Electronic Colloquium on Computational Complexity, Report No. 53 (2020)

Understanding the Relative Strength of QBF CDCL Solvers and QBF Resolution

Olaf Beyersdorff   Benjamin Böhm

Institute of Computer Science, Friedrich Schiller University Jena, Germany

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.

ISSN 1433-8092

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 [47]. Starting with the classic DPLL algorithm from the 1960s [19,20], 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) [38, 49], the predominant technique 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. [37] 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. [45] for applications).

One of the main techniques in QBF solving is the propositional CDCL technique, lifted to QBF in the form of QCDCL [50]. 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 [30] and dependency schemes in dependency-aware solving [35, 46]. Compared to SAT solving, QBF solving is still at an earlier stage. However, QBF solving has seen huge improvements during the past 15 years [43], and there are problems of practical relevance where QBF solvers outperform SAT solvers [22].

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 [15,41]. 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 [41], which is arguably the most studied proof system in proof complexity.

Propositional resolution operates on clauses and uses the resolution rule

$$\frac{C \vee x \qquad D \vee \bar{x}}{C \vee D} \tag{1.1}$$

as its only inference rule to derive a new clause $C \vee D$ from the two parent clauses $C \vee x$ and $D \vee \bar{x}$. There is a host of lower bounds and lower bound techniques available for propositional resolution (cf. [5, 32, 44] for surveys).

While it is relatively easy to see that the classic DPLL branching algorithm [19, 20] 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, 38]. The opposite simulation is considerably more difficult, with a series of works [1, 4, 28, 42] culminating in the

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

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 [40].

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 [4,42] use arbitrary decision heuristics and perform excessive restarts, both of which diverge from practical CDCL policies. Indeed, in very recent work [48] it was shown that CDCL with practical decision heuristics such as VSIDS [49] is exponentially weaker than resolution, and similar results have been obtained for further decision heuristics [39]. Regarding restarts there is intense research aiming to determine the power of CDCL without restarts from a proof complexity perspective (cf. [14,16]).

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 [31] 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 (1.1) 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*

$$\frac{C \vee u}{C} \quad , \tag{1.2}$$

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 [3,6,10] as well as lower bound techniques, some of them lifted from propositional proof complexity [11,12], but more interestingly some of them genuine to the QBF domain [6,8] that unveil deep connections between proof size and circuit complexity [9], unparalleled in the propositional domain.

Unlike in the relation between SAT and CDCL, it is 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 [2,50]. 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 [50]. Long-distance Q-resolution implements a more liberal use of the resolution rule (1.1), which allows to derive certain tautologies (cf. Section 2.3 for details). In general, allowing to derive tautologies with (1.1) is unsound, an example is given in Section 2.3. 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 [2,50]. However, it is known that long-distance Q-resolution allows exponentially shorter proofs than Q-resolution for some QBFs [6,7,21].

We also remark that there are further QBF resolution systems (cf. [10] for an overview), some of them corresponding to other solving approaches in QBF, such as the system ∀Exp+Res that captures expansion QBF solving [30].

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 [1,4,42] is currently absent in the QBF domain. This brings us to the topic of this paper.

1.1 Our contributions

We state and explain our main contributions and provide pointers to where these are proven in the main part.

1.1.1 QCDCL and Q-resolution are incomparable

Our first contribution establishes that QCDCL and $\mathsf{Q\text{-}resolution}$ are incomparable by exponential separations, i.e., there exist QBFs that are easy for QCDCL, but require exponential-size $\mathsf{Q\text{-}resolution}$ refutations, and vice versa. As explained above, this is in stark contrast to the propositional setting, where CDCL and resolution are equivalent.

Theorem 1.1 (Thm 4.8). *The systems* $\mathsf{Q\text{-}resolution}$ *and* QCDCL *are incomparable.*

Proving Theorem 1.1 requires two families of QBFs. For the first we take the parity formulas.

Definition 1.2 ([10]). *The QCNF* $\mathtt{QParity}_n$ *consists of the prefix* $\exists x_1 \dots x_n \forall z \exists t_2 \dots t_n$ *and the matrix*

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

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

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 $\mathsf{Q\text{-}resolution}$ [2,10] and the result that the parity function is hard for bounded-depth circuits [23,27], one can show that the $\mathtt{QParity}_n$ formulas require exponential-size $\mathsf{Q\text{-}resolution}$ refutations [10].

Here we show that $\mathtt{QParity}_n$ is easy for QCDCL.

Proposition 1.3 (Prop. 4.2). $\mathtt{QParity}_n$ *has polynomial-size proofs in* QCDCL.

This requires to formally state QCDCL in terms of a proof system (we will denote this by QCDCL and explain it in Sections 1.1.3 and 3) and 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 1.4 (Def. 4.5). *Let* $\mathtt{PHP}_n^{n+1}$ *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* $\mathtt{Trapdoor}_n$ *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*

$$\mathtt{PHP}_n^{n+1}(x_1, \dots, x_{s_n}) \tag{1.3}$$

$$\bar{y}_i \vee x_i \vee u,\ y_i \vee \bar{x}_i \vee u \tag{1.4}$$

$$y_i \vee w \vee t,\ y_i \vee w \vee \bar{t},\ \bar{y}_i \vee w \vee t,\ \bar{y}_i \vee w \vee \bar{t} \tag{1.5}$$

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

We show that these formulas $\mathtt{Trapdoor}_n$ require exponential-size QCDCL refutations (Proposition 4.6). In QCDCL, variables have to be decided in order of the quantifier prefix, hence each QCDCL trail for $\mathtt{Trapdoor}_n$ 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 (1.4). Therefore the trail runs into a conflict on the PHP clauses (1.3). This happens repeatedly, forcing QCDCL

to produce a resolution refutation of the clauses (1.3), which by the propositional resolution lower bound by Haken [26] has to be of exponential size. On the other hand, it is easy to obtain short Q-resolution refutations of $\texttt{Trapdoor}_n$ by just using the clauses (1.5) (Proposition 4.7).

This establishes the separation of QCDCL and Q-resolution. We remark that in earlier work, Janota [29] 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 1.1 rules out any simulation of Q-resolution by QCDCL (or vice versa), regardless of the learning scheme used.

1.1.2 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. [6, 10, 31]). However, by our results from the last subsection, 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.

Here we show that several formulas from the QBF literature, including the equality formulas and a large class of random QBFs [6] are indeed hard for QCDCL. The equality formulas from [6] are arguably one of the simplest families of QBFs that are interesting from a proof complexity perspective. The formula $\texttt{Equality}_n$ 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, [6] considered a class of randomly generated QCNFs, again of type Σ_3^b.

Definition 1.5 ([6])**.** *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) [6]. This is shown in [6] 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* (Definition 5.2), that we can use to lift hardness from Q-resolution to QCDCL. Intuitively, it 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 (there are some further condition on clauses containing only T variables (Definition 5.2)).

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

Theorem 1.6 (Thm 5.6)**.** *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.*

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 1.7 (Cor. 5.9 & Cor. 5.13)**.**

- $\texttt{Equality}_n$ *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 1.3 and Corollary 1.7 imply that *not all* Q-resolution *hardness results lift to* QCDCL: the lower bounds for equality and random formulas shown via size-cost-capacity [6] *do*, but the lower bounds for parity shown via circuit complexity [10] *do not*.

Secondly, it is worth to compare the QCDCL hardness results for `Trapdoor` from the previous subsection 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 [13], 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 5.16).

On the other hand, the parity formulas also exhibit 'genuine' QBF hardness, as they are hard in the NP-oracle version of Q-resolution [8]. Since they are easy for QCDCL (Proposition 1.3), 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 simulates the former.[2] However, there are only very few hardness results for long-distance Q-resolution known in the literature [3,10], 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 [7], hence our results imply an exponential separation between QCDCL and long-distance Q-resolution (Corollary 5.10).

1.1.3 Our framework: QCDCL as formal proof systems

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. This involves formalising a number of QCDCL ingredients (cf. Section 2.4 for an informal account on how QCDCL works). We will just sketch this here, the full formal details are given in Section 3.

We start with the notion of a *QCDCL trail $\mathcal{T}$* for a QCNF Φ. The trail $\mathcal{T}$ is a sequence of literals, which we typically denote as

$$\mathcal{T} = (p_{(0,1)}, \ldots, p_{(0,g_0)}; \mathbf{d_1}, p_{(1,1)}, \ldots, p_{(1,g_1)}; \ldots; \mathbf{d_r}, p_{(r,1)}, \ldots, p_{(r,g_r)}).$$

Here $d_1, \ldots, d_r$ are the decision variables and the p variables are propagated by unit propagation. While decisions can be either existential or universal, propagated variables are always existential. In classical QCDCL, the following *decision policy* is adopted:

[2] 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 [18]. If the systems p-simulate each other, they are p-equivalent.

- Lev-Ord - For each $d_i \in \mathcal{T}$, all variables from quantifier blocks left of d_i in the prefix of Φ appear left of d_i in $\mathcal{T}$, i.e., all variables on which d_i depends have been decided or propagated before d_i is decided.

This decision policy therefore follows the order of quantification in the prefix, for which reason we call it level ordered (Lev-Ord).

In addition to Lev-Ord, we consider three more decision policies. The first one stems from propositional CDCL where the order of decisions is completely arbitrary:

- Any-Ord - Given a trail $\mathcal{T}$, we can choose any remaining literal as the next decision.

Before defining the remaining two decision policies, we explain our policies for unit propagation. The first comes again just from propositional CDCL:

- No-Red - For each propagated literal $p_{(i,j)} \in \mathcal{T}$ there has to be a clause C in Φ such that C becomes a single-literal clause under the sub-trail $\mathcal{T}[i, j-1]$ of $\mathcal{T}$ that contains all decisions and propagations in $\mathcal{T}$ before $p_{(i,j)}$.

To illustrate this with a small example, assume that Φ contains a clause $C = x \vee \bar{y} \vee z$ and $\mathcal{T}$ contains the decisions $\bar{x}, y$. Then C is simplified to the single literal z under the assignment $\mathcal{T}$, and hence z is propagated and included as the next variable in $\mathcal{T}$.

This is just CDCL propagation. It is, however, not what is done in QCDCL. Assume again we have a clause $C = x \vee \bar{y} \vee z \vee u$ in Φ and $\exists x, y, z \forall u$ appears in the prefix of Φ. If the trail contains $\bar{x}, y$, we cannot propagate z with the policy No-Red. However, we can use universal reduction on u as in rule (1.2) of Q-resolution to reduce the clause $z \vee u$ (the clause C under the assignment corresponding to $\mathcal{T}$) to the single-literal clause z. Hence we can immediately propagate z with the following unit propagation policy:

- Red - For each propagated literal $p_{(i,j)} \in \mathcal{T}$ there is a clause C in Φ such that C becomes a single-literal clause under the trail $\mathcal{T}[i, j-1]$ using universal reduction.

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).

We prove that this learning process works independently from our policies, e.g., even when Any-Ord or No-Red is used, we can correctly perform long-distance Q-resolution for clause learning as in QCDCL (Proposition 3.7). For practical (Q)CDCL, it is important that we do not just learn any clause, but an *asserting clause* D, which means that D becomes unit after backtracking.

We notice that though the policy Any-Ord is sound, it does not always allow to learn asserting clauses (Remark 3.10). Therefore, we introduce further policies, which are intermediate between Lev-Ord and Any-Ord and still guarantee that asserting clauses can be learned (Lemmas 3.13 and 3.14). We define two policies Ass-Ord and Ass-R-Ord, to be used with the unit propagation policies No-Red and Red, respectively.

- Ass-Ord - Let $(d_1, \dots, d_r)$ be the decision literals of $\mathcal{T}$ and d_k be the rightmost universal literal in $\mathcal{T}$. Then we have $\mathrm{lv}(d_1) \leq \ldots \leq \mathrm{lv}(d_k)$.[3]

- Ass-R-Ord - We can only decide an existential variable x next, if and only if we already decided all universal variables u with $\mathrm{lv}(u) < \mathrm{lv}(x)$ before in $\mathcal{T}$.

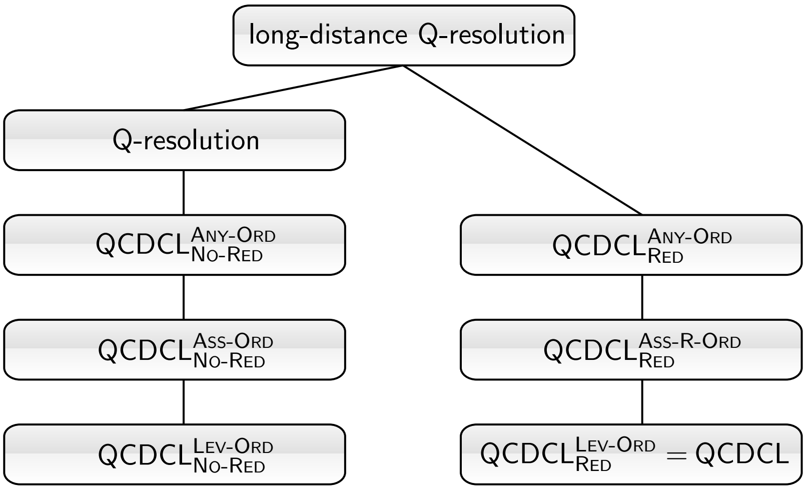

long-distance Q-resolution
Q-resolution
$\mathsf{QCDCL}^{\text{Any-Ord}}_{\text{No-Red}}$
$\mathsf{QCDCL}^{\text{Any-Ord}}_{\text{Red}}$
$\mathsf{QCDCL}^{\text{Ass-Ord}}_{\text{No-Red}}$
$\mathsf{QCDCL}^{\text{Ass-R-Ord}}_{\text{Red}}$
$\mathsf{QCDCL}^{\text{Lev-Ord}}_{\text{No-Red}}$
$\mathsf{QCDCL}^{\text{Lev-Ord}}_{\text{Red}} = \mathsf{QCDCL}$

Figure 1: Overview of the defined QCDCL proof systems. Lines denote p-simulations and follow by definition and Theorem 1.8.

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 $\mathsf{QCDCL}^{\text{Ass-Ord}}_{\text{Red}}$ and $\mathsf{QCDCL}^{\text{Ass-R-Ord}}_{\text{No-Red}}$). As mentioned, combining LEV-ORD with RED yields the standard QCDCL system. The other five variants are introduced here for the first time.

To actually show that these systems are sound and to proof-theoretically analyse their strength, we turn these six systems into formal refutational proof systems for QBF (Definition 3.5). A *proof of a clause C_m from a QCNF $\Phi = \mathcal{Q} \cdot \phi$ in the calculus* QCDCL^P_R is of the form

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

where

- $\mathcal{T}_i$ are trails under the decision policy P and unit-propagation policy R,
- C_i is a clause learnable from $\mathcal{T}_i$, and
- π_i is a long-distance Q-resolution derivation of the clause C_i from $\mathcal{Q} \cdot (\phi \cup \{C_1, \dots, C_{i-1}\})$.

The proofs π_i are not arbitrary, but are defined recursively by resolving along the trails $\mathcal{T}_i$ (Definition 3.3). If C_m is the empty clause, then we speak of a *refutation* of Φ.

We establish that all these systems are sound and complete QBF proof systems.

Theorem 1.8 (Thm 3.8 & Thm 3.15). *All defined QCDCL proof systems are sound and complete. 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 next two subsections (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.

[3] Under a prefix $Q_1X_1Q_2X_2\dots Q_sX_s$ with disjoint blocks of variables X_i and alternating blocks of quantifiers $Q_i \in \{\exists, \forall\}$, the *quantifier level* of a variable x is $\text{lv}(x) = i$, if $x \in X_i$.

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. [24,36,50]).[4] While this is true for QDPLL [17,24],[5] we show here that this 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 1.8). Hence our theoretical work also opens the door towards *new solving approaches in practice* (cf. the discussion in the concluding Section 9).

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 already was the underlying feature of our results in the previous two subsections, showing the incomparability of Q-resolution and QCDCL (Section 1.1.1) and the lower bounds for QCDCL (Section 1.1.2). We will now use it further to obtain a version of QCDCL that is even p-equivalent to Q-resolution.

1.1.4 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 1.1.1), we cannot hope to achieve such a characterisation by simply strengthening some of the QCDCL policies.[6] As explained in the previous subsection, 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 1.9 (Thm 6.9). Q-resolution*,* $\mathsf{QCDCL}^{\mathit{ANY\text{-}ORD}}_{\mathit{NO\text{-}RED}}$*, and* $\mathsf{QCDCL}^{\mathit{ASS\text{-}ORD}}_{\mathit{NO\text{-}RED}}$ *are p-equivalent proof systems.*

In particular, each Q-resolution *refutation π of a QCNF in n variables can be transformed into a* $\mathsf{QCDCL}^{\mathit{ASS\text{-}ORD}}_{\mathit{NO\text{-}RED}}$*-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 1.8, 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 $\mathsf{QCDCL}^{\mathsf{ASS\text{-}ORD}}_{\mathsf{NO\text{-}RED}}$ trails from Q-resolution proofs. The main conceptual notion we use is that of *reliable* clauses (Definition 6.1). Intuitively, a reliable clause C can be used to form a $\mathsf{QCDCL}^{\mathsf{ASS\text{-}ORD}}_{\mathsf{NO\text{-}RED}}$ 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 [42], but poses further technical challenges due to quantification and the additional rules of Q-resolution. In the inductive argument we therefore need to distinguish three cases on whether C is an axiom or derived by resolution or reduction, each requiring its own lemma (6.6, 6.7, and 6.8).

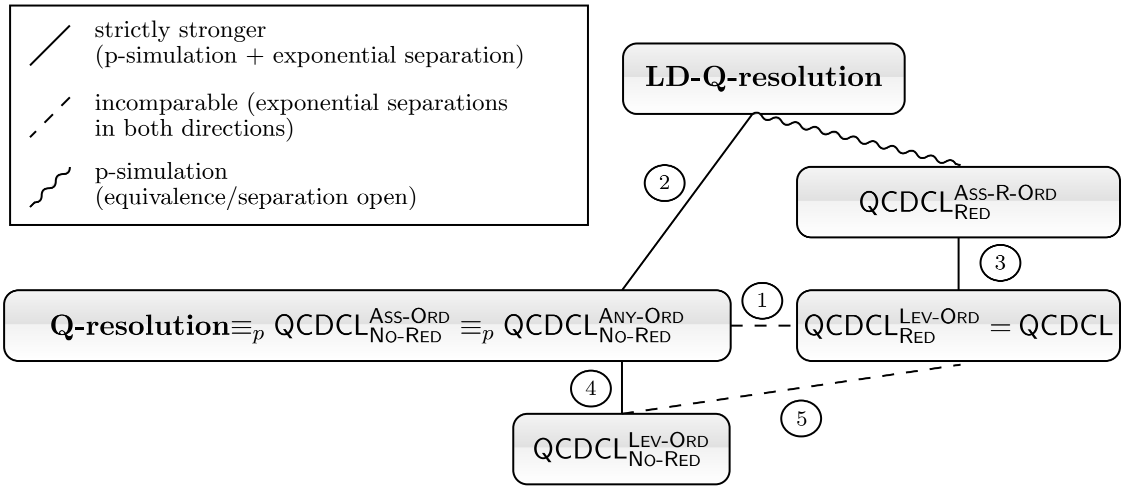
We also point out that in comparison to the notion of 1-empowering clauses from [42], our argument via reliability yields somewhat better bounds on the simulation, thereby implying a slight quantitative improvement by a factor of n in the simulation in [42] (cf. the more elaborate discussion in Section 7):

[4] In fact we thought so too, prior to this paper.

[5] The fact that the earlier QDPLL algorithm [17] needs to obey the quantifier order might have been the reason why this policy was adopted in QCDCL as well [50].

[6] Such hope might not have seemed totally implausible prior to this paper, e.g. [29] states that 'CDCL QBF solving appears to be quite weak compared to general Q-resolution.'

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


strictly stronger
(p-simulation + exponential separation)
incomparable (exponential separations in both directions)
p-simulation
(equivalence/separation open)
LD-Q-resolution
$\mathsf{QCDCL}^{\textsc{Ass-R-Ord}}_{\textsc{Red}}$
Q-resolution$\equiv_p$ $\mathsf{QCDCL}^{\textsc{Ass-Ord}}_{\textsc{No-Red}}$ $\equiv_p$ $\mathsf{QCDCL}^{\textsc{Any-Ord}}_{\textsc{No-Red}}$
$\mathsf{QCDCL}^{\textsc{Lev-Ord}}_{\textsc{Red}} = \mathsf{QCDCL}$
$\mathsf{QCDCL}^{\textsc{Lev-Ord}}_{\textsc{No-Red}}$
1
2
3
4
5

①	Theorem 4.8 ($\texttt{QParity}_n$, $\texttt{Trapdoor}_n$)
②	[6, 10, 21] ($\texttt{Equality}_n$, $\texttt{QParity}_n$, $\texttt{KBKF}_n$)
③	Theorem 8.6 ($\texttt{Equality}_n$)
④	Proposition 8.2 ($\texttt{Lon}_n$)
⑤	Theorem 8.4 ($\texttt{QParity}_n$, $\texttt{Trapdoor}_n$)

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

Theorem 1.10 (Thm 7.2)**.** *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 $\mathcal{O}(n^3|\pi|)$.*

1.1.5 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 9 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 $\textsc{Red}$ and $\textsc{No-Red}$ are incomparable when fixing the decision policy $\textsc{Lev-Ord}$ used in practical QCDCL.

Theorem 1.11 (Thm 8.4)**.** *The systems $\mathsf{QCDCL}^{\textsc{Lev-Ord}}_{\textsc{No-Red}}$ and $\mathsf{QCDCL}^{\textsc{Lev-Ord}}_{\textsc{Red}}$ are incomparable.*

For the separations we use the QBFs $\texttt{QParity}_n$ and $\texttt{Trapdoor}_n$. 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 $\textsc{Lev-Ord}$ in QCDCL with the more liberal decision policy $\textsc{Ass-R-Ord}$ yields exponentially shorter QCDCL runs, which we demonstrate on the $\texttt{Equality}_n$ formulas.

Theorem 1.12 (Thm 8.6)**.** *$\mathsf{QCDCL}^{\textsc{Ass-R-Ord}}_{\textsc{Red}}$ is exponentially stronger than $\mathsf{QCDCL}^{\textsc{Lev-Ord}}_{\textsc{Red}}$.*

Again, this theoretical result identifies potential for improvements in practical solving (cf. also the discussion in the concluding Section 9).

1.2 Organisation

The main part of the article is organised slightly differently from the order of results as outlined above. We start in Section 2 with reviewing relevant notions concerning quantified Boolean logic, QBF proof systems, and QCDCL.

Section 3 formalises QCDCL with a number of different policies for variable decision and unit propagation as proof systems. This constitutes the formal framework for the rest of the paper. The proof systems are shown to be sound and complete.

Section 4 shows the incomparability of QCDCL and $\mathsf{Q\text{-}resolution}$ by exponential separations. This is followed in Section 5 by further hardness results for QCDCL.

In Section 6 we obtain the characterisation of $\mathsf{Q\text{-}resolution}$ in terms of the new QCDCL proof system $\mathsf{QCDCL}^{\textsc{Any-Ord}}_{\textsc{No-Red}}$. We discuss implications of this result to the equivalence of propositional resolution and CDCL in Section 7.

Section 8 reveals the full picture of the simulation order of the defined QCDCL proof systems. We conclude in Section 9 with some open questions and a discussion of the potential impact of our results for practice.

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 $\ell_1 \vee \ldots \vee \ell_m$ of some literals $\ell_1, \ldots, \ell_m$. 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 $(\square)$. For reasons of consistency it is helpful to define an *empty literal*, denoted by $\square$ in our case. As a consequence, we have $\square \in (\square)$, although we define the empty clause as a clause with zero literals.

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

Let $C = \ell_1 \vee \ldots \vee \ell_m$. We define $\text{var}(C) := \{\text{var}(\ell_1), \ldots, \text{var}(\ell_m)\}$. For a CNF $\phi = C_1 \wedge \ldots \wedge C_n$ 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

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

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 $\langle X \rangle$. Assignments can also operate on CNFs in the natural sense. A CNF ϕ *entails* another CNF ψ if each assignment that satisfies ϕ also satisfies ψ (denoted by $\phi \vDash \psi$).

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 $Q_1 x_1 Q_2 x_2 \ldots Q_k x_k$ consists of variables $x_1, \ldots, x_k$

and quantifiers $Q_1, \ldots, Q_k \in \{\exists, \forall\}$. We obtain an equivalent formula if we unite adjacent quantifiers of the same type. Therefore we can always assume that our prefix is in the form of

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

with nonempty sets of variables $X_1, \ldots, X_s$ and quantifiers $Q'_1, \ldots, Q'_s \in \{\exists, \forall\}$ 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 $\operatorname{lv}(x) = \operatorname{lv}_\Phi(x) = i$, 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 $\operatorname{var}_\exists(\Phi)$, and the set of universal variables by $\operatorname{var}_\forall(\Phi)$.

A QBF whose matrix is a CNF is called a *QCNF*. We require that all clauses from a matrix of a QCNF are non-tautological, otherwise we would just delete these clauses. This requirement is crucial for the correctness of the derivation rules we define later for our 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 Proof systems

A *proof system* for a language $\mathcal{L}$ is a polynomial-time computable surjective function $f : \{0,1\}^* \to \mathcal{L}$ [18]. A *proof* for $\phi \in \mathcal{L}$ is some $\pi \in \{0,1\}^*$ such that $f(\pi) = \phi$. In our case, the language $\mathcal{L}$ will be mostly UNSAT (unsatisfiable formulas) or FQBF (false QBFs). For unsatisfiable or false formulas we often call the system *refutational*.

To show that such a polynomial-time function f is actually a proof system for $\mathcal{L}$, we have to verify two properties:

- *soundness*: $f(\{0,1\}^*) \subseteq \mathcal{L}$.
- *completeness*: $f(\{0,1\}^*) \supseteq \mathcal{L}$.

A proof system f for a language $\mathcal{L}$ is *simulated* by another proof system g for $\mathcal{L}$, if there exists a function $h : \{0,1\}^* \to \{0,1\}^*$ such that $g \circ h = f$ (denoted $f \leq g$) [33]. If h is polynomial-time computable, then we say that g *p-simulates* f (denoted $f \leq_p g$) [18]. If two systems p-simulate each other, they are *p-equivalent* (denoted $f \equiv_p g$).

2.3 Q-resolution and long-distance Q-resolution

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

$$\operatorname{res}(C_1 \vee \ell, C_2 \vee \bar{\ell}, \ell) := C_1 \vee C_2.$$

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

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

Then we can perform a reduction step and obtain

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

For a CNF $\phi = \{C_1, \dots, C_k\}$ we define

$$\mathrm{red}(\phi) := \{\mathrm{red}(C_1), \dots, \mathrm{red}(C_k)\}.$$

Q-resolution [31] 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:* $C_i = \mathrm{res}(C_j, C_k, x)$ for some $j, k < i$ and $x \in \mathrm{var}_\exists(\Phi)$, and C_i is non-tautological;
- *Reduction:* $C_i = \mathrm{red}(C_j)$ 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 $(\Box)$.

For the simulation of the original version of QCDCL, the proof system long-distance Q-resolution was introduced in [2, 50]. This extension of Q-resolution allows to derive universal tautologies under specific 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):* $C_i = \mathrm{res}(C_j, C_k, x)$ for some $j, k < i$ and $x \in \mathrm{var}_\exists(\Phi)$. The resolvent C_i is allowed to contain a tautology $u \vee \bar{u}$ if u is a universal variable. If $u \in \mathrm{var}(C_j) \cap \mathrm{var}(C_k)$, then we additionally require $\mathrm{lv}(u) > \mathrm{lv}(x)$.

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 $\mathrm{red}(\mathrm{res}(u \vee \bar{x}, \bar{u} \vee x, x)) = (\Box)$ is unsound since we resolved over an existential literal x with $\mathrm{lv}_\Psi(x) > \mathrm{lv}_\Psi(u)$ while generating $u \vee \bar{u}$.

2.4 QCDCL

Quantified conflict-driven clause learning (QCDCL) is the quantified version of the well-known CDCL algorithm (see [37,49] for further details on CDCL, and [25,34,50] for QCDCL). Let $\Phi = \mathcal{Q} \cdot \phi$ be a false QCNF. Roughly speaking, QCDCL consists of two processes: The *propagation process* and the *learning process.*

In the *propagation process* we generate assignments to the end that we obtain a conflict. We start with clauses from ϕ that force us to assign literals such that we do not falsify these clauses (subsequently called unit clauses). The underlying idea of this process is *unit propagation.* One can think of a clause $x_1 \vee \ldots \vee x_n$ as an implication $(\bar{x}_1 \wedge \ldots \wedge \bar{x}_{n-1}) \rightarrow x_n$. That is, if we already assigned the literals $\bar{x}_1, \dots, \bar{x}_{n-1}$, then we are forced to assign x_n in order to verify this clause. If x_n was universal, this would already be a conflict since this clause must be true for both assignments of x_n in order to not get falsified. In general, we also have to insert reduction steps into this process. Hence we are interested in clauses that become unit after reduction. For example, the clause $(\bar{x}_1 \wedge \ldots \wedge \bar{x}_{n-1}) \rightarrow (x_n \vee u)$ for an existential literal x_n and a universal literal u with $\mathrm{lv}(x_n) < \mathrm{lv}(u)$ can also be used as an implication of x_n for unit propagation.

Of course we could also assign $\bar{x}_n$, immediately leading to a conflict by falsifying the clause $x_1 \vee \ldots \vee x_n$, but this conflict would have been solely caused by this clause and would not give us any new information for the learning process. Our goal is to prolong a conflict as long as possible in the hope of learning something helpful from it. However, it is not guaranteed that we can even perform any unit propagations by just starting with the formula.

Therefore we will make *decisions*, i.e., we assign literals without any solid reason. With the aid of these decisions (one can also think of assumptions) we can provoke further unit propagations. Since decision making is one of the non-deterministic components of the algorithm, we will try to keep its influence as low as possible. In detail, this means we will only make decisions if there are no more unit propagations available. In the classical QCDCL these decisions have to follow a level-order. This means we always have to decide the next available variable with the lowest quantifier level. However, we will later show that this condition is not necessary for soundness or completeness. There are even QCNFs whose hardness are based simply on this level-order, so leaving out this limitation would actually strengthen the algorithm. In our model these two policies will be denoted by LEV-ORD and ANY-ORD.

After we obtained a conflict, we can start the *clause learning process*. Here the underlying idea is to use Q-resolution resp. long-distance Q-resolution. We start with the clause that caused our conflict and resolve it with clauses that implied previous literals in the assignment in the reversed propagation order. At the end we get a (hopefully) new clause such that each assignment that falsifies this clause also leads to a conflict. In addition we get a long-distance Q-resolution-derivation of this learned clause from Φ. We will add the learned clause to ϕ, backtrack to a state before we assigned all literals of this clause and start with the propagation process again. The algorithm ends as soon as we learn the empty clause ($\Box$) and therefore obtain a refutation of Φ.

In our work we want to formalize this algorithm as a proof system and slightly modify the system such that this new proof system becomes equivalent to Q-resolution. To ensure we actually construct a Q-resolution proof out of the learning process, we have to prevent the introduction of universal tautologies. As will become clear later, the reasons for these tautologies are reductions in the propagation process. This is the only way where both literals of universal variables can be introduced. This motivates disallowing these reductions in the propagation process. Later in the definitions we will denote these policies by RED and NO-RED.

Usually QCDCL has to handle both refutations of false formulas as well as proving the validity of true formulas. For this purpose one would need to implement the so called *cube learning* (or *term learning*) for fulfilling assignments. But since we are only interested in the refutation of formulas (otherwise we could not compare this system to Q-resolution), we will omit this aspect of QCDCL.

3 Our framework: versions of QCDCL as proof systems

In this section we define formal proof systems that capture QCDCL solving. For this we need to formally define central ingredients of QCDCL solving, including trails, decision policies, unit propagation, and clause learning. 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 3.1 (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 $\Box$) of variables from Φ with some 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 $\Box$. 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 in the i^{th} decision level,* *determined by the corresponding decision d_i. The decision level 0 is the only level where we do not have a decision literal. Similarly to 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 the time *(s,t) is the trail consisting of all literals from the leftmost literal in $\mathcal{T}$ up to (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 defined as the empty trail.*

Now, we need 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 $var(d_i) \neq var(d_k)$ for each $i \neq k \in \{0, \dots, r\}$. If $\Box \in \mathcal{T}$, then this must be the last (rightmost) literal in $\mathcal{T}$. In this case we will 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 $x \in var(\phi) \backslash var(\mathcal{T}[i-1, g_{i-1}])$. That means we have to decide the variables along the quantification order.*
- ASS-ORD - *Let $(d_1, \dots, d_r)$ be the decision literals of $\mathcal{T}$ and d_k be the rightmost universal literal in this order. Then we have $lv(d_1) \leq \dots \leq lv(d_k)$.*
- 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 $red(C|_{\mathcal{T}[i,j-1]}) = (p_{(i,j)})$.*
- 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 policies are called antecedent clauses*, which will be denoted by $ante_{\mathcal{T}}(p_{(i,j)}) := C$. There could be more than one such suitable clause, in this case we will just choose one of them arbitrarily. These antecedent clauses clearly depend on the unit propagation policy we use.*

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

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

Definition 3.2 (unit clauses)**.** *Let C be a clause. In the policy RED, we call C a* unit clause *if $red(C) = (x)$ for an existential literal x or $x = \Box$.*

Otherwise, for NO-RED, we call C a unit clause *if $C = (x)$ for an existential literal x or $x = \Box$.*

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

Next we will formalise the process of clause learning from trails that run into a conflict. The idea is to resolve all antecedent clauses, starting from the end of the trail, until we stop at some point. We will always resolve over the corresponding propagated literal and skip literals not used for the implication of the conflict.

Definition 3.3 (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 $p_{(r,g_r)} = \Box$ that follows policies $P \in \{$*LEV-ORD*, *ASS-ORD*, *ANY-ORD*$\}$ and $R \in \{$*RED*, *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 $C_{(r,g_r)} := \mathit{red}(\mathit{ante}(p_{(r,g_r)}))$,

$$C_{(i,j)} := \begin{cases} \mathit{red}[\mathit{res}(C_{(i,j+1)}, \mathit{red}(\mathit{ante}(p_{(i,j)})), p_{(i,j)})] & \text{if } \bar{p}_{(i,j)} \in C_{(i,j+1)}, \\ C_{(i,j+1)} & \text{otherwise} \end{cases}$$

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

$$C_{(i,g_i)} := \begin{cases} \mathit{red}[\mathit{res}(C_{(i+1,1)}, \mathit{red}(\mathit{ante}(p_{(i,g_i)})), p_{(i,g_i)})] & \text{if } \bar{p}_{(i,g_i)} \in C_{(i+1,1)}, \\ C_{(i+1,1)} & \text{otherwise} \end{cases}$$

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.

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

Definition 3.4 (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 under the assignment $\mathcal{T}[s, t-1]$ with literals $\ell_1, \dots, \ell_h$, 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 $\Box$, then we have to choose this ℓ_i. I.e., conflicts have higher priority.*

The next definition presents the main framework for the whole paper. After having defined trails in a general sense, we want to specify the way a trail can be generated during a QCDCL run. We will give the notion of QCDCL-based proofs consisting of three components: the naturally created trails, the clauses we learned from each trail, and the proof of each learned clause.

Definition 3.5 (QCDCL proof systems). *Let $\Phi = \mathcal{Q} \cdot \phi$ be a QCNF in n variables. We call a triple of sequences*

$$\iota = ((\mathcal{T}_1, \dots, \mathcal{T}_m), (C_1, \dots, C_m), (\pi_1, \dots, \pi_m))$$

a QCDCL_R^P proof *from Φ of a clause C for $P \in \{$LEV-ORD, ASS-ORD, ASS-R-ORD, ANY-ORD$\}$ and $R \in \{$RED, 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 3.3. Note that all these trails need to run into a conflict in order to start clause learning. If $C = (\Box)$ 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 $\Box \notin \mathcal{T}_i[a,b]$ 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 $\ell_1, \dots, \ell_h$. 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 $\Box \in \{\ell_1, \dots, \ell_h\}$, then $\Box \in \mathcal{T}_i[a, b+1]$. I.e., we have to run into a conflict as soon as we find one.*

We call that backtracking *to $\mathcal{T}_i[s,t]$. Backtracking to $\mathcal{T}_i[0,0]$ is called* restarting.

The size of such a proof ι is measured by $|\iota| := \sum_{j=1}^{m} |T_j| \in \mathcal{O}(mn)$.

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 sometimes also called a QCDCL_R^P* trail.

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

In combination with RED, the policy $\mathsf{LEV\text{-}ORD}$ represents the original QCDCL algorithm without further modifications. The order policies $\mathsf{ASS\text{-}ORD}$ and $\mathsf{ASS\text{-}R\text{-}ORD}$ might seem slightly unintuitive at first sight. We will show later that these policies guarantee the learning of so-called asserting clauses (which will be defined in Definition 3.9) in association with $\mathsf{NO\text{-}RED}$ resp. RED. Since $\mathsf{ASS\text{-}ORD}$ (resp. $\mathsf{ASS\text{-}R\text{-}ORD}$) will only unfold its impact in combination with $\mathsf{NO\text{-}RED}$ (resp. RED), we will not consider the systems $\mathsf{QCDCL}_{\mathsf{RED}}^{\mathsf{ASS\text{-}ORD}}$ or $\mathsf{QCDCL}_{\mathsf{NO\text{-}RED}}^{\mathsf{ASS\text{-}R\text{-}ORD}}$.

We will show later that $\pi_1, \dots, \pi_m$ in Definition 3.5 are valid long-distance Q-resolution proofs. In order to prove the correctness of these proofs, we will now argue that in proof systems with $\mathsf{NO\text{-}RED}$ we cannot derive any tautologies, while with RED we can at most derive universal tautologies.

Lemma 3.6. *It is not possible to create tautological clauses in any of the QCDCL proof systems with $\mathsf{NO\text{-}RED}$ during the derivation of learnable clauses as described in Definition 3.3. If we use the policy RED instead, we are at least able to avoid any tautologies with existential literals.*

Proof. Let $\mathcal{T}$ be a trail and $\mathcal{L}_{\mathcal{T}}$ be the sequence of learnable clauses as described in Definition 3.3. It suffices to concentrate on the case $C_{(i,j)}$, in particular

$$C_{(i,j)} = \text{red}[\text{res}(C_{(i,j+1)}, \text{red}(\text{ante}(p_{(i,j)})), p_{(i,j)})]$$

for $i \in \{0, \dots, r\}$, $j \in [g_i - 1]$. This is analogous to the case for $C_{(i,g_i)}$ (the case $C_{(r,g_r)}$ is trivial).

Assume there is a literal x with $\text{var}(x) \neq \text{var}(p_{(i,j)})$ such that $x \in C_{(i,j+1)}$ and $\bar{x} \in \text{red}(\text{ante}_{\mathcal{T}}(p_{(i,j)}))$. If $\mathsf{NO\text{-}RED}$ is chosen, we need $A|_{\mathcal{T}[i,j-1]} = (p_{(i,j)})$ for $A := \text{ante}_{\mathcal{T}}(p_{(i,j)})$ and therefore $x \in \mathcal{T}[i, j-1]$.

In case of RED we have $\text{red}(A|_{\mathcal{T}[i,j-1]}) = (p_{(i,j)})$. Because we can assume that x is existential in this case, we also conclude $x \in \mathcal{T}[i, j-1]$ since existential literals cannot be reduced.

On the other hand, we have $x \in C_{(i,j+1)}$, where $C_{(i,j+1)}$ is a learnable clause which is derived with the aid of antecedent clauses of literals occurring right of $p_{(i,j)}$ in the trail. In particular, we can find some $p_{(k,m)}$ right of $p_{(i,j)}$ in the trail with $x \in \text{ante}_{\mathcal{T}}(p_{(k,m)})$. Because of $x \in \mathcal{T}[i, j-1]$, this gives a contradiction since $\text{ante}_{\mathcal{T}}(p_{(k,m)})$ must not become true before propagating $p_{(k,m)}$. □

We have seen that systems with $\mathsf{NO\text{-}RED}$ cannot contain tautologies in their extracted proofs. It remains to show that the derivation of tautological clauses in systems with RED fulfils the properties of long-distance Q-resolution.

Proposition 3.7. *Let Φ 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 under a QCDCL proof system with the policy RED. Then the corresponding derivation of any learned clause is a valid long-distance Q-resolution *derivation.*

Proof. Let

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

be the sequence of learnable clauses. Because of Lemma 3.6 it remains to show that the derivation of clauses with universal tautologies are sound.

Assume otherwise. That means there are $i \in \{0, \dots, r\}$, $j \in [g_i]$ and $u \in \text{var}_{\forall}(\Phi)$ such that $\text{lv}(u) < \text{lv}(p_{(i,j)})$ and one of the following four cases holds, where we resolve over $p_{(i,j)}$:

(i) $u \in C_{(i,j+1)}$ and $\bar{u} \in \text{ante}_{\mathcal{T}}(p_{(i,j)})$,

(ii) $u \vee \bar{u} \in C_{(i,j+1)}$ and $\bar{u} \in \text{ante}_{\mathcal{T}}(p_{(i,j)})$,

(iii) $u \in C_{(i,j+1)}$ and $u \vee \bar{u} \in \text{ante}_{\mathcal{T}}(p_{(i,j)})$,

(iv) $u \vee \bar{u} \in C_{(i,j+1)}$ and $u \vee \bar{u} \in \text{ante}_{\mathcal{T}}(p_{(i,j)})$.

Consider case (i). Since $u \in C_{(i,j+1)}$ there has to be a propagated literal $p_{(k,m)}$ right of $p_{(i,j)}$ in the trail such that $u \in \text{ante}(p_{(k,m)})$. In order to become unit, the u in $\text{ante}(p_{(k,m)})$ needed to vanish. We distinguish two cases:

Case 1: $\bar{u}$ was decided before $p_{(k,m)}$ was propagated, i.e., $\bar{u} \in \mathcal{T}[k, m-1]$.

Then we have $u \notin \mathcal{T}$ since each variable can occur at most once in $\mathcal{T}$. That means reducing $\bar{u}$ is the only way $\text{ante}_{\mathcal{T}}(p_{(i,j)})$ could have become unit. But for the soundness of reduction we need $\text{lv}(u) > \text{lv}(p_{(i,j)})$. This gives a contradiction.

Case 2: $u \in \text{ante}_{\mathcal{T}}(p_{(k,m)})$ has vanished via reduction.

Assume that u was decided before $p_{(i,j)}$ was propagated, i.e., $u \in \mathcal{T}[i, j-1]$. But then $\text{ante}(p_{(k,m)})$ would have become true under $\mathcal{T}[k, m-1] \supseteq \mathcal{T}[i, j-1]$. Therefore $\text{ante}_{\mathcal{T}}(p_{(k,m)})$ could not have been used for unit propagation. Thus $\bar{u} \in \text{ante}_{\mathcal{T}}(p_{(i,j)})$ must have vanished via reduction, which implies $\text{lv}(u) > \text{lv}(p_{(i,j)})$, a contradiction.

The same reasoning works for case (ii). Cases (iii) and (iv) are easier since the only way for $u \vee \bar{u}$ to vanish in $\text{ante}_{\mathcal{T}}(p_{(i,j)})$ is via reduction. Then we get the contradiction $\text{lv}(u) > \text{lv}(p_{(i,j)})$ as well.

Also the same argumentation works if we consider $C_{(i+1,1)}$ and $\text{ante}_{\mathcal{T}}(p_{(i,g_i)})$ instead of $C_{(i,j+1)}$ and $\text{ante}_{\mathcal{T}}(p_{(i,j)})$. □

We combine the two results above to an argument of soundness of the defined QCDCL proof systems.

Theorem 3.8. *All defined QCDCL proof systems are p-simulated by* long-distance Q-resolution *and the systems with* NO-RED *are even p-simulated by* Q-resolution. *In fact, for a proof ι of a clause C in the QCDCL system we can get a* long-distance Q-resolution *(resp.* Q-resolution*) proof π with $|\pi| \in \mathcal{O}(|\iota|)$.*

In particular, all defined QCDCL proof systems are sound.

Proof. This follows directly from Lemma 3.6 and Proposition 3.7. Let

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

be a proof of a clause C in the QCDCL system. We get a long-distance Q-resolution (resp. Q-resolution) proof by sticking together the proofs $\pi_1, \ldots, \pi_m$. □

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 (we will discuss this later).

Definition 3.9 (asserting clauses and asserting learning schemes)**.** *Let $\Phi := \mathcal{Q} \cdot \phi$ be a QCNF in any of the defined QCDCL systems. Let*

$$\mathcal{T} = (p_{(0,1)}, \ldots, p_{(0,g_0)}; \mathbf{d_1}, p_{(1,1)}, \ldots, p_{(1,g_1)}; \ldots; \mathbf{d_r}, p_{(r,1)}, \ldots, p_{(r,g_r)} = \square)$$

be a trail which follows the corresponding policies and $\mathcal{L}_{\mathcal{T}}$ the sequence of learnable clauses. A clause $(\square) \neq C \in \mathcal{L}_{\mathcal{T}}$ is called an asserting clause, *if it becomes unit after backtracking, i.e., there exists a time (s, t) with $s \in \{0, \ldots, r-1\}$ 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 $\square \in \mathcal{T}$. 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 $(\square)$ as long as $\mathcal{L}_{\mathcal{T}}$ contains such.*

Remark 3.10. *It is not guaranteed that we will always find asserting clauses for our 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 $\mathcal{T} = (\boldsymbol{x}; \boldsymbol{u}, \square)$ under the system $\mathsf{QCDCL}^{\textsf{Any-Ord}}_{\textsf{No-Red}}$. We can only learn the clause $(\bar{u} \vee \bar{x})$, which is non-unit under $\mathcal{T}[0,0] = \emptyset$.*

Therefore despite allowing any decision order, we still need some restrictions in order to be able to learn asserting clauses.

Next we show that learning schemes return new clauses that have not been learned before. This fact will help us later to prove the completeness of our systems.

Lemma 3.11. *Let $\mathcal{T}$ be a natural trail for a QCNF $\Phi = \mathcal{Q} \cdot \phi$ in any of the defined QCDCL systems and $A \in \mathcal{L}_{\mathcal{T}}$ be an asserting clause. Then we have $A \notin \phi$.*

Proof. Let $\mathcal{T}$ be the trail

$$\mathcal{T} = (p_{(0,1)}, \ldots, p_{(0,g_0)}; \mathbf{d_1}, p_{(1,1)}, \ldots, p_{(1,g_1)}; \ldots; \mathbf{d_r}, p_{(r,1)}, \ldots, p_{(r,g_r)} = \square).$$

Since we learned an asserting clause, we obviously have $r > 0$ by definition.

Assume that $A \in \phi$. Then there is a time (s,t) with $s \in \{0, \ldots, r-1\}$ and $t \in [g_s]$ such that $A|_{\mathcal{T}[s,t]}$ is a unit clause with a literal ℓ. This literal is either existential or $\square$. Because of Definition 3.4 we are forced to propagate ℓ not later than level s. Since $\mathcal{T}$ has run into a conflict at level r, we could not get a conflict before level r, hence $\ell \neq \square$. Therefore $\ell = p_{(s,b)} \in \mathcal{T}$ for some $b \in [g_s]$. However, this is a contradiction because we need $\ell \in A$, which is only possible if $\bar{\ell} \in \mathcal{T}$ by definition of clause learning. □

Remark 3.12. *In general, the above result is not true for arbitrary trails. For example, let $\mathcal{T} = (\bar{\mathbf{x}}_\mathbf{1}; \bar{\mathbf{x}}_\mathbf{2}, \square)$ be a trail for the false QCNF $\exists x_1, x_2 \forall u \cdot (x_1 \vee x_2) \wedge (u)$. Then $x_1 \vee x_2$ is an asserting learnable clause, which is already an axiom.*

Now we know that learning schemes always return new clauses. We also need to examine the circumstances under which we are actually able to learn such asserting clauses.

Lemma 3.13. *Let $\Phi = \mathcal{Q} \cdot \phi$ be a QCNF and let*

$$\mathcal{T} = (p_{(0,1)}, \ldots, p_{(0,g_0)}; \mathbf{d_1}, p_{(1,1)}, \ldots, p_{(1,g_1)}; \ldots; \mathbf{d_r}, p_{(r,1)}, \ldots, p_{(r,g_r)} = \square)$$

be a trail under the policies $\textsf{Ass-Ord}$ *and* $\textsf{No-Red}$*. If $(\square) \notin \mathcal{L}_{\mathcal{T}}$, then there exists an asserting clause $D \in \mathcal{L}_{\mathcal{T}}$.*

Proof. Consider the sequence of learnable clauses:

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

The learning scheme $\mathtt{DEC}$ that maps each trail to the rightmost clause in $\mathcal{L}_{\mathcal{T}}$ is asserting. In particular we learn a clause $D := \operatorname{red}(C)$ such that C is a subclause of $\bar{d}_1 \vee \ldots \vee \bar{d}_r$. Suppose that $D \neq (\square)$. Write $D = \bar{d}_{h_1} \vee \ldots \vee \bar{d}_{h_a}$ for some $h_1 < \ldots < h_a$. Because of $\textsf{Ass-Ord}$ the last literal d_{h_a} will always be existential. Then we can backtrack to the time $(h_{a-1}, 0)$ (resp. $(0,0)$ if $a = 1$) and get $D|_{\mathcal{T}[h_{a-1},0]} = (\bar{d}_{h_a})$. □

The above result is not true for QCDCL systems with the policies $\mathsf{Ass\text{-}Ord}$ and Red. Consider the following counterexample: $\Phi := \exists x \forall u \exists z \cdot (\bar{x} \vee u \vee \bar{z})$ and the trail $\mathcal{T} = (\mathbf{x}; \mathbf{z}, \Box)$. We can only learn the clause $(\bar{x} \vee u \vee \bar{z})$. But since this is already an axiom, this clause cannot become unit at an earlier level.

However, the next lemma shows that under the policy $\mathsf{Ass\text{-}R\text{-}Ord}$ we can always learn asserting clauses.

Lemma 3.14. *Let $\Phi = \mathcal{Q} \cdot \phi$ be a QCNF and let*

$$\mathcal{T} = (p_{(0,1)}, \dots, p_{(0,g_0)}; \mathbf{d_1}, p_{(1,1)}, \dots, p_{(1,g_1)}; \dots; \mathbf{d_r}, p_{(r,1)}, \dots, p_{(r,g_r)} = \Box)$$

be a trail under the policies $\mathsf{Ass\text{-}R\text{-}Ord}$ and Red. If $(\Box) \notin \mathcal{L}_{\mathcal{T}}$, then there exists an asserting clause $D \in \mathcal{L}_{\mathcal{T}}$.

Proof. Consider the sequence of learnable clauses:

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

If $r = 0$ or if all decision literals are universal, we can just take the rightmost clause in $\mathcal{L}_{\mathcal{T}}$, which is $(\Box)$. Hence we can assume that $r > 0$ and there exists at least one existential decision literal.

Let $k \in [r]$ be maximal with respect that $\bar{d}_k$ is contained in some clause from $\mathcal{L}_{\mathcal{T}}$. This must exist since otherwise we could resolve over all propagation literals $p_{(i,j)}$ and reduce all universal literals during the learning process. Then we would be able to learn $(\Box)$.

Intuitively, d_k is the last existential decision that contributed to the conflict. Let $p_{(\ell,m)}$ be the next propagated literal right of d_k in $\mathcal{T}$ (this does not necessarily have to be $p_{(k,1)}$). Set $D := C_{(\ell,m)}$.

We claim that this clause D is asserting. Let us learn this clause and backtrack to $\mathcal{T}[k-1, g_{k-1}]$. It is easy to see that for all existential literals $y \in D \backslash \{\bar{d}_k\}$ we need $\bar{y} \in \mathcal{T}[k-1, g_{k-1}]$. All universal variables z with $\text{lv}_\Phi(z) < \text{lv}_\Phi(d_k)$ are assigned earlier than d_k in $\mathcal{T}$ because of the policy $\mathsf{Ass\text{-}R\text{-}Ord}$.

Now consider $E := D|_{\mathcal{T}[k-1,g_{k-1}]}$. The only type of literals that can occur in E, aside from $\bar{d}_k$, are universal literals u with $\text{lv}_\Phi(u) > \text{lv}_\Phi(d_k)$. Suppose there are such literals $u_1, \dots, u_m$ with $C = \bar{d}_k \vee u_1 \vee \dots \vee u_m$ and $\text{lv}_\Phi(d_k) < \text{lv}_\Phi(u_i)$ for all $i \in [m]$. But then we can conclude $\text{red}(E) = (\bar{d}_k)$ and therefore E is a unit clause. □

Now that we have clarified how to gain asserting clauses, we can finally prove the completeness of all systems.

Theorem 3.15. *All defined QCDCL proof systems are complete.*

Proof. We will concentrate on the systems $\mathsf{QCDCL}^{\mathsf{Lev\text{-}Ord}}_{\mathsf{Red}}$ and $\mathsf{QCDCL}^{\mathsf{Lev\text{-}Ord}}_{\mathsf{No\text{-}Red}}$ since all the other systems are just strengthenings of these two proof systems.

Let $\Phi = \mathcal{Q} \cdot \phi$ be a false QCNF over the set of variables V. W.l.o.g. we write the prefix as

$$\mathcal{Q} = \exists X_1 \forall U_1 \exists X_2 \forall U_2 \dots \exists X_m \forall U_m$$

with $X_1 \uplus \dots \uplus X_m \uplus U_1 \uplus \dots \uplus U_m = V$. This is possible if we allow empty sets.

Now because Φ is false, there exists a winning strategy for the $\forall$-player. That means we can find functions $f_i : \langle X_1 \cup \dots \cup X_i \rangle \to \langle U_i \rangle$ such that ϕ gets falsified under any assignment

$$\sigma_1 \cup f_1(\sigma_1) \cup \sigma_2 \cup f_2(\sigma_1 \cup \sigma_2) \cup \dots \cup \sigma_m \cup f_m(\sigma_1 \cup \dots \cup \sigma_m).$$

We can now construct a trail for Φ in the respective system. All the unit propagations can be done automatically. When we have to make decisions, we distinguish two cases:

Case 1: We have to decide an existential variable x.

We can choose an arbitrary polarity for this decision of x (e.g. set all variables to 1).

Case 2: We have to decide a universal variable.

Suppose we have to handle the variable $u \in U_i$. Since our trail follows the policy LEV-ORD, all variables from $X_1 \cup \ldots \cup X_i$ are already assigned. Let σ be the corresponding assignment. Then we decide u in the same polarity as it occurs in $f_i(\sigma)$.

After making the decisions we continue with unit propagation as usual. The trail that is generated this way represents an assignment as it gets created in the game between the two players. The $\forall$-player was able to use the winning strategy, therefore we falsify the matrix ϕ at some point and run into a conflict. After this we start clause learning where we can always learn an asserting clause C by Lemma 3.13 (resp. Lemma 3.14) until we learn the empty clause. By Lemma 3.11 we conclude that C is in fact a new clause. We add C to ϕ, restart (i.e., backtrack to $(0, 0)$) and start again. Since there are only finitely many clauses with variables in V, this process will end after finitely many runs. In the last run we have to learn the empty clause, hence we created a (possibly exponential-size) refutation

$$\iota = ((\mathcal{T}_1, \ldots, \mathcal{T}_n), (C_1, \ldots, C_n = (\Box)), (\pi_1, \ldots, \pi_n))$$

of Φ in the QCDCL system. □

4 Separating classic QCDCL and Q-resolution

In this section we will show that classic QCDCL (i.e., $\mathsf{QCDCL}^{\mathsf{LEV\text{-}ORD}}_{\mathsf{RED}}$) and Q-resolution are incomparable. This requires exponential separations in both directions between the two systems. This also motivates that when searching for a QCDCL algorithm that exploits the full power of Q-resolution (a topic we will address in Section 6), we will have to modify the policies in $\mathsf{QCDCL}^{\mathsf{LEV\text{-}ORD}}_{\mathsf{RED}}$.

We start the separation by introducing QBFs that will turn out to be easy for $\mathsf{QCDCL}^{\mathsf{LEV\text{-}ORD}}_{\mathsf{RED}}$, but hard for Q-resolution.

Definition 4.1 ([10])**.** *The QCNF* $\mathtt{QParity}_n$ *consists of the prefix* $\exists x_1 \ldots x_n \forall z \exists t_2 \ldots t_n$ *and the matrix*

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

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

One can interpret $\mathtt{QParity}_n$ as follows: Each t_i symbolizes the partial sum $x_1 \oplus x_2 \oplus \ldots \oplus x_i$. The last two clauses can only be true if t_n, which represents $x_1 \oplus x_2 \oplus \ldots \oplus x_n$, is neither 0 nor 1. Hence this formula is obviously false. It was shown in [10] that $\mathtt{QParity}_n$ needs exponential-size Q-resolution (and even QU-resolution) refutations.

We will demonstrate that $\mathtt{QParity}_n$ is in fact easy for $\mathsf{QCDCL}^{\mathsf{LEV\text{-}ORD}}_{\mathsf{RED}}$, i.e., for classical QCDCL.

Proposition 4.2. $\mathtt{QParity}_n$ *has polynomial-size* $\mathsf{QCDCL}^{\mathsf{LEV\text{-}ORD}}_{\mathsf{RED}}$ *refutations.*

Proof. We construct the trails $\theta(\iota) = (\mathcal{T}_n, \mathcal{U}_n, \mathcal{T}_{n-1}, \mathcal{U}_{n-1}, \ldots, \mathcal{T}_2, \mathcal{U}_2, \mathcal{T}_1, \mathcal{U}_1)$, but omit the other two components $\lambda(\iota)$ and $\rho(\iota)$ as it will be clear during the computation.

Let $\mathcal{T}_n$ be the trail

$$\mathcal{T}_n = (\mathbf{\bar{x}_1}; \mathbf{\bar{x}_2}, \bar{t}_2; \mathbf{\bar{x}_3}, \bar{t}_3; \ldots; \mathbf{\bar{x}_{n-1}}, \bar{t}_{n-1}; \mathbf{\bar{x}_n}, \bar{t}_n, \Box)$$

with antecedent clauses

$$\begin{aligned}
\mathrm{ante}_{\mathcal{T}_n}(\bar{t}_2) &= x_1 \vee x_2 \vee \bar{t}_2,\\
\mathrm{ante}_{\mathcal{T}_n}(\bar{t}_i) &= x_i \vee t_{i-1} \vee \bar{t}_i \text{ for } i = 3, \ldots, n,\\
\mathrm{ante}_{\mathcal{T}_n}(\square) &= t_n \vee z.
\end{aligned}$$

After resolving over $\bar{t}_n$ we can derive $C_n := x_n \vee t_{n-1} \vee z$ as a learnable clause and backtrack to $\mathcal{T}_n[n-2, 1]$.

Further, let $\mathcal{U}_n$ be the trail

$$\mathcal{U}_n = (\bar{\mathbf{x}}_\mathbf{1}; \bar{\mathbf{x}}_\mathbf{2}, \bar{t}_2; \bar{\mathbf{x}}_\mathbf{3}, \bar{t}_3; \ldots; \bar{\mathbf{x}}_\mathbf{n-2}, \bar{t}_{n-2}; \mathbf{x}_\mathbf{n-1}, t_{n-1}; \bar{\mathbf{x}}_\mathbf{n}, t_n, \square)$$

with antecedent clauses

$$\begin{aligned}
\mathrm{ante}_{\mathcal{U}_n}(\bar{t}_2) &= x_1 \vee x_2 \vee \bar{t}_2,\\
\mathrm{ante}_{\mathcal{U}_n}(\bar{t}_i) &= x_i \vee t_{i-1} \vee \bar{t}_i \text{ for } i = 3, \ldots, n-2,\\
\mathrm{ante}_{\mathcal{U}_n}(t_{n-1}) &= \bar{x}_{n-1} \vee t_{n-2} \vee t_{n-1},\\
\mathrm{ante}_{\mathcal{U}_n}(t_n) &= x_n \vee \bar{t}_{n-1} \vee t_n,\\
\mathrm{ante}_{\mathcal{U}_n}(\square) &= \bar{t}_n \vee \bar{z}.
\end{aligned}$$

Symmetrically to $\mathcal{T}_n$, we can learn the clause $D_n := x_n \vee \bar{t}_{n-1} \vee z$ by resolving over t_n and backtrack up to $\mathcal{U}_n[n-2, 1]$.

We continue with the construction of $\mathcal{T}_{n-1}$.

$$\mathcal{T}_{n-1} = (\bar{\mathbf{x}}_\mathbf{1}; \bar{\mathbf{x}}_\mathbf{2}, \bar{t}_2; \bar{\mathbf{x}}_\mathbf{3}, \bar{t}_3; \ldots; \bar{\mathbf{x}}_\mathbf{n-2}, \bar{t}_{n-2}; \bar{\mathbf{x}}_\mathbf{n-1}, \bar{t}_{n-1}, x_n, t_n, \square)$$

with antecedent clauses

$$\begin{aligned}
\mathrm{ante}_{\mathcal{T}_{n-1}}(\bar{t}_2) &= x_1 \vee x_2 \vee \bar{t}_2,\\
\mathrm{ante}_{\mathcal{T}_{n-1}}(\bar{t}_i) &= x_i \vee t_{i-1} \vee \bar{t}_i \text{ for } i = 3, \ldots, n-1,\\
\mathrm{ante}_{\mathcal{T}_{n-1}}(x_n) &= C_n = x_n \vee t_{n-1} \vee z,\\
\mathrm{ante}_{\mathcal{T}_{n-1}}(t_n) &= \bar{x}_n \vee t_{n-1} \vee t_n,\\
\mathrm{ante}_{\mathcal{T}_{n-1}}(\square) &= \bar{t}_n \vee \bar{z}.
\end{aligned}$$

After resolving over t_n, x_n and finally $\bar{t}_{n-1}$ we get the learned clause $C_{n-1} := x_{n-1} \vee t_{n-2} \vee z \vee \bar{z}$ and backtrack up to $\mathcal{T}_{n-1}[n-3, 1]$.

As before, we can symmetrically create the next trail $\mathcal{U}_{n-1}$.

$$\mathcal{U}_{n-1} = (\bar{\mathbf{x}}_\mathbf{1}; \bar{\mathbf{x}}_\mathbf{2}, \bar{t}_2; \bar{\mathbf{x}}_\mathbf{3}, \bar{t}_3; \ldots; \bar{\mathbf{x}}_\mathbf{n-3}, \bar{t}_{n-3}; \mathbf{x}_\mathbf{n-2}, t_{n-2}; \bar{\mathbf{x}}_\mathbf{n-1}, t_{n-1}, x_n, \bar{t}_n, \square)$$

with antecedent clauses

$$\begin{aligned}
\mathrm{ante}_{\mathcal{U}_{n-1}}(\bar{t}_2) &= x_1 \vee x_2 \vee \bar{t}_2,\\
\mathrm{ante}_{\mathcal{U}_{n-1}}(\bar{t}_i) &= x_i \vee t_{i-1} \vee \bar{t}_i \text{ for } i = 3, \ldots, n-3,\\
\mathrm{ante}_{\mathcal{U}_{n-1}}(t_{n-2}) &= \bar{x}_{n-2} \vee t_{n-3} \vee t_{n-2},\\
\mathrm{ante}_{\mathcal{U}_{n-1}}(t_{n-1}) &= x_{n-1} \vee \bar{t}_{n-2} \vee t_{n-1},\\
\mathrm{ante}_{\mathcal{U}_{n-1}}(x_n) &= D_n = x_n \vee \bar{t}_{n-1} \vee z,\\
\mathrm{ante}_{\mathcal{U}_{n-1}}(t_n) &= \bar{x}_n \vee \bar{t}_{n-1} \vee t_n,\\
\mathrm{ante}_{\mathcal{U}_{n-1}}(\square) &= \bar{t}_n \vee \bar{z}.
\end{aligned}$$

By resolving over $\bar{t}_n$, x_n and t_{n-1} we can derive $D_{n-1} := x_{n-1} \vee \bar{t}_{n-2} \vee z \vee \bar{z}$ and backtrack up to $\mathcal{U}_{n-1}[n-3, 1]$.

We now describe the general step. Let $j \in \{3, \dots, n-2\}$ and suppose we already learned the clauses $C_n, D_n, C_{n-1}, D_{n-1}, \dots, C_{j+1}, D_{j+1}$ with the aid of the earlier trails $\mathcal{T}_n, \mathcal{U}_n, \dots, \mathcal{T}_{j+1}, \mathcal{U}_{j+1}$, where $C_k := x_k \vee t_{k-1} \vee z \vee \bar{z}$ and $D_k := x_k \vee \bar{t}_{k-1} \vee z \vee \bar{z}$ for $k \in \{j+1, \dots, n-1\}$. Then we can construct $\mathcal{T}_j$ as follows:

$$\mathcal{T}_j = (\mathbf{\bar{x}_1}; \mathbf{\bar{x}_2}, \bar{t}_2; \mathbf{\bar{x}_3}, \bar{t}_3; \dots; \mathbf{\bar{x}_{j-1}}, \bar{t}_{j-1}; \mathbf{\bar{x}_j}, \bar{t}_j, x_{j+1}, t_{j+1}, x_{j+2}, \bar{t}_{j+2}, \dots, x_n, t_n^e, \square)$$

with $e \in \{0, 1\}$ such that $e \equiv n - j \pmod 2$ and antecedent clauses

$$\begin{aligned}
\mathrm{ante}_{\mathcal{T}_j}(\bar{t}_2) &= x_1 \vee x_2 \vee \bar{t}_2, \\
\mathrm{ante}_{\mathcal{T}_j}(\bar{t}_i) &= x_i \vee t_{i-1} \vee \bar{t}_i \text{ for } i = 3, \dots, j, \\
\mathrm{ante}_{\mathcal{T}_j}(x_{j+1}) &= C_{j+1} = x_{j+1} \vee t_j \vee z \vee \bar{z}, \\
\mathrm{ante}_{\mathcal{T}_j}(t_{j+1}) &= \bar{x}_{j+1} \vee t_j \vee t_{j+1}, \\
\mathrm{ante}_{\mathcal{T}_j}(x_{j+2}) &= D_{j+2} = x_{j+2} \vee \bar{t}_{j+1} \vee z \vee \bar{z}, \\
\mathrm{ante}_{\mathcal{T}_j}(\bar{t}_{j+2}) &= \bar{x}_{j+2} \vee \bar{t}_{j+1} \vee \bar{t}_{j+2}, \\
&\vdots \\
\mathrm{ante}_{\mathcal{T}_j}(x_n) &= \begin{cases} C_n & = x_n \vee t_{n-1} \vee \bar{z}, \text{ if } n-j \text{ is odd}, \\ D_n & = x_n \vee \bar{t}_{n-1} \vee z, \text{ if } n-j \text{ is even}, \end{cases} \\
\mathrm{ante}_{\mathcal{T}_j}(t_n^e) &= \bar{x}_n \vee t_{n-1}^e \vee t_n^e, \\
\mathrm{ante}_{\mathcal{T}_j}(\square) &= t_n^{1-e} \vee z^{1-e}.
\end{aligned}$$

We resolve over $t_n^e, x_n, \dots, t_{j+1}, x_{j+1}$ and $\bar{t}_j$, derive $C_j := x_j \vee t_{j-1} \vee z \vee \bar{z}$ and backtrack up to $\mathcal{T}_j[j-2, 1]$.

In a similar way we will create $\mathcal{U}_j$:

$$\mathcal{U}_j = (\mathbf{\bar{x}_1}; \mathbf{\bar{x}_2}, \bar{t}_2; \mathbf{\bar{x}_3}, \bar{t}_3; \dots; \mathbf{\bar{x}_{j-2}}, \bar{t}_{j-2}; \mathbf{x_{j-1}}, t_{j-1}; \mathbf{\bar{x}_j}, t_j, x_{j+1}, \bar{t}_{j+1}, x_{j+2}, t_{j+2}, \dots, x_n, t_n^f, \square)$$

with $f \in \{0, 1\}$ such that $f \equiv n - j + 1 \pmod 2$ and antecedent clauses

$$\begin{aligned}
\mathrm{ante}_{\mathcal{U}_j}(\bar{t}_2) &= x_1 \vee x_2 \vee \bar{t}_2, \\
\mathrm{ante}_{\mathcal{U}_j}(\bar{t}_i) &= x_i \vee t_{i-1} \vee \bar{t}_i \text{ for } i = 3, \dots, j-2, \\
\mathrm{ante}_{\mathcal{U}_j}(t_{j-1}) &= \bar{x}_{j-1} \vee t_{j-2} \vee t_{j-1}, \\
\mathrm{ante}_{\mathcal{U}_j}(t_j) &= x_j \vee \bar{t}_{j-1} \vee t_j, \\
\mathrm{ante}_{\mathcal{U}_j}(x_{j+1}) &= D_{j+1} = x_{j+1} \vee \bar{t}_j \vee z \vee \bar{z}, \\
\mathrm{ante}_{\mathcal{U}_j}(\bar{t}_{j+1}) &= \bar{x}_{j+1} \vee \bar{t}_j \vee \bar{t}_{j+1}, \\
\mathrm{ante}_{\mathcal{U}_j}(x_{j+2}) &= C_{j+2} = x_{j+2} \vee t_{j+1} \vee z \vee \bar{z}, \\
\mathrm{ante}_{\mathcal{U}_j}(t_{j+2}) &= \bar{x}_{j+2} \vee t_{j+1} \vee \bar{t}_{j+2}, \\
&\vdots \\
\mathrm{ante}_{\mathcal{U}_j}(x_n) &= \begin{cases} C_n & = x_n \vee t_{n-1} \vee \bar{z}, \text{ if } n-j \text{ is even}, \\ D_n & = x_n \vee \bar{t}_{n-1} \vee z, \text{ if } n-j \text{ is odd}, \end{cases} \\
\mathrm{ante}_{\mathcal{U}_j}(t_n^f) &= \bar{x}_n \vee t_{n-1}^f \vee t_n^f, \\
\mathrm{ante}_{\mathcal{U}_j}(\square) &= t_n^{1-f} \vee z^{1-f}.
\end{aligned}$$

Resolving over $t_n^f, x_n, \dots \bar{t}_{j+1}, x_{j+1}$ and t_j gives us $D_j := x_j \vee \bar{t}_{j-1} \vee z \vee \bar{z}$. We backtrack to $\mathcal{U}_j[j-2, 1]$.

We end the refutation with the following four trails whose antecedent clauses are almost as before:

$$\mathcal{T}_2 = (\bar{\mathbf{x}}_\mathbf{1}; \bar{\mathbf{x}}_\mathbf{2}, \bar{t}_2, x_3, t_3, x_4, \bar{t}_4, \dots, \square),$$

from which we learn $C_2 := x_1 \vee x_2$, and

$$\mathcal{U}_2 = (\bar{\mathbf{x}}_\mathbf{1}, x_2, t_2, x_3, \bar{t}_4, \dots, \square),$$

where we can derive the unit clause (x_1). We continue with

$$\mathcal{T}_1 = (x_1; \bar{\mathbf{x}}_\mathbf{2}, t_2, x_3, \bar{t}_3, x_4, t_4, \dots, \square),$$

learn (x_2), and can finally derive the empty clause $(\square)$ via the last trail

$$\mathcal{U}_1 = (x_1, x_2, \bar{t}_2, x_3, t_3, x_4, \bar{t}_4, \dots, \square).$$

□

For the separation in the other direction we need CNFs that are hard for general resolution. One of the famous examples is the pigeonhole principle, but also any other formula that is hard for propositional resolution would serve the purpose.

Definition 4.3. *The pigeonhole principle* $\mathtt{PHP}^m_n$ *is a propositional CNF consisting of the variables* $x_{i,j}$*, for* $i \in [m]$ *and* $j \in [n]$*, and the clauses*

$$\bigvee_{k \in [n]} x_{i,k}$$
$$\bar{x}_{i_1,j} \vee \bar{x}_{i_2,j}$$

for all $i, i_1, i_2 \in [m]$*,* $i_1 \neq i_2$ *and* $j \in [n]$*.*

Proposition 4.4 (Haken [26])**.** *The CNFs* $\mathtt{PHP}^{n+1}_n$ *are unsatisfiable and require exponential-size* resolution *refutations.*

We embed $\mathtt{PHP}^{n+1}_n$ into a QCNF which we will call $\mathtt{Trapdoor}_n$. Intuitively, if we have chosen RED, we are forced to reach a conflict in the propositionally hard formula $\mathtt{PHP}^{n+1}_n$. However, forbidding reduction by choosing the policy NO-RED allows us to avoid the pigeonhole principle and instead derive a conflict in part that is easier to refute.

Definition 4.5. *Let* $\mathtt{PHP}^{n+1}_n$ *be the set of clauses for the pigeonhole principle with parameters* n *and* $n+1$ *in the variables* $x_1, \dots, x_{s_n}$*. Let* $\mathtt{Trapdoor}_n$ *be the QCNF in the variables* $x_1, \dots, x_{s_n}, y_1, \dots, y_{s_n}, u, t, w$ *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

$$\mathtt{PHP}^{n+1}_n(x_1, \dots, x_{s_n})$$
$$\bar{y}_i \vee x_i \vee u, \;\; y_i \vee \bar{x}_i \vee u$$
$$y_i \vee w \vee t, \;\; y_i \vee w \vee \bar{t}, \;\; \bar{y}_i \vee w \vee t, \;\; \bar{y}_i \vee w \vee \bar{t}$$

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

The next result shows the hardness for $\mathtt{Trapdoor}_n$ in $\mathsf{QCDCL}^{\text{LEV-ORD}}_{\text{RED}}$. It is clear that this hardness is directly caused by $\mathtt{PHP}^{n+1}_n$. However, it remains to prove that it is still retained by the embedding.

Proposition 4.6. *The QCNFs* $\texttt{Trapdoor}_n$ *require exponential-size* $\mathsf{QCDCL}_{\mathsf{RED}}^{\mathsf{LEV\text{-}ORD}}$ *refutations.*

Proof. Each $\mathsf{QCDCL}_{\mathsf{RED}}^{\mathsf{LEV\text{-}ORD}}$ trail for $\texttt{Trapdoor}_n$ starts with some decisions of y variables. Before reaching the variables w and t, we can only have propagated x or y variables. Since PHP_n^{n+1} is unsatisfiable, we will reach a conflict before deciding w. The learned clauses will only contain x, y and u variables, hence the same situation will happen after restarting which leads to a refutation that does not use the last four types of axioms in $\texttt{Trapdoor}_n$.

Therefore we have gained a long-distance Q-resolution refutation π'' of $\texttt{Trapdoor}_n$ which only makes use of the axioms

$$\begin{aligned}&\mathsf{PHP}_n^{n+1}(x_1, \dots, x_{s_n})\\&\bar{y}_i \vee x_i \vee u\\&y_i \vee \bar{x}_i \vee u.\end{aligned}$$

Define $\psi' := \{\bar{y}_i \vee x_i \vee u, y_i \vee \bar{x}_i \vee u : i = 1, \dots, s_n\}$. We can construct a Q-resolution refutation π' by reducing the u-variable right after any introduction of a clause from ψ' and performing the same resolution steps as in π'' afterwards. Then $|\pi'| \in \mathcal{O}(|\pi''|)$.

Now deleting any axiom of ψ' from π' results in a resolution refutation π of the unsatisfiable CNF

$$\begin{aligned}&\mathsf{PHP}_n^{n+1}(x_1, \dots, x_{s_n})\\&\bar{y}_i \vee x_i\\&y_i \vee \bar{x}_i\end{aligned}$$

with $|\pi| \in \mathcal{O}(|\pi'|)$.

Let $\psi = \{\bar{y}_i \vee x_i, y_i \vee \bar{x}_i : i = 1, \dots, s_n\}$. Next we will create a resolution refutation μ of PHP_n^{n+1}.

Let π consist of the clauses $C'_1, \dots, C'_m$. W.l.o.g. we can assume that none of these clauses are of the form $D \vee \bar{y}_i \vee x_i$ or $D \vee y_i \vee \bar{x}_i$ for a non-empty subclause D, because otherwise we can shorten the refutation by taking the corresponding axioms in ψ instead.

Also let f be the function on clauses that replaces all occurrences of y_i (resp. $\bar{y}_i$) in a clause with x_i (resp. $\bar{x}_i$). We will show that the proof μ we get by deleting all clauses of π contained in ψ and replacing all other clauses C by $f(C)$ is a correct resolution refutation.

Let C'_ℓ be a clause in π. If C'_ℓ is an axiom, then either $C'_\ell \in \mathsf{PHP}_n^{n+1}$ and therefore $C_\ell := f(C'_\ell) = C'_\ell$, or $C'_\ell \in \psi$, in which case we delete C'_ℓ (or replace it with a placeholder in μ).

If C'_ℓ was not an axiom, then we can find two parental clauses C'_j and C'_k of C'_ℓ. We distinguish two cases.

Case 1: One of the clauses C'_j, C'_k is from ψ.

W.l.o.g. let $C'_j \in \psi$ and $C'_k \notin \psi$. Note that it is not possible for both clauses to be contained in ψ. By induction we know that $C_k := f(C'_k) \in \mu$. Since C'_j was deleted during the transition into μ, we can only set $C_\ell := C_k = f(C'_k)$. Because resolving with clauses of ψ just swap x_i-variable and y_i-variables, we immediately get $f(C'_k) = f(C'_\ell)$, hence $C_\ell = f(C'_\ell)$.

Case 2: None of the clauses C'_j, C'_k are contained in ψ.

Then there exists $C_j := f(C'_j) \in \mu$ and $C_k := f(C'_k) \in \mu$. Let $C'_\ell = \operatorname{res}(C'_j, C'_k, z)$ with $z \in \{x_i, y_i\}$ for an $i \in [s_n]$. We set $C_\ell := \operatorname{res}(C_j, C_k, x_i)$. The only chance for this resolution to become unsound (with respect to resolution) is w.l.o.g. $x_a \in C'_j$ and $\bar{y}_a \in C'_k$ for an $i \neq a \in \{1, \dots, s_n\}$. Then we would receive $x_a \vee \bar{y}_a$ in C'_ℓ and a tautology $x_a \vee \bar{x}_a$ in C_ℓ. However, this cannot happen due to our assumption that the clauses of ψ are no subclauses of C'_j, C'_k, C'_ℓ. It is easy to see that $C_\ell = f(C'_\ell)$.

We now have constructed a resolution refutation μ with $|\mu| \in \mathcal{O}(|\pi|) = \mathcal{O}(|\pi''|)$ which only uses the clauses of PHP_n^{n+1} as axioms. By Proposition 4.4 the formula PHP_n^{n+1} needs exponential sized resolution refutations. Therefore $|\pi''| \in 2^{\Omega(n)}$, which is the size of the corresponding $\mathsf{QCDCL}_{\mathsf{RED}}^{\mathsf{LEV\text{-}ORD}}$ refutation as well. □

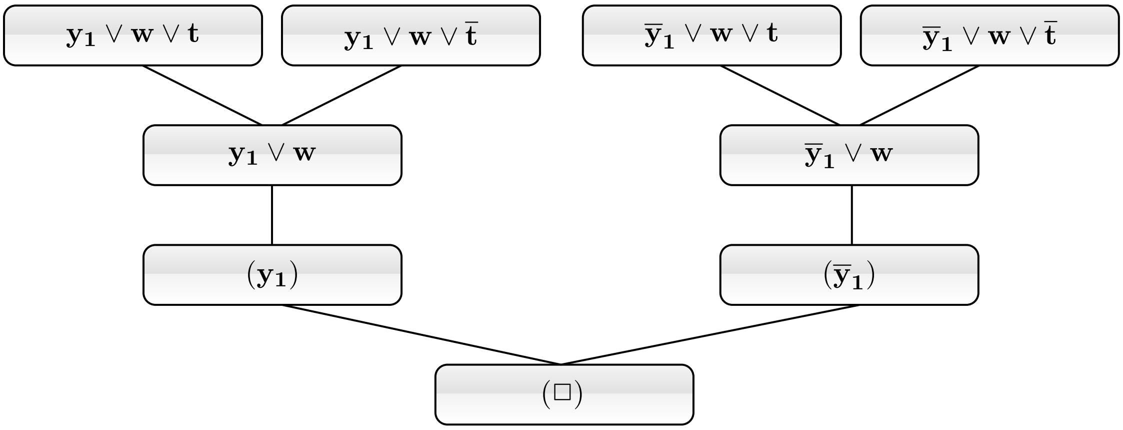

$\mathbf{y_1} \vee \mathbf{w} \vee \mathbf{t}$
$\mathbf{y_1} \vee \mathbf{w} \vee \bar{\mathbf{t}}$
$\bar{\mathbf{y}}_1 \vee \mathbf{w} \vee \mathbf{t}$
$\bar{\mathbf{y}}_1 \vee \mathbf{w} \vee \bar{\mathbf{t}}$
$\mathbf{y_1} \vee \mathbf{w}$
$\bar{\mathbf{y}}_1 \vee \mathbf{w}$
$(\mathbf{y_1})$
$(\bar{\mathbf{y}}_1)$
$(\Box)$

Figure 3: Short refutations of $\texttt{Trapdoor}_n$ in Q-resolution

We remark that the hardness of $\texttt{Trapdoor}_n$ crucially depends on propositional hardness. Note that it is not possible to just substitute $\texttt{PHP}_n^{n+1}$ in $\texttt{Trapdoor}_n$ by some QCNF that is hard for long-distance Q-resolution or $\mathsf{QCDCL}^{\textsf{Lev-Ord}}_{\textsf{Red}}$ since it is not guaranteed that the conflict in the trail will occur in this embedded formula.

In contrast, the $\texttt{Trapdoor}_n$ formulas are easy in Q-resolution.

Proposition 4.7. *The QCNFs* $\texttt{Trapdoor}_n$ *have constant-size* Q-resolution *refutations.*

Proof. The refutation is given in Figure 3. □

These two results immediately lead to the following separation.

Theorem 4.8. *The systems* Q-resolution *and* $\mathsf{QCDCL}^{\textsf{Lev-Ord}}_{\textsf{Red}}$ *are incomparable.*

From Theorem 4.8 we can conclude not only that we have to modify the QCDCL proof system if we aim to characterise Q-resolution, it also implies that a simple strengthening (or weakening) of one of the two systems cannot result in the desired equivalence. We mentioned earlier that the policies RED and NO-RED seem to operate orthogonally to each other. In Section 8 we will get back to this point and formally prove this intuition. This also motivates why switching from policy RED to NO-RED can be helpful in obtaining a QCDCL system that characterises Q-resolution.

5 Hard formulas for QCDCL

As we have shown in Section 4, Q-resolution is incomparable to QCDCL. This leaves open the question of what formulas are hard for QCDCL, without relying on hardness of Q-resolution or long-distance Q-resolution.

Definition 5.1. *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* $\mathsf{QCDCL}^{\textsf{Lev-Ord}}_{\textsf{Red}}$ *proof ι of C from Φ such that the* long-distance Q-resolution *proof π is obtained by pasting together the sub-proofs $(\pi_1, \dots, \pi_m)$ from ι (cf. Definition 3.5).*

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 $\mathsf{QCDCL}^{\textsf{Lev-Ord}}_{\textsf{Red}}$ derivations. By definition therefore, long-distance QCDCL resolution and $\mathsf{QCDCL}^{\textsf{Lev-Ord}}_{\textsf{Red}}$ are p-equivalent proof systems.

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

Definition 5.2. *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 $var(C) \cap X = \emptyset$, $var(C) \cap U = \emptyset$ and $var(C) \cap T \neq \emptyset$,*
- XT-clause, *if $var(C) \cap X \neq \emptyset$, $var(C) \cap U = \emptyset$ and $var(C) \cap T \neq \emptyset$,*
- XUT-clause, *if $var(C) \cap X \neq \emptyset$, $var(C) \cap U \neq \emptyset$ and $var(C) \cap T \neq \emptyset$.*

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.*

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

Lemma 5.3. *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 5.4. *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.*

Proof. Assume that there is a T-clause C, which is not an axiom, that was derived from Φ by a long-distance Q-resolution proof π. Let D be the first T-clause in π that is not an axiom. As in the proof of the previous lemma, D could not have been derived via reduction.

This means that D is again a resolvent of two clauses $D_1, D_2 \in \pi$. If we resolve over an X-literal, then D_1 or D_2 has to be an XT-clause, which is not possible by Lemma 5.3. Otherwise, if this is a resolution over a T-literal, then D_1 and D_2 have to be both T-clauses. One of them is not an axiom because Φ has the XT-property. This contradicts the fact, that D was the first non-axiomatic T-clause in π. □

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 5.5. *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 $\mathsf{QCDCL}^{\mathsf{Lev\text{-}Ord}}_{\mathsf{Red}}$ 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 decided 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 5.4. 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 5.3.

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 $E' := \text{ante}_{\mathcal{T}}(\ell)$ 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. □

We combine the above results and show that the hardness of QCNFs with the XT-property lifts from Q-resolution to long-distance QCDCL resolution.

Theorem 5.6. *If Φ fulfils the XT-property and requires* Q-resolution *refutations of size s, then each* long-distance QCDCL resolution *refutation (and therefore also each* $\text{QCDCL}_{\textsf{RED}}^{\textsf{LEV-ORD}}$ *refutation) of Φ has at least size s as well. In detail, each* long-distance QCDCL resolution *refutation of Φ is in fact a* Q-resolution *refutation.*

Proof. Let π be a long-distance QCDCL resolution refutation of Φ. 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 5.5.

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. □

One can conclude that QCNFs with the XT-property that are hard for Q-resolution are also hard for long-distance Q-resolution and therefore QCDCL. We will give some examples for these cases.

Definition 5.7 ([6])**.** *The formula* $\texttt{Equality}_n$ *is defined as the QCNF*

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

This QCNF is obviously false since the $\forall$-player has a winning strategy by assigning each u_i equal to the assignment of x_i.

Theorem 5.8 ([6])**.** $\texttt{Equality}_n$ *requires* Q-resolution *refutations of size* 2^n.

It is easy to see that $\texttt{Equality}_n$ fulfils the XT-property for $n \geq 2$. Therefore we obtain:

Corollary 5.9. $\texttt{Equality}_n$ *requires* $\text{QCDCL}_{\textsf{RED}}^{\textsf{LEV-ORD}}$ *refutations of size* 2^n.

Since the equality formulas are easy in long-distance Q-resolution [7], we obtain an exponential separation between QCDCL and long-distance Q-resolution.

Corollary 5.10. 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.

Next we will define a whole class of randomly generated QCNFs. With high probability, they also serve as hard examples for QCDCL.

Definition 5.11 ([6]). *For each $1 \leq i \leq n$ let $C_i^{(1)}, \ldots, C_i^{(cn)}$ be clauses picked uniformly at random from the set of clauses containing 1 literal from the set $U_i = \{u_i^{(1)}, \ldots, u_i^{(m)}\}$ and 2 literals from $X_i = \{x_i^{(1)}, \ldots, x_i^{(n)}\}$. Define the randomly generated QCNF $Q(n, m, c)$ as:*

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

Suitably choosing the parameters c and m, we gain false and indeed hard formulas.

Theorem 5.12 ([6]). *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* Q-resolution *refutations of size $2^{\Omega(n^\epsilon)}$.*

Again, it is easy to see that all $Q(n,m,c)$-formulas fulfil the XT-property.

Corollary 5.13. *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* $\mathsf{QCDCL}_{\mathsf{RED}}^{\mathsf{LEV\text{-}ORD}}$ *refutations of size $2^{\Omega(n^\epsilon)}$.*

The hardness of $\texttt{Equality}_n$ and the random formulas does not rely on propositional hardness, in contrast to $\texttt{Trapdoor}_n$, for example. In order to make the notion of 'propositional hardness' formal, we will use a strengthening of Q-resolution that allows oracle calls. This framework was introduced in [13] and tailored towards Q-resolution in [8]. The oracle allows to collapse arbitrary propositional sub-derivations into just one inference step.

Definition 5.14 ([8]). Q^{NP}-resolution *is defined as* Q-resolution*, but the resolution rule is replaced by the following:*

- $\Sigma_1^{\exists}$-rule: *For some $\mathcal{G} \subseteq \{C_1, \ldots, C_{i-1}\}$,*

  1. *$\bigwedge_{B \in \mathcal{G}} B^{\exists} \vDash C_i^{\exists}$, and*
  2. *for each $B \in \mathcal{G}$, $B^{\forall}$ is a subclause from $C_i^{\forall}$,*

  where $C^{\exists}$ and $C^{\forall}$ denote the existential and universal subclauses of any clause C.

We use the notation $C_1 \wedge \ldots \wedge C_{i-1} \vDash_{\Sigma_1^{\exists}} C_i$ for referring to such a step.

In fact, the lower bounds for the equality and random formulas above hold in this stronger model.

Theorem 5.15 ([6, 8]). $\texttt{Equality}_n$ *requires* Q^{NP}-resolution *refutations of size 2^n. Likewise, the random formulas $Q(n,m,c)$ (with the same parameters as in Theorem 5.12) are false and require* Q^{NP}-resolution *refutations of size $2^{\Omega(n^\epsilon)}$.*

An equivalent way of stating this theorem is to say that the equality and random formulas require exponentially many reduction steps in Q-resolution proofs. This measure is also applicable to QCDCL trails and in particular to long-distance QCDCL resolution proofs.

Proposition 5.16. *The number of reduction steps in each* long-distance QCDCL resolution *refutation (and also each* $\mathsf{QCDCL}_{\mathsf{RED}}^{\mathsf{LEV\text{-}ORD}}$ *refutation) of $\texttt{Equality}_n$ is at least 2^n. The same holds for the false formulas $Q(n,m,c)$ with $2^{\Omega(n^\epsilon)}$ reduction steps.*

Proof. Let $\pi = C_1, \dots, C_m$ be a long-distance QCDCL resolution refutation of $\texttt{Equality}_n$ or $Q(n, m, c)$ and let r be the number of clauses in π that were directly derived by reduction of some preceding clause. By Proposition 5.6, π is even a Q-resolution refutation. W.l.o.g. all axioms are introduced at the beginning of π. I.e., for an $\ell \in \mathbb{N}$ we have that $C_1, \dots, C_\ell$ are all axioms from $\texttt{Equality}_n$. This does not change the size of π as we need all axioms anyway.

We can assume that π is in the form of

$$\pi = C_1, \dots, C_\ell, R_{(1,1)}, \dots, R_{(1,s_1)}, T_1, R_{(2,1)}, \dots, R_{(r,1)}, \dots, R_{(r,s_r)}, T_r, R_{(r+1,1)}, \dots, R_{(r+1,s_{r+1})},$$

where each $R_{(i,j)}$ is (directly) derived by the resolution rule, and for each T_i we have $T_i = \text{red}(R_{(i,s_i)})$. One can inductively show that for each $R_{(c,d)}$ we have

$$\bigwedge_{i=1}^{\ell} C_i \wedge \bigwedge_{j=1}^{c-1} T_j \vDash_{\Sigma_1^\exists} R_{(c,d)}$$

due to

$$\bigwedge_{i=1}^{\ell} C_i \wedge \bigwedge_{j=1}^{c-1} T_j \vDash_{\Sigma_1^\exists} R_{(a,b)}$$

for every $R_{(a,b)}$ left of $R_{(c,d)}$ in π.

We can replace consecutive resolution steps with the $\Sigma_1^\exists$-rule and obtain a Q^{NP}-resolution refutation π' with the same number of reductions as π. We have $|\pi'| \in 2^{\Omega(n)}$ (resp. $2^{\Omega(n^\epsilon)}$ for $Q(n, m, c)$) by Proposition 5.15. This new refutation π' is of the form

$$\pi' = C_1, \dots, C_\ell, R_{(1,s_1)}, T_1, R_{(2,s_2)}, T_2, \dots, R_{(r,s_r)}, T_r, R_{(r+1,s_{r+1})}.$$

I.e., these two kinds of derivation steps are alternating after C_ℓ. Therefore the number of reductions r of π' (and also π) can be estimated by

$$r \in \Omega\left(\frac{|\pi'| - \ell}{2}\right) = \Omega(|\pi'|).$$

□

Compared to $\texttt{Equality}_n$, the formulas $\texttt{Trapdoor}_n$ just need a linear number of reduction steps in $\mathsf{QCDCL}^{\textsf{Lev-Ord}}_{\textsf{Red}}$.

6 A QCDCL system equivalent to Q-resolution

We now show that the QCDCL system $\mathsf{QCDCL}^{\textsf{Ass-Ord}}_{\textsf{No-Red}}$ has exactly the right strength to characterise Q-resolution.

Our central notion in this section will be that of the *reliability* of a clause. The motivation for this definition is as follows: In our QCDCL systems we generate trails and proofs in a natural way, i.e., we are not allowed to skip propagations or conflicts. Nevertheless, when aiming to simulate Q-resolution proofs, we also want to prescribe a sequence of literals as decisions in order to create trails along these decisions. However, it is not guaranteed that we can even make all these decisions without problems. We could run into situations that prevent us from continuing with the desired decisions. To classify these situations, we use the notion of reliability.

Definition 6.1. *Let $\Phi = \mathcal{Q} \cdot \phi$ be a QCNF and C be a non-tautological clause. If there is a $\mathsf{QCDCL}^{\textsf{Ass-Ord}}_{\textsf{No-Red}}$ trail $\mathcal{T}$, an existential literal $\ell \in C$ and a set of literals $\alpha \subseteq \bar{C} \backslash \{\bar{\ell}\}$ 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.

Remark 6.2. *Let $\Phi = \mathcal{Q} \cdot \phi$ be a QCNF and let C be reliable with respect to Φ. Then we can construct a trail $\mathcal{T}$ by choosing $\bar{C}$ as decisions one by one and doing the propagations more or less automatically. These decisions cannot block each other. However, it is possible that we propagate a literal from $\bar{C}$ in the same polarity before deciding it. In this case we have to skip the decision. Also, we could reach a conflict before deciding all literals, then we abort the trail as usual. Both of these cases are still fine for our purposes. We stop the construction when we either reach a conflict, or if all literals from $\bar{C}$ are assigned and we cannot make further propagations.*

We will use this technique in our later proofs and refer to it as constructing $\mathcal{T}$ with decisions *$\bar{C}$. Note that all trails we construct in this way are natural.*

Example 6.3. *We give some minimal examples of the situations of Remark 6.2 in the system $\mathsf{QCDCL}^{\mathit{Ass\text{-}Ord}}_{\mathit{No\text{-}Red}}$:*

1. *Let $\Phi := \exists x \forall u \exists y, z \cdot (\bar{x} \vee \bar{u} \vee \bar{y} \vee z)$. Constructing a trail with decisions (x, u, y) in this order gives us $\mathcal{T} := (\boldsymbol{x}; \boldsymbol{u}; \boldsymbol{y}, z)$. In this situation the decisions (x, u, y) are not blocking each other. In fact, the clause $(\bar{x} \vee \bar{u} \vee \bar{y})$ is reliable with respect to Φ since we would need to decide $\bar{z}$ in order to propagate $\bar{x}$ or $\bar{y}$.*

2. *Let $\Phi := \exists x \forall u \exists y, z \cdot (\bar{x} \vee \bar{u} \vee \bar{y} \vee z) \wedge (\bar{x} \vee \bar{u} \vee y)$. After constructing a trail $\mathcal{T}$ with decisions (x, u, y) we get $\mathcal{T} = (\boldsymbol{x}; \boldsymbol{u}, y, z)$. This is still fine even if we did not actually use y as a decision. The clause $\bar{x} \vee \bar{u} \vee \bar{y}$ is still reliable with respect to Φ because we would need to decide $\bar{z}$ or $\bar{y}$ in order to imply $\bar{x}$.*

3. *Let $\Phi := \exists x \forall u \exists y, z \cdot (\bar{x} \vee \bar{u} \vee \bar{y} \vee z) \wedge (\bar{x} \vee \bar{u} \vee z) \wedge (\bar{x} \vee \bar{u} \vee \bar{z})$. When we want to construct a trail with decisions (x, u, y) we get $\mathcal{T} = \{\boldsymbol{x}, \boldsymbol{u}, z, \square\}$. This is fine even though y does not appear in $\mathcal{T}$, since we run into a conflict beforehand. The clause $\bar{x} \vee \bar{u} \vee \bar{y}$ is reliable with respect to Φ, because the only way to achieve a situation of unreliability would be implying $\bar{x}$ with decisions u and y, or implying $\bar{y}$ with decisions x and u. In the first case we would not get any unit clauses, and in the second one we would propagate $\square$ before $\bar{y}$.*

4. *Let $\Phi := \exists x \forall u \exists y, z \cdot (\bar{x} \vee \bar{u} \vee z) \wedge (\bar{z} \vee \bar{y})$. When we try to construct a trail $\mathcal{T}$ with decisions (x, u, y) we get stuck at $\mathcal{T} = (\boldsymbol{x}; \boldsymbol{u}, z, \bar{y})$. Now we have propagated $\bar{y}$ although we wanted to decide y. This shows that these decisions block each other and the clause $(\bar{x} \vee \bar{u} \vee \bar{y})$ is unreliable with respect to Φ and $\mathcal{T}$ serves as a witness.*

The next lemma shows that we can basically 'copy' a trail that was created at a previous point by deciding or propagating all of its decisions. This will help us later during the simulation of resolution or reductions steps, where we want to copy the trails that serve as a witness for the unreliability of the parent clauses.

Lemma 6.4. *Let $\Phi = \mathcal{Q} \cdot \phi$ and $\Psi = \mathcal{Q} \cdot \psi$ be two QCNFs with the same prefix such that $\phi \subseteq \psi$. Let $\mathcal{T}$ be a $\mathsf{QCDCL}^{\mathit{Ass\text{-}Ord}}_{\mathit{No\text{-}Red}}$ trail for Φ and $\mathcal{U}$ be a natural trail for Ψ. Let α be the decisions in $\mathcal{T}$ and $\alpha \subseteq \mathcal{U}$. If $\mathcal{U}$ does not run into a conflict, then every propagated literal from $\mathcal{T}$ is contained in $\mathcal{U}$.*

Proof. Suppose that $\mathcal{U}$ did not run into a conflict.

Write $\mathcal{T}$ as

$$\mathcal{T} = (p_{(0,1)}, \ldots, p_{(0,g_0)}; \mathbf{d_1}, p_{(1,1)}, \ldots, p_{(1,g_1)}; \ldots; \mathbf{d_r}, p_{(r,1)}, \ldots, p_{(r,g_r)}).$$

Assume that there are some propagated literals from $\mathcal{T}$ that are not contained in $\mathcal{U}$. Let $p_{(i,j)}$ be the first (leftmost) literal in $\mathcal{T}$ of this kind. Then we know that

$$\text{ante}_{\mathcal{T}}(p_{(i,j)})|_{\mathcal{T}[i,j-1]} = (p_{(i,j)})$$

with $\text{ante}_{\mathcal{T}}(p_{(i,j)}) \in \phi \subseteq \psi$. Since $p_{(i,j)}$ was the first propagated literal not contained in $\mathcal{U}$ and all decisions α for $\mathcal{T}$ are in $\mathcal{U}$, all literals from $\mathcal{T}[i, j-1]$ are contained in $\mathcal{U}$. This means that at some point in $\mathcal{U}$ the clause $\text{ante}_{\mathcal{T}}(p_{(i,j)})$ could have been used to imply $p_{(i,j)}$. Because we are not allowed to skip propagations, we must have propagated $p_{(i,j)}$ somewhere in $\mathcal{U}$, contradicting our assumption. □

In the following proposition we will give a simple counting argument. This gives another motivation for using asserting schemes. Informally, it says that we only have to backtrack polynomially often under a specific decision sequence until we reach a desired state of unreliability.

Proposition 6.5. *Let $\Phi := \mathcal{Q} \cdot \phi$ be a QCNF in n variables, ξ be an asserting learning scheme, D be a clause and let $\mathcal{T}$ be a natural $\mathsf{QCDCL}^{\mathit{Ass\text{-}Ord}}_{\mathit{No\text{-}Red}}$ trail for Φ with decision set $\bar{D}$ and $\square \in \mathcal{T}$. Then there exists a clause E and a $\mathsf{QCDCL}^{\mathit{Ass\text{-}Ord}}_{\mathit{No\text{-}Red}}$-proof*

$$\iota = ((\mathcal{T}_1, \dots, \mathcal{T}_{f_n}), (\xi(\mathcal{T}_1), \dots, \xi(\mathcal{T}_{f_n})), (\pi_1, \dots, \pi_{f_n}))$$

from Φ of E such that $|\iota| \in \mathcal{O}(n^3)$. If $E \neq (\square)$, then D is unreliable with respect to $\mathcal{Q} \cdot (\phi \cup \{\xi(\mathcal{T}_1), \dots, \xi(\mathcal{T}_{f_n})\})$.

In particular, in case $E \neq (\square)$ we can find a trail $\mathcal{W}$ that witnesses the unreliability of D after having learnt the clauses $\xi(\mathcal{T}_1), \dots, \xi(\mathcal{T}_{f_n})$ in the proof ι, i.e., there is a trail $\mathcal{W}$ for $\mathcal{Q} \cdot (\phi \cup \{\xi(\mathcal{T}_1), \dots, \xi(\mathcal{T}_{f_n})\})$, an existential literal $\ell \in D$ with $\ell \in \mathcal{W}$ and decisions $\alpha \subseteq \bar{D} \backslash \{\bar{\ell}\}$ that follow the same order as the decisions of $\mathcal{T}$.

Proof. Set $\mathcal{T}_1 := \mathcal{T}$. We will now construct a proof ι with a sequence of trails $\mathcal{T}_1, \dots, \mathcal{T}_{f_n}$ for some $f_n \in \mathbb{N}$. Arguing inductively, suppose that $\mathcal{T}_i$ was created with decisions $\bar{D}$ and runs into a conflict. We start clause learning and derive $\xi(\mathcal{T}_i)$, which is an asserting clause. After that we will backtrack to a point at which $\xi(\mathcal{T}_i)$ becomes unit, say $\mathcal{T}_i[s_i, t_i]$, and go on constructing the next trail $\mathcal{T}_{i+1}$. From there on we will complete the trail by choosing the same decision literals in the same order as before (while still considering the situations described in Remark 6.2). Either $\mathcal{T}_{i+1}$ also runs into a conflict, in which case we repeat this whole process, or the decisions $\bar{D}$ block each other in $\mathcal{T}_{i+1}$. If this happens, or if we derive $(\square)$, we stop. Note that we will always follow the same decision order, even if we skip some decisions. This proves the last statement of the proposition.

We will argue that the number of these backtracking steps is polynomially bounded, in fact $f_n \in \mathcal{O}(n^2)$. For a variable $z \in \text{Var}(\Phi)$ and a trail $\mathcal{U}$ we define $\nu_{\mathcal{U}}(z)$ as the level in which z was propagated or decided in $\mathcal{U}$, regardless of the polarity of z. If z does not occur in $\mathcal{U}$ in either polarity, then we set $\nu_{\mathcal{U}}(z) := \infty$. Let δ be the map defined as follows:

$$\begin{aligned} \delta : [f_n] \times \text{var}(\phi) &\longrightarrow \{0, \dots, n, \infty\} \\ \delta(1, z) &:= \nu_{\mathcal{T}_1}(z), \\ \delta(i, z) &:= \min(\delta(i-1, z), \nu_{\mathcal{T}_i}(z)) \quad \text{ for } i \in \{2, \dots, f_n\}. \end{aligned}$$

Intuitively, $\delta(i, z)$ returns the smallest decision level in which the variable z occurred in the trails $\{\mathcal{T}_1, \dots, \mathcal{T}_i\}$ in any polarity. By construction, we get $\delta(i+1, z) \leq \delta(i, z)$ for all $i \in [f_n - 1]$ and $z \in \text{var}(\phi)$. Furthermore, because ξ is asserting, we can find $y_i \in \text{var}(\phi)$ (e.g. the asserting literal in step i) with $\delta(i+1, y_i) < \delta(i, y_i)$ for each $i \in [f_n - 1]$. This follows from the fact that by definition we have to backtrack at least one level. We have to finish after at most $\mathcal{O}(n^2)$ steps since after that δ would return 0 for each variable. Therefore $f_n \in \mathcal{O}(n^2)$ and $|\iota| \in \mathcal{O}(n^3)$. □

The proof of this proposition not only confirms the existence of such a proof ι, it also gives us an algorithm for creating it (although that is not essential for us here). The connection between $\mathcal{T}$ and ι might not seem obvious at first sight, as $\mathcal{T}$ only serves as a witness in order to start the process of constructing ι.

Let us outline the rest of this section: we aim to construct a $\mathsf{QCDCL}^{\mathsf{Ass\text{-}Ord}}_{\mathsf{No\text{-}Red}}$ refutation from a Q-resolution refutation. For this we will go through all clauses in the Q-resolution refutation and make them unreliable using Proposition 6.5. For this purpose we have to repeatedly find a natural trail $\mathcal{T}$ with decision set $\bar{D}$ that fulfils the postulated properties. There are three ways a clause could have been derived: as an axiom, via resolution or via reduction. Therefore the next three lemmas will concentrate on this goal.

In the following results we will assume that all clauses, which we intend to make unreliable, are in fact reliable at the beginning. This guarantees that all construction steps can be correctly carried out. However, since we want to p-simulate Q-resolution (which includes computing the simulation in polynomial time), we have to argue that we can efficiently find witnesses of unreliability. For example, axioms will typically be unreliable (purely universal clauses are not, for instance), but we might not know a witness in advance (although this witness exists by definition). What we can do is to act as if the axiom is reliable and try to perform the construction steps as described below. Dropping the assumption of reliabilty, we would lose the guarantee that these construction steps work correctly. However, if not we will obtain a witness of unreliability just by constructing this trail, which similarly serves our purpose. We will discuss this in greater detail when proving Theorem 6.9.

We start with considering the axiom case.

Lemma 6.6. *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 $\mathsf{QCDCL}^{\mathsf{Ass\text{-}Ord}}_{\mathsf{No\text{-}Red}}$-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 $E \neq (\Box)$, then C is unreliable with respect to $\mathcal{Q}\cdot(\phi \cup \{\xi(\mathcal{T}_1), \dots, \xi(\mathcal{T}_{f_n})\})$.*

Proof. Construct a (natural) trail $\mathcal{T}$ by choosing $\bar{C}$ as level-ordered decisions. If we do not run into a conflict, we get a contradiction since we falsified C and are not allowed to skip conflicts. After applying Proposition 6.5 we have either derived $(\Box)$, or C becomes unreliable. □

The next two results cover the simulation of resolution and reduction steps. As before, we assume that the clause for which we intend to find a witness of unreliability is actually reliable at the beginning. Dropping this assumption removes the guarantee that the decisions will not block each other in the construction. But then we might find a witness of unreliability even earlier.

Lemma 6.7. *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 $\mathsf{QCDCL}^{\mathsf{Ass\text{-}Ord}}_{\mathsf{No\text{-}Red}}$-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 $E \neq (\Box)$, then $C_1 \vee C_2$ is unreliable with respect to $\mathcal{Q}\cdot(\phi \cup \{\xi(\mathcal{T}_1), \dots, \xi(\mathcal{T}_{f_n})\})$.*

Proof. Since $C_1 \vee x$ is unreliable, there is a literal $\ell_1 \in C_1 \vee x$ and a trail $\mathcal{U}_1$ for Ψ with decisions $\alpha_1 \subseteq \overline{(C_1 \vee x)}\backslash\{\bar{\ell}_1\}$ such that $\ell_1 \in \mathcal{U}_1$. The same is true for $C_2 \vee \bar{x}$, thus we get a trail $\mathcal{U}_2$ for Υ, a literal ℓ_2 and decisions $\alpha_2 \subseteq \overline{(C_2 \vee \bar{x})}\backslash\{\bar{\ell}_2\}$.

We now distinguish three cases. In each case we will define a set of decisions that are not blocking each other, construct a natural trail with these decisions (see Remark 6.2) and run into a conflict. After that we will unite all three cases and start clause learning. The created trail will serve as a starting point for Proposition 6.5.

Case 1: $\ell_1 = x$ and $\ell_2 = \bar{x}$.

We choose the set $\alpha_1 \cup \alpha_2 \subseteq \bar{C}$ as level-ordered decisions for a new, natural trail $\mathcal{T}$. These decisions cannot block each other since C is still reliable. If we assume that $\mathcal{T}$ does not run into a conflict, then we get $\alpha_1 \cup \alpha_2 \subseteq \mathcal{T}$. We can now apply Lemma 6.4 and conclude that all propagated literals from $\mathcal{U}_1$ and $\mathcal{U}_2$ are contained in $\mathcal{T}$. However, this is a contradiction since we would need to propagate both $\ell_1 = x$ and $\ell_2 = \bar{x}$. Therefore $\mathcal{T}$ has to run into a conflict.

Case 2: $\ell_1 = x$ and $\ell_2 \neq \bar{x}$ (or analogously $\ell_1 \neq x$ and $\ell_2 = \bar{x}$).

We choose the set $(\{\bar{\ell}_2\} \cup \alpha_1 \cup \alpha_2)\backslash\{x\} \subseteq \bar{C}$ as level-ordered decisions for a natural trail $\mathcal{T}$. As before, these decisions are not blocking each other and we can at least do the same propagations as in $\mathcal{U}_1$ as long as we do not run into a conflict because of $\alpha_1 \subseteq \mathcal{T}$. In particular we are able to propagate $x = \ell_1$ (e.g. via $\text{ante}_{\mathcal{U}_1}(\ell_1)$). Now we have decided or propagated all decisions of $\mathcal{U}_2$, i.e., $\alpha_2 \subseteq \mathcal{T}$. Hence after applying Lemma 6.4 again we can do at least all propagations of $\mathcal{U}_2$. But then we would need to propagate ℓ_2 in $\mathcal{T}$. That is contradictory to $\bar{\ell}_2 \in \mathcal{T}$. This means likewise $\mathcal{T}$ has to run into a conflict.

Case 3: $\ell_1 \neq x$ and $\ell_2 \neq \bar{x}$.

We know that $\bar{x} \in \alpha_1$, otherwise C would be unreliable. We choose $\{\bar{\ell}_1\} \cup \alpha_1$ as decisions for the natural trail $\mathcal{T}$, but we demand that $\bar{x}$ will be decided last and all the other decisions are level-ordered.

The decisions before $\bar{x}$ cannot block each other, since its negations are literals in C.

If we propagated x somewhere, we can instead start with a new natural trail $\mathcal{T}'$ using the non-blocking level-ordered decisions $(\{\bar{\ell}_1, \bar{\ell}_2\} \cup \alpha_1 \cup \alpha_2)\backslash\{x, \bar{x}\} \subseteq \bar{C}$. Provided that $\mathcal{T}'$ does not run into a conflict, applying Lemma 6.4 gives us $x \in \mathcal{T}'$ since we already implied x in $\mathcal{T}$. Therefore we have decided or propagated all decisions of $\mathcal{U}_2$, i.e., $\alpha_2 \subseteq \mathcal{T}'$. Hence we get a contradiction by Lemma 6.4 because we would need to propagate ℓ_2. Therefore $\mathcal{T}'$ runs into a conflict.

If we actually decide or propagate $\bar{x}$, we will run into a conflict afterwards. Otherwise we would have made all propagations from $\mathcal{U}_1$ (since $\alpha_1 \subseteq \mathcal{T}$), receiving a contradiction by Lemma 6.4 again.

Using Proposition 6.5, for Case 1, Case 2 and the first part of Case 3 (where we propagated x before we could decide $\bar{x}$) we can construct a $\mathsf{QCDCL}^{\mathsf{Ass\text{-}Ord}}_{\mathsf{No\text{-}Red}}$-proof ι with $|\iota| \in \mathcal{O}(n^3)$ that fulfils the desired properties. Note that in each case we have followed the policy $\mathsf{Ass\text{-}Ord}$.

The last part of Case 3 (where we actually decide or propagate $\bar{x}$) works slightly different. Here our decisions were $\{\bar{\ell}_1\} \cup \alpha_1 \subseteq \overline{(C_1 \vee x)}$, therefore we also apply Proposition 6.5 and construct a $\mathsf{QCDCL}^{\mathsf{Ass\text{-}Ord}}_{\mathsf{No\text{-}Red}}$-proof ι' with $\theta(\iota') = \mathcal{T}'_1, \ldots, \mathcal{T}'_{f'_n}$ from Φ of a clause E' that uses ξ such that $|\iota'| \in \mathcal{O}(n^3)$. If we have $E' = (\Box)$, we are done. If not, then $C_1 \vee x$ became unreliable with respect to $\mathcal{Q} \cdot (\phi \cup \{\xi(\mathcal{T}'_1), \ldots, \xi(\mathcal{T}'_{f'_n})\})$. Here we need the last statement of Proposition 6.5: There is a trail $\mathcal{W}$ for $\mathcal{Q} \cdot (\phi \cup \{\xi(\mathcal{T}'_1), \ldots, \xi(\mathcal{T}'_{f'_n})\})$, an existential literal $\ell \in C_1 \vee x$ with $\ell \in \mathcal{W}$ and decisions $\alpha \subseteq \overline{(C_1 \vee x)}\backslash\{\bar{\ell}\}$ that follow the same order as the decisions of $\mathcal{T}$. Since in this order the literal $\bar{x}$ was always the last literal, we conclude $\bar{x} \notin \alpha$ (otherwise we would not have a chance to achieve a situation where the decisions block each other). There are two remaining possibilities: If $\ell \neq x$, then $\mathcal{W}$ is a witness for the unreliability of C_1 (and therefore also $C_1 \vee C_2$), which gives us the desired result.

However, if $\ell = x$, we can go back to Case 2 and use the trails $\mathcal{W}$ and $\mathcal{U}_2$. We create another $\mathsf{QCDCL}^{\mathsf{Ass\text{-}Ord}}_{\mathsf{No\text{-}Red}}$-proof ι'' with $\theta(\iota'') = \mathcal{T}''_1, \ldots, \mathcal{T}''_{f''_n}$ from $\mathcal{Q} \cdot (\phi \cup \{\xi(\mathcal{T}'_1), \ldots, \xi(\mathcal{T}'_{f'_n})\})$ of a clause E that uses the learning scheme ξ such that $|\iota''| \in \mathcal{O}(n^3)$. We can combine these two proofs ι' and ι'' into a proof ι with $\theta(\iota) = \mathcal{T}'_1, \ldots, \mathcal{T}'_{f'_n}, \mathcal{T}''_1, \ldots, \mathcal{T}''_{f''_n}$ from Φ of the clause E by connecting the components $\theta(\iota')$ with $\theta(\iota'')$, $\lambda(\iota')$ with $\lambda(\iota'')$ and $\rho(\iota')$ with $\rho(\iota'')$ such that $|\iota| \in \mathcal{O}(n^3)$. Between the trails $\mathcal{T}'_{f'_n}$ and $\mathcal{T}''_1$ we backtrack to the time $(0, 0)$ (i.e., we restart the trail). If $E \neq (\Box)$, then $C_1 \vee C_2$ became unreliable with respect to $\mathcal{Q} \cdot (\phi \cup \{\xi(\mathcal{T}'_1), \ldots, \xi(\mathcal{T}'_{f'_n}), \xi(\mathcal{T}''_1), \ldots, \xi(\mathcal{T}''_{f''_n})\})$. $\Box$

It remains to show a similar result for the reduction step.

Lemma 6.8. *Let $\Phi := \mathcal{Q} \cdot \phi$ be a QCNF in n variables, let $D := C \vee u_1 \vee \ldots \vee u_m$ be a non-tautological clause with universal literals $u_1, \ldots, u_m$ and 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 $\mathsf{QCDCL}^{\mathsf{Ass\text{-}Ord}}_{\mathsf{No\text{-}Red}}$-proof ι with $\theta(\iota) = \mathcal{T}_1, \ldots, \mathcal{T}_{f_n}$ from Φ*

of some clause E that uses the learning scheme ξ such that $|\iota| \in \mathcal{O}(n^3)$. If $E \neq (\Box)$, then C is unreliable with respect to $\mathcal{Q} \cdot (\phi \cup \{\xi(\mathcal{T}_1), \ldots, \xi(\mathcal{T}_{f_n})\})$.

Proof. Because D is unreliable, we can find a literal $\ell \in D$ and a trail $\mathcal{U}$ for Ψ with decisions $\alpha \subseteq \bar{D} \backslash \{\bar{\ell}\}$ and $\ell \in \mathcal{U}$. The literal ℓ has to be existential, hence $\ell \in C$.

Now we choose the set $\{\bar{\ell}\} \cup \alpha \subseteq \bar{D}$ as level-ordered decisions for the natural trail $\mathcal{T}$, i.e., we decide the literals in $(\{\bar{\ell}\} \cup \alpha) \cap \{\bar{u}_1, \ldots, \bar{u}_m\}$ at the end. The decisions before $(\{\bar{\ell}\} \cup \alpha) \cap \{\bar{u}_1, \ldots, \bar{u}_m\}$ cannot block each other since C is reliable. After this we would only decide universal literals, which can not be propagated. Therefore, in this order, the decisions $\{\bar{\ell}\} \cup \alpha$ are not blocking each other. If $\mathcal{T}$ does not run into a conflict, we can at least do the same propagations as in $\mathcal{U}$ by Lemma 6.4. Then we would get the contradiction $\ell, \bar{\ell} \in \mathcal{T}$.

Now we can apply Proposition 6.5 and construct a proof ι with $\theta(\iota) = \mathcal{T}_1, \ldots, \mathcal{T}_{f_n}$ from Φ of a clause E such that $|\iota| \in \mathcal{O}(n^3)$. If $E \neq (\Box)$, then D is unreliable with respect to $\mathcal{Q} \cdot (\phi \cup \{\xi(\mathcal{T}_1), \ldots, \xi(\mathcal{T}_{f_n})\})$. In this case there is a trail $\mathcal{W}$ for $\mathcal{Q} \cdot (\phi \cup \{\xi(\mathcal{T}_1), \ldots, \xi(\mathcal{T}_{f_n})\})$, an existential literal $y \in D$ with $y \in \mathcal{W}$ and decisions $\beta \subseteq \bar{D} \backslash \{\bar{y}\}$ that follows the same order as the decisions of $\mathcal{T}$ (level-order). Since y is existential, we have $y \in C$. The blocking situation in $\mathcal{W}$ had to occur before we could decide $\bar{u}_1, \ldots, \bar{u}_m$. We conclude $\beta \subseteq \bar{C} \backslash \{\bar{y}\}$ and thus C is unreliable with respect to $\mathcal{Q} \cdot (\phi \cup \{\xi(\mathcal{T}_1), \ldots, \xi(\mathcal{T}_{f_n})\})$. □

Now we can combine these three auxiliary results into the main theorem of this section.

Theorem 6.9. $\mathsf{QCDCL}^{\textsf{Ass-Ord}}_{\textsf{No-Red}}$ *p-simulates* $\mathsf{Q\text{-}resolution}$*. I.e., each* $\mathsf{Q\text{-}resolution}$ *refutation π of a QCNF in n variables can be transformed into a* $\mathsf{QCDCL}^{\textsf{Ass-Ord}}_{\textsf{No-Red}}$*-refutation of size $\mathcal{O}(n^3 \cdot |\pi|)$ that uses an arbitrary asserting learning scheme ξ.*

In particular, $\mathsf{Q\text{-}resolution}$*,* $\mathsf{QCDCL}^{\textsf{Any-Ord}}_{\textsf{No-Red}}$ *and* $\mathsf{QCDCL}^{\textsf{Ass-Ord}}_{\textsf{No-Red}}$ *are p-equivalent proof systems.*

Proof. First we show that $\mathsf{QCDCL}^{\textsf{Ass-Ord}}_{\textsf{No-Red}}$ simulates $\mathsf{Q\text{-}resolution}$. Let $\Phi := \mathcal{Q} \cdot \phi$ be a QCNF in n variables with a $\mathsf{Q\text{-}resolution}$ refutation $\pi = C_1, \ldots, C_k$. We will go through this proof from left to right and check whether or not the clauses are reliable with respect to the corresponding current QCNF. Suppose that $C_1, \ldots, C_{i-1}$ are already unreliable with respect to QCNFs $\mathcal{Q} \cdot \phi_1, \ldots, \mathcal{Q} \cdot \phi_{i-1}$ with $\phi \subseteq \phi_1 \subseteq \ldots \subseteq \phi_{i-1}$. If C_i is unreliable with respect to $\mathcal{Q} \cdot \phi_{i-1}$, then we can set $\phi_i := \phi_{i-1}$ (let $\phi_0 := \phi$), $\iota_i := (\emptyset, \emptyset, \emptyset)$ (empty proof) and continue with C_{i+1}.

However, if C_i is reliable with respect to $\mathcal{Q} \cdot \phi_{i-1}$, then we can apply either Lemma 6.6 or Lemma 6.7 or Lemma 6.8, depending whether C_i is an axiom, or was derived via resolution or reduction. We construct a $\mathsf{QCDCL}^{\textsf{Ass-Ord}}_{\textsf{No-Red}}$-proof ι_i with $\theta(\iota_i) = \mathcal{T}_1^{(i)}, \ldots, \mathcal{T}^{(i)}_{f_n^{(i)}}$ of a clause E_i from $\mathcal{Q} \cdot (\phi_{i-1} \cup \{\xi(\mathcal{T}_1^{(i)}), \ldots, \xi(\mathcal{T}^{(i)}_{f_n^{(i)}})\})$ with $|\iota_i| \in \mathcal{O}(n^3)$ that uses the learning scheme ξ. If $E_i = (\Box)$, we are done and connect the components of the proofs $\iota_1, \ldots, \iota_i$, in particular $\theta(\iota) = \theta(\iota_1), \ldots, \theta(\iota_i)$. Otherwise C_i became unreliable with respect to $\mathcal{Q} \cdot (\phi_{i-1} \cup \{\xi(\mathcal{T}_1^{(i)}), \ldots, \xi(\mathcal{T}^{(i)}_{f_n^{(i)}})\})$. In this case we set $\phi_i := \phi_{i-1} \cup \{\xi(\mathcal{T}_1^{(i)}), \ldots, \xi(\mathcal{T}^{(i)}_{f_n^{(i)}})\}$ and continue with the next clause C_{i+1}.

At the latest when we reach $C_k = (\Box)$ in π, we will create a refutation ι since $(\Box)$ can never become unreliable by definition. The $\mathsf{QCDCL}^{\textsf{Ass-Ord}}_{\textsf{No-Red}}$-refutation ι consists of some $\mathsf{QCDCL}^{\textsf{Ass-Ord}}_{\textsf{No-Red}}$-proofs $\iota_1, \ldots, \iota_j$, $1 \leq j \leq k$. Between two proofs ι_a and ι_{a+1} we will just restart, i.e., between the trails $\mathcal{T}^{(a)}_{f_n^{(a)}}$ and $\mathcal{T}_1^{(a+1)}$ we will backtrack to the point $(0,0)$. At the end we have $|\iota| \in \mathcal{O}(n^3 \cdot |\pi|)$.

We have now shown $\mathsf{Q\text{-}resolution} \leq \mathsf{QCDCL}^{\textsf{Ass-Ord}}_{\textsf{No-Red}}$. In order to actually prove that this simulation is polynomial-time computable, one should actually argue that the described construction steps are in fact polynomial-time computable. However, we needed to decide whether or not a clause in the given proof is reliable. This might not be computable in polynomial time. Alternatively, we can pretend that a clause C is reliable unless we have found a witness that proves the opposite. We can do the exact same steps as described in the above results. In detail, we can still try to create the trail $\mathcal{T}$ from Lemma 6.6, Lemma 6.7 or Lemma 6.8. If the decisions do not block each other, we proceed as if the clause C was reliable. Otherwise, we immediately receive a witness for our unreliability, even if we were not able to take the steps as described.

Therefore $\mathsf{QCDCL}^{\mathsf{Ass\text{-}Ord}}_{\mathsf{No\text{-}Red}}$ p-simulates Q-resolution. By Theorem 3.8 the systems $\mathsf{QCDCL}^{\mathsf{Any\text{-}Ord}}_{\mathsf{No\text{-}Red}}$ and $\mathsf{QCDCL}^{\mathsf{Ass\text{-}Ord}}_{\mathsf{No\text{-}Red}}$ are p-simulated by Q-resolution. Obviously, $\mathsf{QCDCL}^{\mathsf{Ass\text{-}Ord}}_{\mathsf{No\text{-}Red}}$ is p-simulated by $\mathsf{QCDCL}^{\mathsf{Any\text{-}Ord}}_{\mathsf{No\text{-}Red}}$. As a consequence, all of these three systems are p-equivalent. □

7 Comparison to the correspondence between propositional resolution and CDCL

There are a few similarities between our definitions of *unreliable/reliable* and the notions *1-empowering/absorbed* used in [42]. Pipatsrisawat and Darwiche [42] focused on propositional logic and CNFs, so let us for this section restrict our attention to the propositional case (e.g. by considering only QCNFs with existential quantifiers) and recall the definition that was used in the work of Pipatsrisawat and Darwiche.

Definition 7.1 (Pipatsrisawat & Darwiche [42])**.** *Let $\alpha \Rightarrow \ell$ be a clause where ℓ is some literal and α is a conjunction of literals. The clause is* 1-empowering *with respect to CNF Δ iff*

1. *$\Delta \vDash (\alpha \Rightarrow \ell)$: the clause is implied by Δ.*
2. *$\Delta \wedge \alpha$ is 1-consistent: asserting α does not result in a conflict that is detectable by unit resolution.*
3. *$\Delta \wedge \alpha \nvdash_1 \ell$: the literal ℓ cannot be derived from $\Delta \wedge \alpha$ using unit resolution.*

In this case, ℓ is called an empowering literal *of a clause. On the other hand, a clause implied by Δ that contains no empowering literals is said to be* absorbed *by Δ.*

For further details concerning the notations see [42].

Translating the third point into our framework could lead to the following interpretation: ℓ cannot be propagated in a trail for Δ with decisions α. This is related to our definition of reliability. Simply put, for reliability we require that no literal in the clause is derivable by unit propagation. In Definition 7.1, however, it suffices to find at least one literal ℓ with this property.

Consequently the differences between 'unreliable' and 'absorbing' could be formulated as follows: for a clause C to be unreliable, we need at least one literal $\ell \in C$ such that ℓ is 'accidentally' propagated in a trail with decisions contained in $\bar{C} \setminus \{\bar{\ell}\}$. In an absorbed clause, on the other hand, each literal has to be propagated accidentally in this way.

In [42] this difference in definition caused an additional factor n in the complexity of the CDCL simulation of resolution. Roughly speaking, their idea consists of searching for 1-empowering (and 1-provable) clauses in a given resolution refutation π of a CNF Δ. They described how these clauses get absorbed after $\mathcal{O}(n^4)$ CDCL steps, where the last n-factor is incurred by the fact that they have to handle all empowering literals, not just one as in our results. The remaining factor n^3 can be explained in a similar way as in Proposition 6.5 (cf. [42, Prop. 3]).

Consequently, we obtain a slight quantitative improvement of the simulation of resolution by CDCL [42] from $\mathcal{O}(n^4|\pi|)$ to $\mathcal{O}(n^3|\pi|)$.

Theorem 7.2. *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 $\mathcal{O}(n^3|\pi|)$.*

An advantage of the '1-empowering/absorbed' notion is the simplification when it comes to cover the resolution steps in the given proof π. In the proof of Lemma 6.7 we had to distinguish three cases depending on the literal that witnessed the unreliability. However, this is not necessary when using the definition of 'absorption'. Since in this case all literals from a clause shall be propagated accidentally, we can pick an arbitrary literal that simplifies the following reasoning steps. In fact, it then suffices to consider Case 1 in Lemma 6.7.

Furthermore, in [42] the authors refrained from using the concept of trails or introducing algorithm-based proof systems. Instead, they relied on the notions of unit resolution, 1-consistency, and 1-provability. However, these notions cannot be fully translated into our framework that enables the construction of trails as it is done in practical CDCL. For example, consider the following:

Let $\Delta := (\bar{x}_1 \vee \bar{x}_2 \vee \bar{x}_3)$ be a CNF consisting only of one clause. Then clearly this CNF together with decisions x_1, x_2, x_3 is 1-inconsistent (one could think of 1-inconsistent CDCL states, that are in the form of $\Delta \wedge \ell_1 \wedge \ldots \wedge \ell_m$ with a CNF Δ and decision literals $\ell_1, \ldots, \ell_m$, as formulas which are refutable via unit propagation). However, this conflict cannot occur in practice when creating trails as described in our work (i.e., in a natural sense). We would obtain a situation where the decisions block each other:

$$\mathcal{T} = (\mathbf{x_1}; \mathbf{x_2}, \bar{x}_3) \ .$$

This blocking can only be resolved by skipping the propagation $\bar{x}_3$. Of course, we can observe this artificial conflict by our notion of trails as well:

$$\mathcal{T}' = (\mathbf{x_1}; \mathbf{x_2}; \mathbf{x_3}, \square) \ .$$

But since we want to avoid this inherent kind of non-determinism (cf. also [4]), we defined our proof systems in such a way as to circumvent these situations. As a consequence, it is impossible for the trail $\mathcal{T}'$ to appear in a proof under one of our QCDCL systems. This distinction between non-deterministic, more liberal systems and algorithm-based, stricter versions of these systems seems in some way harder to clarify with the notions used by Pipatsrisawat and Darwiche.

8 The simulation order of QCDCL proof systems

Now that we characterised the complexity of classical systems, such as QCDCL and Q-resolution, we want to examine the connections between the remaining QCDCL systems that we have not fully considered yet. We refer again to Figure 2 on page 10, depicting the resulting simulation order.

First we define the formulas $\texttt{Lon}_n$ that were introduced by Lonsing in [34]. Originally, these QCNFs were constructed to separate QBF solvers that differ in the implemented dependency schemes (we will not consider these concepts here, though).

Definition 8.1 (Lonsing [34]). *Let* $\texttt{Lon}_n$ *be the QCNF*

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

It was shown in [34] that this formula becomes easy to refute by choosing the standard dependency scheme. However, $\texttt{Lon}_n$ serves as a witness for separating our systems as well.

Proposition 8.2. *The QCNFs* $\texttt{Lon}_n$ *require exponential-size proofs in the systems* $\mathsf{QCDCL}^{\mathit{LEV\text{-}ORD}}_{\mathit{RED}}$ *and* $\mathsf{QCDCL}^{\mathit{LEV\text{-}ORD}}_{\mathit{NO\text{-}RED}}$*, but have constant-size proofs in* $\mathsf{QCDCL}^{\mathit{ASS\text{-}R\text{-}ORD}}_{\mathit{RED}}$ *and* Q-resolution*.*

Proof. With the policy LEV-ORD we are forced to start assigning the variables $a, b, b_1, \ldots, b_{s_n}$. As long as we do this, we can only use the clauses from $(a \vee b \vee b_1 \vee \ldots \vee b_{s_n}) \wedge \texttt{PHP}_n^{n+1}(b_1, \ldots, b_{s_n})$ as antecedent clauses. Since $\texttt{PHP}_n^{n+1}$ contains subclauses of $b_1 \vee \ldots \vee b_{s_n}$, we do not need the clause

$a \vee b \vee b_1 \vee \ldots \vee b_{s_n}$ either. The propositional formula $\mathtt{PHP}_n^{n+1}$ is unsatisfiable, therefore we will falsify this formula before reaching the decisions x and y. We will always learn clauses C whose long-distance Q-resolution proofs consist only of axioms from $\mathtt{PHP}_n^{n+1}$. These proofs are in fact resolution proofs because the contained clauses do not include any universal variables. At the end we obtain a QCDCL-proof ι (in the system $\mathsf{QCDCL}_{\mathsf{RED}}^{\mathsf{LEV\text{-}ORD}}$ or $\mathsf{QCDCL}_{\mathsf{NO\text{-}RED}}^{\mathsf{LEV\text{-}ORD}}$) that contains a resolution refutation of $\mathtt{PHP}_n^{n+1}$, which is exponential by [26]. Hence also $|\iota|$ is exponential.

We obtain short Q-resolution refutations of $\mathtt{Lon}_n$ by $\text{red}(\text{res}(x \vee c, x \vee \bar{c}, c)) = (\Box)$. The following trail yields a short $\mathsf{QCDCL}_{\mathsf{RED}}^{\mathsf{ASS\text{-}R\text{-}ORD}}$ refutation of $\mathtt{Lon}_n$:

$$\mathcal{T} = (\bar{\mathbf{x}}, c, \Box).$$

From this we can learn $(\Box)$ in the same way as in the Q-resolution refutation. □

Next we want to compare the policies RED and NO-RED when fixing the decision policy LEV-ORD. As we indicated before, these two policies seem to operate orthogonally to each other. We will prove this intuition now, again using the $\mathtt{Trapdoor}_n$ formulas.

Proposition 8.3. *The QCNFs $\mathtt{Trapdoor}_n$ have polynomial-size $\mathsf{QCDCL}_{\mathsf{NO\text{-}RED}}^{\mathsf{LEV\text{-}ORD}}$ refutations.*

Proof. The refutation consists of the trails $\mathcal{T}_1, \mathcal{T}_2$:

$$\mathcal{T}_1 := (\mathbf{y_1}; \mathbf{y_2}; \ldots; \mathbf{y_{s_n}}; \bar{\mathbf{w}}, t, \Box)$$

with $\text{ante}_{\mathcal{T}_1}(t) = \bar{y}_1 \vee w \vee t$ and $\text{ante}_{\mathcal{T}_1}(\Box) = \bar{y}_1 \vee w \vee \bar{t}$. We learn the clause $(\bar{y}_1)$ and backtrack to $(0, 0)$.

$$\mathcal{T}_2 := (\bar{y}_1; \bar{\mathbf{y}}_\mathbf{2}; \ldots; \bar{\mathbf{y}}_{\mathbf{s_n}}; \bar{\mathbf{w}}, t, \Box)$$

with $\text{ante}_{\mathcal{T}_2}(t) = y_1 \vee w \vee t$ and $\text{ante}_{\mathcal{T}_2}(\Box) = y_1 \vee w \vee \bar{t}$. We finally learn the clause $(\Box)$. □

Combined with previous results, we can conclude the following:

Theorem 8.4. *The systems $\mathsf{QCDCL}_{\mathsf{NO\text{-}RED}}^{\mathsf{LEV\text{-}ORD}}$ and $\mathsf{QCDCL}_{\mathsf{RED}}^{\mathsf{LEV\text{-}ORD}}$ are incomparable.*

Proof. The QCNFs $\mathtt{QParity}_n$ are hard for $\mathsf{QCDCL}_{\mathsf{NO\text{-}RED}}^{\mathsf{LEV\text{-}ORD}}$ by [6] since this system is p-simulated by Q-resolution, but easy for $\mathsf{QCDCL}_{\mathsf{RED}}^{\mathsf{LEV\text{-}ORD}}$ as proven in Proposition 4.2. The formulas $\mathtt{Trapdoor}_n$ are hard for $\mathsf{QCDCL}_{\mathsf{RED}}^{\mathsf{LEV\text{-}ORD}}$ (Proposition 4.6), but easy for $\mathsf{QCDCL}_{\mathsf{NO\text{-}RED}}^{\mathsf{LEV\text{-}ORD}}$ (Proposition 8.3). □

In Section 5 we already introduced a whole class of QCNFs that require large $\mathsf{QCDCL}_{\mathsf{RED}}^{\mathsf{LEV\text{-}ORD}}$ (=QCDCL) refutations. We can exponentially improve this classical QCDCL system by exchanging the decision policy LEV-ORD for a more liberal one. Although we have already shown this in Proposition 8.2, we will give another, more interesting separation by QBFs whose hardness does not rely on propositional resolution complexity.

Proposition 8.5. *The formulas $\mathtt{Equality}_n$ have polynomial-size $\mathsf{QCDCL}_{\mathsf{RED}}^{\mathsf{ASS\text{-}R\text{-}ORD}}$ refutations.*

Proof. First we define the clauses

$$L_i := \bar{x}_i \vee \bar{u}_i \vee \bigvee_{j=i+1}^{n} (u_j \vee \bar{u}_j) \vee \bigvee_{k=1}^{i-1} \bar{t}_k,$$

$$R_i := x_i \vee u_i \vee \bigvee_{j=i+1}^{n} (u_j \vee \bar{u}_j) \vee \bigvee_{k=1}^{i-1} \bar{t}_k$$

for $i = 2, \ldots, n$.

We will construct $\mathsf{QCDCL}_{\mathsf{RED}}^{\mathsf{ASS\text{-}R\text{-}ORD}}$ trails $\mathcal{T}_n, \mathcal{U}_n, \ldots, \mathcal{T}_2, \mathcal{U}_2$ from which we learn the clauses $L_n, R_n, \ldots, L_2, R_2$. We will restart after each trail.

The initial trail is

$$\mathcal{T}_n = (\mathbf{x_1}; \mathbf{x_2}; \ldots; \mathbf{x_n}; \mathbf{u_1}, t_1; \mathbf{u_2}, t_2; \ldots; \mathbf{u_{n-1}}, t_{n-1}, \bar{t}_n, \square)$$

coupled with the antecedent clauses

$$\begin{aligned} \text{ante}_{\mathcal{T}_n}(t_j) &= \bar{x}_j \vee \bar{u}_j \vee t_j \quad \text{ for } j = 1, \ldots, n-1, \\ \text{ante}_{\mathcal{T}_n}(\bar{t}_n) &= \bar{t}_1 \vee \ldots \vee \bar{t}_n, \\ \text{ante}_{\mathcal{T}_n}(\square) &= \bar{x}_n \vee \bar{u}_n \vee t_n. \end{aligned}$$

After resolving over $\bar{t}_n$ we learn L_n.

We restart and can create $\mathcal{U}_n$, symmetrically to $\mathcal{T}_n$:

$$\mathcal{U}_n = (\mathbf{\bar{x}_1}; \mathbf{\bar{x}_2}; \ldots; \mathbf{\bar{x}_n}; \mathbf{\bar{u}_1}, t_1; \mathbf{\bar{u}_2}, t_2; \ldots; \mathbf{\bar{u}_{n-1}}, t_{n-1}, \bar{t}_n, \square)$$

with

$$\begin{aligned} \text{ante}_{\mathcal{U}_n}(t_j) &= x_j \vee u_j \vee t_j \quad \text{ for } j = 1, \ldots, n-1, \\ \text{ante}_{\mathcal{U}_n}(\bar{t}_n) &= \bar{t}_1 \vee \ldots \vee \bar{t}_n, \\ \text{ante}_{\mathcal{U}_n}(\square) &= x_n \vee u_n \vee t_n. \end{aligned}$$

Analogously, we learn R_n.

Now suppose that we already learned the clauses $L_n, \ldots, L_i$ and $R_n, \ldots, R_i$ for $3 \leq i \leq n$. Next, let us learn the clause L_{i-1} using the trail $\mathcal{T}_{i-1}$:

$$\mathcal{T}_{i-1} = (\mathbf{x_1}; \mathbf{x_2}; \ldots; \mathbf{x_{i-1}}; \mathbf{u_1}, t_1; \mathbf{u_2}, t_2; \ldots; \mathbf{u_{i-1}}, t_{i-1}, \bar{x}_i, \square)$$

with

$$\begin{aligned} \text{ante}_{\mathcal{T}_{i-1}}(t_j) &= \bar{x}_j \vee \bar{u}_j \vee t_j \quad \text{ for } j = 1, \ldots, i-1, \\ \text{ante}_{\mathcal{T}_{i-1}}(\bar{x}_i) &= L_i, \\ \text{ante}_{\mathcal{T}_{i-1}}(\square) &= R_i. \end{aligned}$$

We resolve over $\bar{x}_i$ and t_{i-1}, obtaining L_{i-1}.

Again, in a symmetrical way we derive R_{i-1}:

$$\mathcal{U}_{i-1} = (\mathbf{\bar{x}_1}; \mathbf{\bar{x}_2}; \ldots; \mathbf{\bar{x}_{i-1}}; \mathbf{\bar{u}_1}, t_1; \mathbf{\bar{u}_2}, t_2; \ldots; \mathbf{\bar{u}_{i-1}}, t_{i-1}, \bar{x}_i, \square)$$

with

$$\begin{aligned} \text{ante}_{\mathcal{U}_{i-1}}(t_j) &= x_j \vee u_j \vee t_j \quad \text{ for } j = 1, \ldots, i-1, \\ \text{ante}_{\mathcal{U}_{i-1}}(\bar{x}_i) &= L_i, \\ \text{ante}_{\mathcal{U}_{i-1}}(\square) &= R_i. \end{aligned}$$

We end the proof with two trails:

$$\mathcal{T}_1 = (\mathbf{x_1}; \mathbf{u_1}, t_1, \bar{x}_1, \square),$$

with similar antecedent clauses as before, from which we learn the unit clause $(\bar{x}_1)$, and

$$\mathcal{U}_1 = (\bar{x}_1; \mathbf{\bar{u}_1}, t_1, \bar{x}_1, \square),$$

from which we finally derive $(\square)$. This whole proof has size $\mathcal{O}(n^2)$. □

Using Ass-R-Ord instead of Lev-Ord allowed us to skip existential decisions. As a result we were able to restrict ourselves to the decisions $x_1, \ldots, x_{i-1}$ in the trails $\mathcal{T}_{i-1}$ and $\mathcal{U}_{i-1}$ since the other variables $x_i, \ldots, x_n$ are either resolved away or useless for the current resolution step.

This leads to the following separation:

Theorem 8.6. $\mathsf{QCDCL}_{\mathsf{Red}}^{\mathsf{Ass\text{-}R\text{-}Ord}}$ *is exponentially stronger than* $\mathsf{QCDCL}_{\mathsf{Red}}^{\mathsf{Lev\text{-}Ord}}$.

Note that the decision policy Ass-R-Ord in $\mathsf{QCDCL}_{\mathsf{Red}}^{\mathsf{Ass\text{-}R\text{-}Ord}}$ guarantees the possibility to learn asserting clauses. Having shown that $\mathsf{QCDCL}_{\mathsf{Red}}^{\mathsf{Ass\text{-}R\text{-}Ord}}$ is actually stronger than classical QCDCL, the system $\mathsf{QCDCL}_{\mathsf{Red}}^{\mathsf{Ass\text{-}R\text{-}Ord}}$ seems to be a promising candidate for practical implementation.

9 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 $\mathsf{QCDCL}_{\mathsf{No\text{-}Red}}^{\mathsf{Ass\text{-}Ord}}$ and $\mathsf{QCDCL}_{\mathsf{Red}}^{\mathsf{Ass\text{-}R\text{-}Ord}}$. 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 $\mathsf{QCDCL}_{\mathsf{Red}}^{\mathsf{Ass\text{-}R\text{-}Ord}}$ 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 8.6) 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 $\mathtt{Trapdoor}_n$ 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 $\mathsf{QCDCL}_{\mathsf{Red}}^{\mathsf{Ass\text{-}R\text{-}Ord}}$ and long-distance Q-resolution?
- Can we even find a separation between $\mathsf{QCDCL}_{\mathsf{Red}}^{\mathsf{Any\text{-}Ord}}$ and long-distance Q-resolution, or are the systems possibly even equivalent?

Acknowledgements

Research was supported by grants from the Carl Zeiss Foundation and the John Templeton Foundation (grant no. 60842).

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, August 2012.

[3] Valeriy Balabanov, Magdalena Widl, and Jie-Hong R. Jiang. QBF resolution systems and their proof complexities. In *SAT'14*, 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.

[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, Joshua Blinkhorn, and Luke Hinde. Size, cost, and capacity: A semantic technique for hard random QBFs. *Logical Methods in Computer Science*, 15(1), 2019.

[7] 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.

[8] 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), accepted*, 2020. Full version at the Electronic Colloquium on Computational Complexity (ECCC), report no. TR20-005.

[9] Olaf Beyersdorff, Ilario Bonacina, Leroy Chew, and Jan Pich. Frege systems for quantified Boolean logic. *J. ACM*, 67(2), 2020.

[10] 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.

[11] Olaf Beyersdorff, Leroy Chew, Meena Mahajan, and Anil Shukla. Feasible interpolation for QBF resolution calculi. *Logical Methods in Computer Science*, 13, 2017.

[12] Olaf Beyersdorff, Leroy Chew, and Karteek Sreenivasaiah. A game characterisation of tree-like Q-Resolution size. *J. Comput. Syst. Sci.*, 104:82–101, 2019.

[13] Olaf Beyersdorff, Luke Hinde, and Ján Pich. Reasons for hardness in QBF proof systems. *ACM Transactions on Computation Theory*, 12(2), 2020.

[14] 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.

[15] Samuel R. Buss. Towards NP-P via proof complexity and search. *Ann. Pure Appl. Logic*, 163(7):906–917, 2012.

[16] 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.

[17] Marco Cadoli, Andrea Giovanardi, and Marco Schaerf. An algorithm to evaluate quantified Boolean formulae. In *Proceedings of the Fifteenth National Conference on Artificial Intelligence and Tenth Innovative Applications of Artificial Intelligence Conference, AAAI 98*, pages 262–267, 1998.

[18] Stephen A. Cook and Robert A. Reckhow. The relative efficiency of propositional proof systems. *The Journal of Symbolic Logic*, 44(1):36–50, 1979.

[19] Martin Davis, George Logemann, and Donald W. Loveland. A machine program for theorem-proving. *Commun. ACM*, 5(7):394–397, 1962.

[20] Martin Davis and Hilary Putnam. A computing procedure for quantification theory. *Journal of the ACM*, 7:210–215, 1960.

[21] Uwe Egly, Florian Lonsing, and Magdalena Widl. Long-distance resolution: Proof generation and strategy extraction in search-based QBF solving. In *Logic for Programming, Artificial Intelligence, and Reasoning - 19th International Conference, LPAR-19*, pages 291–308, 2013.

[22] Peter Faymonville, Bernd Finkbeiner, Markus N. Rabe, and Leander Tentrup. Encodings of bounded synthesis. In *Tools and Algorithms for the Construction and Analysis of Systems - 23rd International Conference, TACAS 2017, Held as Part of the European Joint Conferences on Theory and Practice of Software, ETAPS 2017, Proceedings*, pages 354–370, 2017.

[23] Merrick L. Furst, James B. Saxe, and Michael Sipser. Parity, circuits, and the polynomial-time hierarchy. *Mathematical Systems Theory*, 17(1):13–27, 1984.

[24] 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.

[25] Enrico Giunchiglia, Massimo Narizzano, and Armando Tacchella. Clause/term resolution and learning in the evaluation of quantified Boolean formulas. *J. Artif. Intell. Res.*, 26:371–416, 2006.

[26] Amin Haken. The intractability of resolution. *Theoretical Computer Science*, 39:297–308, 1985.

[27] J. Håstad. *Computational Limitations of Small Depth Circuits.* MIT Press, Cambridge, 1987.

[28] Philipp Hertel, Fahiem Bacchus, Toniann Pitassi, and Allen Van Gelder. Clause learning can effectively p-simulate general propositional resolution. In *AAAI*, 2008.

[29] Mikolás Janota. On Q-resolution and CDCL QBF solving. In Nadia Creignou and Daniel Le Berre, editors, *Theory and Applications of Satisfiability Testing - SAT 2016 - 19th International Conference, 2016, Proceedings*, volume 9710 of *Lecture Notes in Computer Science*, pages 402–418. Springer, 2016.

[30] Mikolás Janota and Joao Marques-Silva. Expansion-based QBF solving versus Q-resolution. *Theor. Comput. Sci.*, 577:25–42, 2015.

[31] Hans Kleine Büning, Marek Karpinski, and Andreas Flögel. Resolution for quantified boolean formulas. *Inf. Comput.*, 117(1):12–18, 1995.

[32] Jan Krajíček. *Proof complexity*, volume 170 of *Encyclopedia of Mathematics and Its Applications*. Cambridge University Press, 2019.

[33] Jan Krajíček and Pavel Pudlák. Propositional proof systems, the consistency of first order theories and the complexity of computations. *The Journal of Symbolic Logic*, 54(3):1063–1079, 1989.

[34] Florian Lonsing. *Dependency Schemes and Search-Based QBF Solving: Theory and Practice*. PhD thesis, Johannes Kepler University Linz, 2012.

[35] Florian Lonsing and Uwe Egly. DepQBF 6.0: A search-based QBF solver beyond traditional QCDCL. In *Automated Deduction - CADE 26 - 26th International Conference on Automated Deduction, Proceedings*, pages 371–384, 2017.

[36] Florian Lonsing, Uwe Egly, and Allen Van Gelder. Efficient clause learning for quantified boolean formulas via QBF pseudo unit propagation. In *Theory and Applications of Satisfiability Testing - SAT 2013 - 16th International Conference*, pages 100–115, 2013.

[37] João P. Marques Silva, Inês Lynce, and Sharad Malik. Conflict-driven clause learning SAT solvers. In *Handbook of Satisfiability*. IOS Press, 2009.

[38] João P. Marques Silva and Karem A. Sakallah. GRASP - a new search algorithm for satisfiability. In *ICCAD*, pages 220–227, 1996.

[39] Nathan Mull, Shuo Pang, and Alexander A. Razborov. On CDCL-based proof systems with the ordered decision strategy. *CoRR*, abs/1909.04135, 2019.

[40] Jakob Nordström. *Short Proofs May Be Spacious : Understanding Space in Resolution*. PhD thesis, Royal Institute of Technology, Stockholm, Sweden, 2008.

[41] Jakob Nordström. On the interplay between proof complexity and SAT solving. *SIGLOG News*, 2(3):19–44, 2015.

[42] Knot Pipatsrisawat and Adnan Darwiche. On the power of clause-learning SAT solvers as resolution engines. *Artif. Intell.*, 175(2):512–525, 2011.

[43] Luca Pulina and Martina Seidl. The 2016 and 2017 QBF solvers evaluations (QBFEVAL'16 and QBFEVAL'17). *Artif. Intell.*, 274:224–248, 2019.

[44] Nathan Segerlind. The complexity of propositional proofs. *Bulletin of Symbolic Logic*, 13(4):417–481, 2007.

[45] Ankit Shukla, Armin Biere, Luca Pulina, and Martina Seidl. A survey on applications of quantified Boolean formulas. In *31st IEEE International Conference on Tools with Artificial Intelligence, ICTAI 2019*, pages 78–84, 2019.

[46] Friedrich Slivovsky and Stefan Szeider. Soundness of Q-resolution with dependency schemes. *TCS*, 612:83–101, 2016.

[47] Moshe Y. Vardi. Boolean satisfiability: theory and engineering. *Commun. ACM*, 57(3):5, 2014.

[48] Marc Vinyals. Hard examples for common variable decision heuristics. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, 2020.

[49] Lintao Zhang, Conor F. Madigan, Matthew W. Moskewicz, and Sharad Malik. Efficient conflict driven learning in Boolean satisfiability solver. In Rolf Ernst, editor, *Proceedings of the 2001 IEEE/ACM International Conference on Computer-Aided Design, ICCAD 2001*, pages 279–285. IEEE Computer Society, 2001.

[50] Lintao Zhang and Sharad Malik. Conflict driven learning in a quantified boolean satisfiability solver. In *IEEE/ACM International Conference on Computer-aided Design, ICCAD 2002*, pages 442–449, 2002.