X -literal, then E is an XT -clause, which is not possible by Lemma 18.

Therefore E contains at least another T -literal $\ell \in E$. As before, the literal $\bar{\ell}$ was propagated before we implied \bar{t} and x . We set $E' := \text{ante}_{\mathcal{T}}(\bar{\ell})$ and argue in the same way as with E . This process would repeat endlessly, which is a contradiction since we only have finitely many T -variables. ◀

12:14 Understanding the Relative Strength of QBF CDCL Solvers and QBF Resolution

Thus for formulas with the XT -property we can lift Q-resolution lower bounds to QCDCL, yielding the next theorem.

► **Theorem 21.** *If Φ fulfils the XT -property and requires Q-resolution refutations of size s , then each QCDCL refutation of Φ has size at least s as well.*

Proof. Let π be a long-distance QCDCL resolution refutation of Φ . We show that π does not contain any tautological clause C and hence π is in fact a Q-resolution proof.

Assume that π contains some tautological clause C . W.l.o.g. let C be the first tautological clause in π . Clearly, C has to be derived by a resolution step over an X -literal. Let C_1 and C_2 be the parent clauses of C . Both of them contain some X -literals and some U -literals. They also have to contain T -literals, otherwise we would reduce all U -literals (in the learning process we reduce as soon as possible). Therefore C_1 and C_2 are both XUT -clauses that are resolved over an X -literal, which is not possible by Lemma 20.

Therefore such a clause C cannot exist. Hence each long-distance QCDCL resolution refutation of Φ is even a Q-resolution refutation and the result follows. ◀

It is quite easy to check that both the equality formulas as well as the random formulas above have the XT -property. Thus we obtain:

► **Corollary 22.**

- Equality $_n$ requires QCDCL refutations of size 2^n .
- Let $1 < c < 2$ be a constant and $m \leq (1 - \epsilon) \log_2 n$ for some constant $\epsilon > 0$. With probability $1 - o(1)$ the random QCNF $Q(n, m, c)$ is false and requires QCDCL refutations of size $2^{\Omega(n^\epsilon)}$.

Our findings so far reveal an interesting picture on QCDCL hardness. Firstly, Proposition 11 and Corollary 22 imply that *not all Q-resolution hardness results lift to QCDCL*: the lower bounds for equality and random formulas shown via size-cost-capacity [8] *do*, but the lower bounds for parity shown via circuit complexity [12] *do not*.

Secondly, it is worth to compare the QCDCL hardness results for **Trapdoor** from the previous section to the QCDCL hardness results shown here for equality and random formulas. The hardness of **Trapdoor** lifts from propositional hardness for PHP, while the hardness of equality and random formulas lifts from Q-resolution hardness. In fact, this can be made formal by using a model of QBF proof systems with access to an NP oracle [17], which allows to collapse propositional subderivations of arbitrary size into just one oracle inference step. Hardness under the NP-oracle version of Q-resolution guarantees that the hardness is “genuine” to QBF and not lifted from propositional resolution. We show here that this notion of “genuine” QBF hardness, tailored towards QCDCL, also holds for the QCDCL lower bounds for equality and the random QBFs.

► **Proposition 23.** *The number of reduction steps in each long-distance QCDCL resolution refutation (and also each QCDCL $_{RED}^{LEV-ORD}$ refutation) of Equality $_n$ is at least 2^n . The analogous result holds for the false formulas $Q(n, m, c)$ with $2^{\Omega(n^\epsilon)}$ reduction steps.*

On the other hand, the parity formulas also exhibit “genuine” QBF hardness, as they are hard in the NP-oracle version of Q-resolution [10]. Since they are easy for QCDCL (Proposition 11), this means that not all genuine Q-resolution lower bounds lift to QCDCL.

Thirdly, hardness for QCDCL can of course also stem from hardness for long-distance Q-resolution, since the latter system p-simulates the former. However, there are only very few hardness results for long-distance Q-resolution known in the literature [3, 12, 13], hence

our hardness results shown here should be also valuable for practitioners, in particular the hardness results for the large class of random QCNFs. It is also worth noting that the equality formulas are easy for long-distance Q-resolution [9], hence our results imply an exponential separation between QCDCL and long-distance Q-resolution.

► **Proposition 24.** Long-distance Q-resolution is exponentially stronger than QCDCL, i.e., long-distance Q-resolution p -simulates QCDCL and there are QCNFs that require exponential-size proofs in QCDCL, but admit polynomial-size proofs in long-distance Q-resolution.

6 A QCDCL system that characterises Q-resolution

In one of our main results we obtain a QCDCL characterisation of Q-resolution. Of course, given that Q-resolution and QCDCL are incomparable (Section 4), we cannot hope to achieve such a characterisation by simply strengthening some of the QCDCL policies.⁶ As explained in the previous section, traditional QCDCL is using the decision policy LEV-ORD and the unit-propagation policy RED. To obtain a QCDCL system equivalent to Q-resolution, we will have to change both policies. We will *strengthen* the decision policy and replace LEV-ORD by ANY-ORD (we could also replace it with the intermediate version ASS-ORD). In addition, we will somewhat *weaken* the unit propagation policy from RED to NO-RED.⁷

This leads to the following characterisation of Q-resolution.

► **Theorem 25.** Q-resolution, $\text{QCDCL}_{\text{NO-RED}}^{\text{ANY-ORD}}$, and $\text{QCDCL}_{\text{NO-RED}}^{\text{ASS-ORD}}$ are p -equivalent proof systems.

In particular, each Q-resolution refutation π of a QCNF in n variables can be transformed into a $\text{QCDCL}_{\text{NO-RED}}^{\text{ASS-ORD}}$ -refutation of size $\mathcal{O}(n^3 \cdot |\pi|)$ that uses an arbitrary asserting learning scheme.

One part of the simulation above was already shown in Theorem 8, where we proved that all QCDCL systems with NO-RED are p -simulated by Q-resolution. The technically most challenging part is the reverse simulation where we need to construct $\text{QCDCL}_{\text{NO-RED}}^{\text{ASS-ORD}}$ trails from Q-resolution proofs. The main conceptual notion we use is that of *reliable* clauses.

► **Definition 26.** Let $\Phi = \mathcal{Q} \cdot \phi$ be a QCNF and C be a non-tautological clause. If there is a $\text{QCDCL}_{\text{NO-RED}}^{\text{ASS-ORD}}$ trail \mathcal{T} , an existential literal $\ell \in C$ and a set of literals $\alpha \subseteq \bar{C} \setminus \{\bar{\ell}\}$ such that α is the set of decision literals in \mathcal{T} and $\ell \in \mathcal{T}$, then C is called *unreliable* with respect to Φ . Alternatively, we say that the decisions \bar{C} are blocking each other.

If C is not unreliable, we call C *reliable*.

Intuitively, a reliable clause C can be used to form a $\text{QCDCL}_{\text{NO-RED}}^{\text{ASS-ORD}}$ trail by using all negated literals from C as decisions. This way we progress through the Q-resolution proof, successively learning clauses and making all clauses C in the Q-resolution proof unreliable until we obtain the empty clause.

This construction bears some similarities to the simulation of Q-resolution by CDCL [48], but poses further technical challenges due to quantification and the additional rules of Q-resolution. In the inductive argument for Theorem 25 we therefore need to distinguish three cases on whether C is an axiom or derived by resolution or reduction, each requiring its own lemma (Lemmas 27, 28, and 29). For the following lemmas, let ξ be an arbitrary, but fixed asserting learning scheme.

⁶ Such hope might not have seemed totally implausible prior to this paper, e.g. [35] states that ‘CDCL QBF solving appears to be quite weak compared to general Q-resolution.’

⁷ While intuitively NO-RED might indeed appear weaker than RED (it produces fewer unit propagations), we show in the next section that they are in fact incomparable, cf. Figure 2.

► **Lemma 27.** *Let $\Phi := \mathcal{Q} \cdot \phi$ be a QCNF in n variables and $C \in \phi$. If C is reliable with respect to Φ , there exists a $\text{QCDCL}_{\text{NO-RED}}^{\text{ASS-ORD}}$ -proof ι with trails $\mathcal{T}_1, \dots, \mathcal{T}_{f_n}$ from Φ of some clause E that uses the learning scheme ξ such that $|\iota| \in \mathcal{O}(n^3)$. If $E \neq (\perp)$, then C is unreliable with respect to $\mathcal{Q} \cdot (\phi \cup \{\xi(\mathcal{T}_1), \dots, \xi(\mathcal{T}_{f_n})\})$.*

► **Lemma 28.** *Let $\Phi := \mathcal{Q} \cdot \phi$ be a QCNF in n variables. Also let $C_1 \vee x$ be a clause that is unreliable with respect to $\Psi := \mathcal{Q} \cdot \psi$ with $\psi \subseteq \phi$ and $C_2 \vee \bar{x}$ unreliable with respect to $\Upsilon := \mathcal{Q} \cdot \tau$ with $\tau \subseteq \phi$, such that $C_1 \vee C_2$ is non-tautological. Let ξ be an asserting learning scheme. If $C_1 \vee C_2$ is reliable with respect to Φ , there exists a $\text{QCDCL}_{\text{NO-RED}}^{\text{ASS-ORD}}$ -proof ι with $\theta(\iota) = \mathcal{T}_1, \dots, \mathcal{T}_{f_n}$ from Φ of some clause E that uses the learning scheme ξ such that $|\iota| \in \mathcal{O}(n^3)$. If $E \neq (\perp)$, then $C_1 \vee C_2$ is unreliable with respect to $\mathcal{Q} \cdot (\phi \cup \{\xi(\mathcal{T}_1), \dots, \xi(\mathcal{T}_{f_n})\})$.*

► **Lemma 29.** *Let $\Phi := \mathcal{Q} \cdot \phi$ be a QCNF in n variables, let $D := C \vee u_1 \vee \dots \vee u_m$ be a non-tautological clause with universal literals u_1, \dots, u_m and $\text{red}(D) = C$, such that D is unreliable with respect to a QCNF $\Psi = \mathcal{Q} \cdot \psi$ with $\psi \subseteq \phi$. Let ξ be an asserting learning scheme. If C is reliable with respect to Φ , there exists a $\text{QCDCL}_{\text{NO-RED}}^{\text{ASS-ORD}}$ -proof ι with $\theta(\iota) = \mathcal{T}_1, \dots, \mathcal{T}_{f_n}$ from Φ of some clause E that uses the learning scheme ξ such that $|\iota| \in \mathcal{O}(n^3)$. If $E \neq (\perp)$, then C is unreliable with respect to $\mathcal{Q} \cdot (\phi \cup \{\xi(\mathcal{T}_1), \dots, \xi(\mathcal{T}_{f_n})\})$.*

We also point out that in comparison to the notion of 1-empowering clauses from [48], our argument via reliability yields somewhat better bounds on the simulation, thereby implying a slight quantitative improvement by a factor of n in the simulation in [48]:

► **Theorem 30.** *Let ϕ be a CNF in n variables and let π be a resolution refutation of ϕ . Then ϕ has a CDCL refutation of size $\mathcal{O}(n^3|\pi|)$.*

7 The simulation order of QCDCL proof systems

We can now analyse the simulation order of the defined QCDCL and QBF resolution systems, cf. Figure 2 which almost completely determines the simulations and separations between the systems involved (cf. Section 8 for the open cases).

We highlight the most interesting findings (in addition to the results already described).

Firstly, we show that the unit-propagation policies RED and NO-RED are incomparable when fixing the decision policy LEV-ORD used in practical QCDCL.

► **Theorem 31.** *The systems $\text{QCDCL}_{\text{NO-RED}}^{\text{LEV-ORD}}$ and $\text{QCDCL}_{\text{RED}}^{\text{LEV-ORD}}$ are incomparable.*

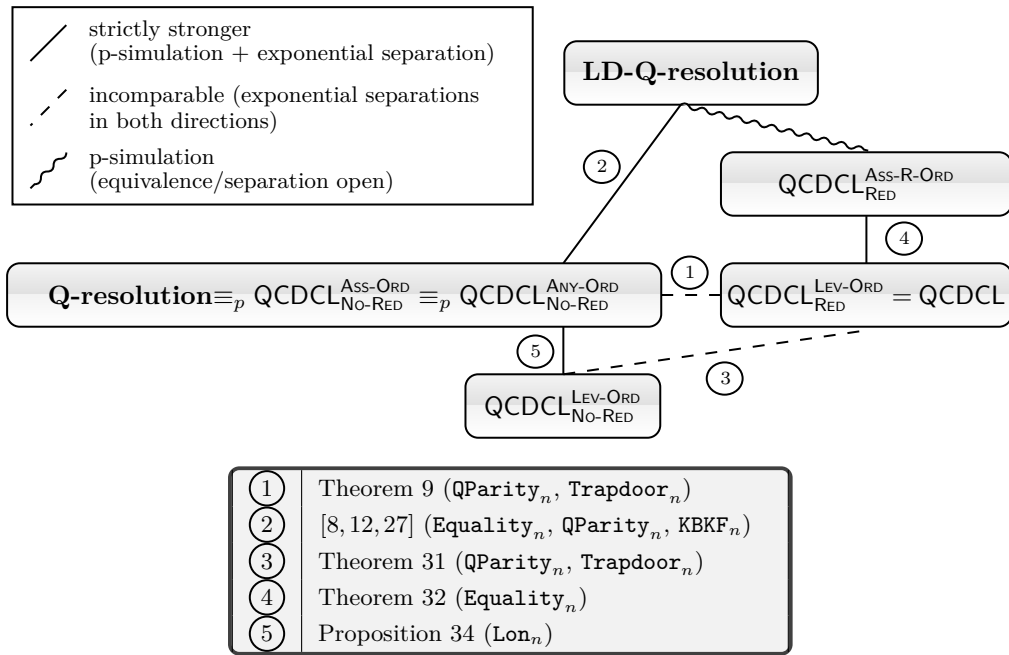
For the separations we use the QBFs QParity_n and Trapdoor_n . For practice, this results means that it is a priori not clear that the unit-propagation policy as used in practical QCDCL is actually preferable to the simpler unit-propagation policy from CDCL (which would work in QCDCL as well).

Secondly, we show that replacing the decision policy LEV-ORD in QCDCL with the more liberal decision policy ASS-R-ORD yields exponentially shorter QCDCL runs, which we demonstrate on the Equality_n formulas.

► **Theorem 32.** *$\text{QCDCL}_{\text{RED}}^{\text{ASS-R-ORD}}$ is exponentially stronger than $\text{QCDCL}_{\text{RED}}^{\text{LEV-ORD}}$.*

Again, this theoretical result identifies potential for improvements in practical solving (cf. also the discussion in the concluding Section 8).

Thirdly, we recall the formulas Lon_n that were introduced by Lonsing in [39]. Originally, these QCNFs were constructed to separate QBF solvers that differ in the implemented dependency schemes (we do not consider these concepts here, though).



■ **Figure 2** The simulation order of QCDCL and QBF resolution systems. The table contains pointers to the separating formulas.

► **Definition 33** (Lonsing [39]). *Let Lon_n be the QCNF*

$$\exists a, b, b_1, \dots, b_{s_n} \forall x, y \exists c, d \cdot (a \vee x \vee c) \wedge (a \vee b \vee b_1 \vee \dots \vee b_{s_n}) \wedge (b \vee y \vee d) \wedge (x \vee c) \wedge (x \vee \bar{c}) \wedge \text{PHP}_n^{n+1}(b_1, \dots, b_{s_n}) .$$

It was shown in [39] that these formulas become easy to refute by choosing the standard dependency scheme. However, Lon_n serve as witnesses for separating our systems as well.

► **Proposition 34.** *The QCNFs Lon_n require exponential-size proofs in the proof systems QCDCL_{RED}^{LEV-ORD} and QCDCL_{NO-RED}^{LEV-ORD}, but have constant-size proofs in QCDCL_{RED}^{ASS-R-ORD} and Q-resolution.*

8 Conclusion

In this paper we performed a formal, proof-theoretic analysis of QCDCL. In particular, we focused on the relation of QCDCL and Q-resolution, showing both the incomparability of practically-used QCDCL to Q-resolution as well as the equivalence of a new QCDCL version to Q-resolution.

In addition to the theoretical contributions of this paper, we believe that our findings will also be interesting for practitioners. Firstly, because we have shown the first rigorous dedicated hardness results for QCDCL, not only in terms of formula families with at most one instance per input size (as is typical in proof complexity), but also in terms of a large family of random QBFs.

Secondly, we believe that it would be interesting to test the potential of our new QCDCL variants for practical solving. Though we have formulated these as proof systems, it should be fairly straightforward to incorporate our new policies into actual QCDCL implementations.

In particular, the insight that decisions do not need to follow the order of quantification in the prefix should be a welcome discovery. Of course, when just using the policy ANY-ORD, it is not clear that asserting clauses can always be learnt. Therefore, we suggest that for practical implementations, the most interesting new systems should be $\text{QCDCL}_{\text{NO-RED}}^{\text{ASS-ORD}}$ and $\text{QCDCL}_{\text{RED}}^{\text{ASS-R-ORD}}$. Both facilitate liberal decision policies, not necessarily following the prefix order, while still allowing to learn asserting clauses. Since both systems are incomparable, it is a priori not clear which one to prefer in practice. However, we would suggest that $\text{QCDCL}_{\text{RED}}^{\text{ASS-R-ORD}}$ should be the more interesting system, since it uses the same unit propagation as QCDCL, but provides an exponential strengthening of QCDCL (as shown in Theorem 32) via the decision policy ASS-R-ORD.

We close with some open questions that are triggered by the results presented here:

- Can we find an alternative formula instead of Trapdoor_n for the separation between Q-resolution and QCDCL (easy for Q-resolution, hard for QCDCL)? I.e., we are primarily interested in formulas whose hardness does not depend on propositional resolution.
- Can we find a separation between $\text{QCDCL}_{\text{RED}}^{\text{ASS-R-ORD}}$ and long-distance Q-resolution?
- Can we even find a separation between $\text{QCDCL}_{\text{RED}}^{\text{ANY-ORD}}$ and long-distance Q-resolution, or are the systems possibly even equivalent?

References

- 1 Albert Atserias, Johannes Klaus Fichte, and Marc Thurley. Clause-learning algorithms with many restarts and bounded-width resolution. *J. Artif. Intell. Res.*, 40:353–373, 2011.
- 2 Valeriy Balabanov and Jie-Hong R. Jiang. Unified QBF certification and its applications. *Form. Methods Syst. Des.*, 41(1):45–65, 2012.
- 3 Valeriy Balabanov, Magdalena Widl, and Jie-Hong R. Jiang. QBF resolution systems and their proof complexities. In *Proc. International Conference on Theory and Applications of Satisfiability Testing (SAT)*, pages 154–169, 2014.
- 4 Paul Beame, Henry A. Kautz, and Ashish Sabharwal. Towards understanding and harnessing the potential of clause learning. *J. Artif. Intell. Res. (JAIR)*, 22:319–351, 2004. doi:10.1613/jair.1410.
- 5 Paul Beame and Toniann Pitassi. Propositional proof complexity: Past, present, and future. In G. Paun, G. Rozenberg, and A. Salomaa, editors, *Current Trends in Theoretical Computer Science: Entering the 21st Century*, pages 42–70. World Scientific Publishing, 2001.
- 6 Olaf Beyersdorff and Joshua Blinkhorn. Lower bound techniques for QBF expansion. *Theory Comput. Syst.*, 64(3):400–421, 2020.
- 7 Olaf Beyersdorff, Joshua Blinkhorn, Leroy Chew, Renate A. Schmidt, and Martin Suda. Reinterpreting dependency schemes: Soundness meets incompleteness in DQBF. *J. Autom. Reason.*, 63(3):597–623, 2019.
- 8 Olaf Beyersdorff, Joshua Blinkhorn, and Luke Hinde. Size, cost, and capacity: A semantic technique for hard random QBFs. *Logical Methods in Computer Science*, 15(1), 2019.
- 9 Olaf Beyersdorff, Joshua Blinkhorn, and Meena Mahajan. Building strategies into QBF proofs. In *STACS, LIPIcs*, pages 14:1–14:18. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2019.
- 10 Olaf Beyersdorff, Joshua Blinkhorn, and Meena Mahajan. Hardness characterisations and size-width lower bounds for QBF resolution. In *Proc. ACM/IEEE Symposium on Logic in Computer Science (LICS)*, pages 209–223. ACM, 2020.
- 11 Olaf Beyersdorff, Ilario Bonacina, Leroy Chew, and Jan Pich. Frege systems for quantified Boolean logic. *J. ACM*, 67(2), 2020.
- 12 Olaf Beyersdorff, Leroy Chew, and Mikolás Janota. New resolution-based QBF calculi and their proof complexity. *ACM Transactions on Computation Theory*, 11(4):26:1–26:42, 2019.

- 13 Olaf Beyersdorff, Leroy Chew, Meena Mahajan, and Anil Shukla. Feasible interpolation for QBF resolution calculi. *Logical Methods in Computer Science*, 13, 2017.
- 14 Olaf Beyersdorff, Leroy Chew, Meena Mahajan, and Anil Shukla. Understanding cutting planes for QBFs. *Inf. Comput.*, 262:141–161, 2018.
- 15 Olaf Beyersdorff, Leroy Chew, and Karteek Sreenivasaiah. A game characterisation of tree-like Q-Resolution size. *J. Comput. Syst. Sci.*, 104:82–101, 2019.
- 16 Olaf Beyersdorff and Luke Hinde. Characterising tree-like Frege proofs for QBF. *Inf. Comput.*, 268, 2019.
- 17 Olaf Beyersdorff, Luke Hinde, and Ján Pich. Reasons for hardness in QBF proof systems. *ACM Transactions on Computation Theory*, 12(2), 2020.
- 18 Olaf Beyersdorff, Mikolás Janota, Florian Lonsing, and Martina Seidl. Quantified Boolean formulas. In Armin Biere, Marijn Heule, Hans van Maaren, and Toby Walsh, editors, *Handbook of Satisfiability, 2nd edition*, Frontiers in Artificial Intelligence and Applications. IOS press, 2021.
- 19 Maria Luisa Bonet, Sam Buss, and Jan Johannsen. Improved separations of regular resolution from clause learning proof systems. *J. Artif. Intell. Res.*, 49:669–703, 2014.
- 20 Samuel R. Buss. Towards NP-P via proof complexity and search. *Ann. Pure Appl. Logic*, 163(7):906–917, 2012.
- 21 Samuel R. Buss, Jan Hoffmann, and Jan Johannsen. Resolution trees with lemmas: Resolution refinements that characterize DLL algorithms with clause learning. *Logical Methods in Computer Science*, 4(4), 2008. doi:10.2168/LMCS-4(4:13)2008.
- 22 Marco Cadoli, Andrea Giovanardi, and Marco Schaerf. An algorithm to evaluate quantified Boolean formulae. In *Proc. AAAI Conference on Artificial Intelligence (AAAI)*, pages 262–267, 1998.
- 23 Stephen A. Cook and Tsuyoshi Morioka. Quantified propositional calculus and a second-order theory for NC¹. *Arch. Math. Log.*, 44(6):711–749, 2005.
- 24 Stephen A. Cook and Robert A. Reckhow. The relative efficiency of propositional proof systems. *The Journal of Symbolic Logic*, 44(1):36–50, 1979.
- 25 Martin Davis, George Logemann, and Donald W. Loveland. A machine program for theorem-proving. *Commun. ACM*, 5(7):394–397, 1962. doi:10.1145/368273.368557.
- 26 Martin Davis and Hilary Putnam. A computing procedure for quantification theory. *Journal of the ACM*, 7:210–215, 1960.
- 27 Uwe Egly, Florian Lonsing, and Magdalena Widl. Long-distance resolution: Proof generation and strategy extraction in search-based QBF solving. In *Proc. Logic for Programming, Artificial Intelligence, and Reasoning (LPAR)*, pages 291–308, 2013.
- 28 Peter Faymonville, Bernd Finkbeiner, Markus N. Rabe, and Leander Tentrup. Encodings of bounded synthesis. In *Proc. Tools and Algorithms for the Construction and Analysis of Systems - 23rd International Conference (TACAS)*, pages 354–370, 2017.
- 29 Merrick L. Furst, James B. Saxe, and Michael Sipser. Parity, circuits, and the polynomial-time hierarchy. *Mathematical Systems Theory*, 17(1):13–27, 1984.
- 30 Enrico Giunchiglia, Paolo Marin, and Massimo Narizzano. Reasoning with quantified Boolean formulas. In Armin Biere, Marijn Heule, Hans van Maaren, and Toby Walsh, editors, *Handbook of Satisfiability*, volume 185 of *Frontiers in Artificial Intelligence and Applications*, pages 761–780. IOS Press, 2009.
- 31 Amin Haken. The intractability of resolution. *Theoretical Computer Science*, 39:297–308, 1985.
- 32 Johan Håstad. *Computational Limitations of Small Depth Circuits*. MIT Press, Cambridge, 1987.
- 33 Philipp Hertel, Fahiem Bacchus, Toniann Pitassi, and Allen Van Gelder. Clause learning can effectively p-simulate general propositional resolution. In *Proc. AAAI Conference on Artificial Intelligence (AAAI)*, pages 283–290, 2008. URL: <http://www.aaai.org/Library/AAAI/2008/aaai08-045.php>.

- 34 Marijn J. H. Heule, Martina Seidl, and Armin Biere. Solution validation and extraction for QBF preprocessing. *J. Autom. Reason.*, 58(1):97–125, 2017.
- 35 Mikolás Janota. On Q-resolution and CDCL QBF solving. In *Proc. International Conference on Theory and Applications of Satisfiability Testing (SAT'16)*, pages 402–418, 2016.
- 36 Mikolás Janota and Joao Marques-Silva. Expansion-based QBF solving versus Q-resolution. *Theor. Comput. Sci.*, 577:25–42, 2015.
- 37 Hans Kleine Büning, Marek Karpinski, and Andreas Flögel. Resolution for quantified Boolean formulas. *Inf. Comput.*, 117(1):12–18, 1995.
- 38 Jan Krajčiček. *Proof complexity*, volume 170 of *Encyclopedia of Mathematics and Its Applications*. Cambridge University Press, 2019.
- 39 Florian Lonsing. *Dependency Schemes and Search-Based QBF Solving: Theory and Practice*. PhD thesis, Johannes Kepler University Linz, 2012.
- 40 Florian Lonsing and Uwe Egly. DepQBF 6.0: A search-based QBF solver beyond traditional QCDCL. In *Proc. International Conference on Automated Deduction (CADE)*, pages 371–384, 2017.
- 41 Florian Lonsing, Uwe Egly, and Allen Van Gelder. Efficient clause learning for quantified Boolean formulas via QBF pseudo unit propagation. In *Proc. International Conference on Theory and Applications of Satisfiability Testing (SAT)*, pages 100–115, 2013.
- 42 João P. Marques Silva, Inês Lynce, and Sharad Malik. Conflict-driven clause learning SAT solvers. In *Handbook of Satisfiability*. IOS Press, 2009.
- 43 João P. Marques Silva and Kareem A. Sakallah. GRASP - a new search algorithm for satisfiability. In *ICCAD*, pages 220–227, 1996. doi:10.1145/244522.244560.
- 44 Nathan Mull, Shuo Pang, and Alexander A. Razborov. On CDCL-based proof systems with the ordered decision strategy. In *Proc. International Conference on Theory and Applications of Satisfiability Testing (SAT'20)*, pages 149–165. Springer, 2020.
- 45 Jakob Nordström. *Short Proofs May Be Spacious : Understanding Space in Resolution*. PhD thesis, Royal Institute of Technology, Stockholm, Sweden, 2008.
- 46 Jakob Nordström. On the interplay between proof complexity and SAT solving. *SIGLOG News*, 2(3):19–44, 2015.
- 47 Tomás Peitl, Friedrich Slivovsky, and Stefan Szeider. Dependency learning for QBF. *J. Artif. Intell. Res.*, 65:180–208, 2019.
- 48 Knot Pipatsrisawat and Adnan Darwiche. On the power of clause-learning SAT solvers as resolution engines. *Artif. Intell.*, 175(2):512–525, 2011. doi:10.1016/j.artint.2010.10.002.
- 49 Luca Pulina and Martina Seidl. The 2016 and 2017 QBF solvers evaluations (QBFVAL'16 and QBFVAL'17). *Artif. Intell.*, 274:224–248, 2019.
- 50 Nathan Segerlind. The complexity of propositional proofs. *Bulletin of Symbolic Logic*, 13(4):417–481, 2007.
- 51 Ankit Shukla, Armin Biere, Luca Pulina, and Martina Seidl. A survey on applications of quantified Boolean formulas. In *Proc. IEEE International Conference on Tools with Artificial Intelligence (ICTAI)*, pages 78–84, 2019.
- 52 Friedrich Slivovsky and Stefan Szeider. Soundness of Q-resolution with dependency schemes. *Theoretical Computer Science*, 612:83–101, 2016.
- 53 Moshe Y. Vardi. Boolean satisfiability: theory and engineering. *Commun. ACM*, 57(3):5, 2014.
- 54 Marc Vinyals. Hard examples for common variable decision heuristics. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, 2020.
- 55 Lintao Zhang, Conor F. Madigan, Matthew W. Moskewicz, and Sharad Malik. Efficient conflict driven learning in Boolean satisfiability solver. In *Proc. IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*, pages 279–285, 2001.
- 56 Lintao Zhang and Sharad Malik. Conflict driven learning in a quantified Boolean satisfiability solver. In *Proc. IEEE/ACM International Conference on Computer-aided Design (ICCAD)*, pages 442–449, 2002.