QCDCL vs QBF Resolution: Further Insights

Benjamin Böhm
Friedrich-Schiller-Universität Jena, Germany

Olaf Beyersdorff
Friedrich-Schiller-Universität Jena, Germany

Abstract

We continue the investigation on the relations of QCDCL and QBF resolution systems. In particular, we introduce QCDCL versions that tightly characterise QU-Resolution and (a slight variant of) long-distance Q-Resolution. We show that most QCDCL variants – parameterised by different policies for decisions, unit propagations and reductions – lead to incomparable systems for almost all choices of these policies.

2012 ACM Subject Classification Theory of computation → Proof complexity

Keywords and phrases QBF, CDCL, resolution, proof complexity, simulations, lower bounds

Digital Object Identifier 10.4230/LIPIcs.SAT.2023.4

Related Version *Extended Version*: `https://eccc.weizmann.ac.il/report/2023/051/`

Funding *Olaf Beyersdorff*: Carl-Zeiss Foundation and DFG grant BE 4209/3-1.

1 Introduction

SAT solving has revolutionised the way we practically handle computationally complex problems [29] and emerged as a central tool for numerous applications [15]. Modern SAT solving crucially relies on the paradigm of conflict-driven clause learning (CDCL) [24], on which almost all current SAT solvers are based.

The main theoretical approach to understanding the success of SAT solving (and its limits) comes through proof complexity [19]. From seminal results [1,5,26] we know that CDCL – viewed as a non-deterministic procedure – is exactly as powerful as propositional resolution, which is by far the best-understood propositional proof system [19,23]. However, we also know that practical CDCL using e.g. VSIDS is exponentially weaker than resolution [30]. Moreover, any deterministic CDCL algorithm will be strictly weaker than resolution unless P=NP [2]. In any case, the mentioned results of [1,5,26] imply that all formulas hard for resolution will be intractable for modern CDCL solvers (at least when disabling preprocessing).

Solving of quantified Boolean formulas (QBF) extends the success of SAT solving to the presumably computationally harder case of deciding QBFs, a PSPACE-complete problem. While QBF solving utilises quite different algorithmic approaches [14], which build on different proof systems, one of the central paradigms again rests on CDCL, lifted to QBFs in form of QCDCL [31]. In comparison to the propositional case, the main changes are (i) different decision strategies using information from the prefix, (ii) differently implemented unit propagation incorporating universal reductions (i.e., dropping trailing universal variables in clauses), and (iii) adapted methods for learning clauses using a QBF resolution system called long-distance Q-Resolution [3].

The advances in QBF solving have also stimulated growing research in QBF proof complexity [6,9,11]. As in the propositional case, QBF resolution systems have received great attention. However, in QBF there are a number of conceptually different resolution systems of varying strength [4,8,12]. The core system is Q-Resolution, introduced in 1995 in [22]. This system generalises propositional resolution to QBF by using the resolution rule

26th International Conference on Theory and Applications of Satisfiability Testing (SAT 2023).
Editors: Meena Mahajan and Friedrich Slivovsky; Article No. 4; pp. 4:1–4:17
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

for existential pivots and handling universal variables by universal reduction. A stronger calculus QU-Resolution [28] also allows universal pivots in resolution steps (and this is perhaps the most natural QBF resolution system from a logical perspective [7, 9]). Yet another generalisation is provided in the form of long-distance Q-Resolution [3] which allows certain merging steps forbidden in Q-Resolution. As mentioned above, QCDCL traces can be efficiently transformed into long-distance Q-Resolution proofs and this was in fact the reason for creating that proof system.

A recent line of research has aimed at understanding the precise relationship between QCDCL and QBF resolution [10, 16–18, 20]. The findings so far reveal both similarities to the tight relation between CDCL and resolution in SAT as well as crucial differences. While the first work [20] by Janota on this topic showed that practical (deterministic) QCDCL is exponentially weaker than Q-Resolution, the paper [10] demonstrated that QCDCL – even in its non-deterministic version – is incomparable to Q-resolution. This also implies that (non-deterministic) QCDCL is exponentially weaker than long-distance Q-Resolution. This is in sharp contrast to the equivalence of SAT and resolution in the propositional case [1, 5, 26], as explained above.

These results were strengthened in [16] by developing a lower-bound technique for QCDCL via a new notion of gauge, by which a number of lower bounds for QCDCL can be demonstrated (which not necessarily hinge on any QBF resolution hardness). Further, [17, 18] showed that several QCDCL variants, utilising e.g. cube learning, pure-literal elimination, and different decision strategies give rise to proof systems of different strength.

1.1 Our contributions

In this paper we continue this recent line of research to try to understand to precisely determine the relationship of QCDCL variants and different QBF resolution systems. The central quest of our research here is to find different QCDCL variants that are as strong as QU-Resolution and long-distance Q-Resolution. While we do not claim that these new algorithms are of immediate practical interest, we believe it is important to theoretically gauge the full potential of QCDCL. Our results can be summarised as follows.

(a) New QCDCL versions. We realise that there are at least three crucial QCDCL components that determine the strength of the algorithm. These are (i) whether decisions are made according to the prefix or not (policies LEV-ORD or ANY-ORD), (ii) whether unit propagation always or never includes universal reduction (policies ALL-RED, NO-RED) or whether this can be freely chosen at each propagation (ANY-RED), and (iii) whether unit propagation can propagate only existential variables (as in practical QCDCL, policy EXI-PROP) or whether also universal variables can be propagated (ALL-PROP).

While some of these policies were already defined and investigated in earlier works [10, 17, 18], the policies ANY-RED and ALL-PROP are considered here for the first time. We note that a solver implementing the strategy ALL-PROP together with LEV-ORD and NO-RED was recently presented by Slivovsky [27] (in fact this motivated our definition of the policies EXI-PROP and ALL-PROP). We demonstrate that in principle, all the aforementioned policies can be combined to yield sound and complete QCDCL algorithms (Proposition 8). We denote these as e.g. $\mathsf{QCDCL}^{\textsc{Lev-Ord}}_{\textsc{All-Red},\textsc{Exi-Prop}}$ (this combination models standard QCDCL).

(b) Characterisation of QBF proof systems. In our main result we tightly characterise the proof systems QU-Resolution by $\mathsf{QCDCL}^{\textsc{Any-Ord}}_{\textsc{No-Red},\textsc{All-Prop}}$ as well as (a slight variant of) long-distance Q-Resolution by $\mathsf{QCDCL}^{\textsc{Any-Ord}}_{\textsc{Any-Red},\textsc{Exi-Prop}}$ (Proposition 18 and Theorem 25). These results are similar in spirit (and proof method) to the characterisation of propositional resolution by CDCL [26] and Q-Resolution by $\mathsf{QCDCL}^{\textsc{Any-Ord}}_{\textsc{No-Red},\textsc{Exi-Prop}}$ [10]. However, quite some technical care is needed for the simulations to go through with the modified policies, for which we use the new notion of a blockade (Definition 22).

The mentioned variant of long-distance Q-Resolution – called mLD-Q-Res (for modified long-distance Q-Resolution, Definition 17) – is defined such as to contain exactly those steps that are needed for clause learning in standard QCDCL. The original definition of long-distance Q-resolution also allows some merging steps that do not occur in clause learning (those that have merged literals left of the pivot in both clauses). We leave open whether mLD-Q-Res is indeed weaker or equivalent to long-distance Q-Resolution (cf. Section 6).

(c) Separations between QCDCL variants. We clarify the joint simulation order of QBF resolution and QCDCL systems (cf. Figure 1 for an overview depicting known and new results). In general, the emerging picture shows that different choices of policies lead to incomparable systems (and could thus in principle be exploited for gains in practical solving over currently used QCDCL, cf. [18, 27]).

One set of results that we highlight here concerns the new system $\mathsf{QCDCL}^{\textsc{Lev-Ord}}_{\textsc{Any-Red},\textsc{Exi-Prop}}$, which we show to be strictly stronger than standard QCDCL, yet still weaker than mLD-Q-Res (and incomparable to Q-Resolution). To show that the system is strictly stronger than standard QCDCL (= $\mathsf{QCDCL}^{\textsc{Lev-Ord}}_{\textsc{All-Red},\textsc{Exi-Prop}}$), we exhibit some new family of QBFs which we show to be hard under the ALL-RED or NO-RED policies, yet tractable under ANY-RED.

1.2 Organisation

The remainder of this paper is organised as follows. We start by reviewing some notions from QBFs and QBF resolution systems in Section 2. In Section 3 we review the existing QCDCL models and define our variants. In Section 4 we investigate the simulation order of the QCDCL proof systems and show various separations. In Section 5 we obtain our main results, the characterisation of the proof systems QU-Res and mLD-Q-Res. We conclude in Section 6 with some open questions.

2 Preliminaries

Propositional and quantified formulas. Variables x and negated variables $\bar{x}$ are called *literals*. We denote the corresponding variable as $\mathrm{var}(x) := \mathrm{var}(\bar{x}) := x$.

A *clause* is a disjunction of literals, but we will sometimes interpret them as sets of literals on which we can perform set-theoretic operations. A *unit clause* (ℓ) is a clause that consists of only one literal. The *empty clause* consists of zero literals, denoted $(\bot)$. We sometimes interpret $(\bot)$ as a unit clause with the "empty literal" $\bot$. A clause C is called *tautological* if $\{\ell, \bar{\ell}\} \subseteq C$ for some literal ℓ. Alternatively, we will sometimes write $\ell^* \in C$ instead of $\{\ell, \bar{\ell}\} \subseteq C$.

A *cube* is a conjunction of literals and can also be viewed as a set of literals. We define a *unit cube* of a literal ℓ, denoted by $[\ell]$, and the *empty cube* $[\top]$ with "empty literal" $\top$. A cube D is *contradictory* if $\{\ell, \bar{\ell}\} \subseteq D$ for some literal ℓ. If C is a clause or a cube, we define $\mathrm{var}(C) := \{\mathrm{var}(\ell) : \ell \in C\}$. The negation of a clause $C = \ell_1 \vee \ldots \vee \ell_m$ is the cube $\neg C := \overline{C} := \bar{\ell}_1 \wedge \ldots \wedge \bar{\ell}_m$.

A *(total) assignment* σ of a set of variables V is a non-tautological set of literals such that for all $x \in V$ there is some $\ell \in \sigma$ with $\mathrm{var}(\ell) = x$. A *partial assignment* σ of V is an assignment of a subset $W \subseteq V$. A clause C is *satisfied* by an assignment σ if $C \cap \sigma \neq \emptyset$.

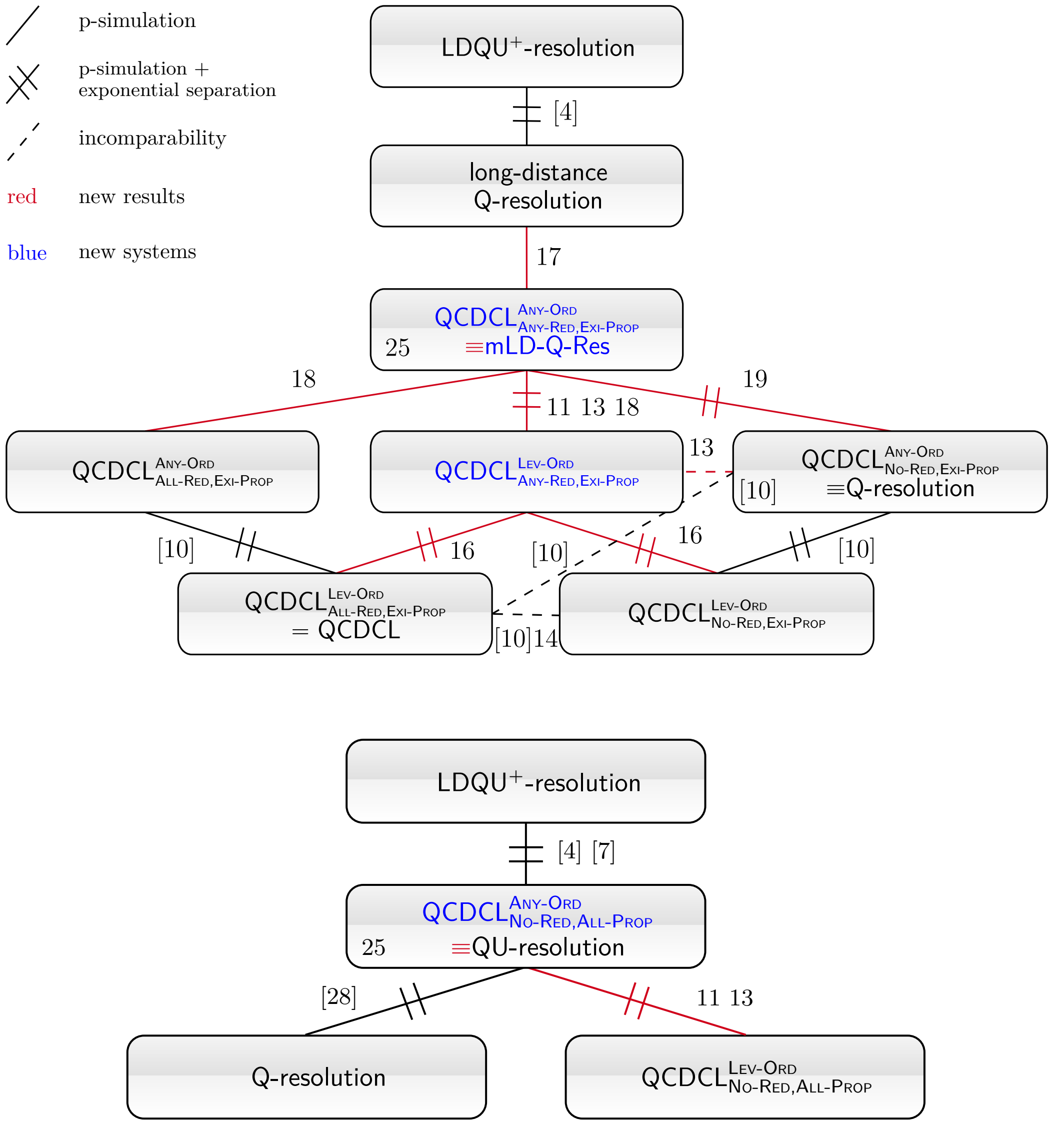

Figure 1 Hasse diagrams of the simulation order of QCDCL with EXI-PROP (above) and ALL-PROP (below) plus corresponding proof systems. Blue names represent new systems introduced here. Numbers in brackets are external references, while numbers without brackets are lemmas, propositions or theorems of this paper.

A cube D is *falsified* by σ if $\neg D \cap \sigma \neq \varnothing$. A clause C that is not satisfied by σ can be *restricted* by σ, defined as $C|_\sigma := \bigvee_{\ell \in C, \bar{\ell} \notin \sigma} \ell$. Similarly we can restrict a non-falsified cube D as $D|_\sigma := \bigwedge_{\ell \in D \setminus \sigma} \ell$. Intuitively, an assignment sets all its literals to true.

A *CNF* (conjunctive normal form) is a conjunction of clauses and a *DNF* (disjunctive normal form) is a disjunction of cubes. We restrict a CNF (resp. DNF) ϕ by an assignment σ as $\phi|_\sigma := \bigwedge_{C \in \phi \text{ non-satisfied}} C|_\sigma$ (resp. $\phi|_\sigma := \bigvee_{D \in \phi \text{ non-falsified}} D|_\sigma$). For a CNF (DNF) ϕ and an assignment σ, if $\phi|_\sigma = \varnothing$, then ϕ is *satisfied* (*falsified*) by σ.

A *QBF* (quantified Boolean formula) $\Phi = \mathcal{Q} \cdot \phi$ consists of a propositional formula ϕ, called the *matrix*, and a *prefix* $\mathcal{Q}$. A *prefix* $\mathcal{Q} = \mathcal{Q}'_1 V_1 \dots \mathcal{Q}'_s V_s$ consists of non-empty and pairwise disjoint sets of variables $V_1, \dots, V_s$ and quantifiers $\mathcal{Q}'_1, \dots, \mathcal{Q}'_s \in \{\exists, \forall\}$ with $\mathcal{Q}'_i \neq \mathcal{Q}'_{i+1}$ for $i \in [s-1]$. For a variable x in $\mathcal{Q}$, the *quantifier level* is $\text{lv}(x) := \text{lv}_\Phi(x) := i$, if $x \in V_i$. For $\text{lv}_\Phi(\ell_1) < \text{lv}_\Phi(\ell_2)$ we write $\ell_1 <_\Phi \ell_2$, while $\ell_1 \leqslant_\Phi \ell_2$ means $\ell_1 <_\Phi \ell_2$ or $\ell_1 = \ell_2$.

For a QBF $\Phi = \mathcal{Q} \cdot \phi$ with ϕ a CNF, we call Φ a *QCNF*. We define $\mathfrak{C}(\Phi) := \phi$. The QBF Φ is an *AQBF* (augmented QBF), if $\phi = \psi \vee \chi$ with CNF ψ and DNF χ. Again we write $\mathfrak{C}(\Phi) := \psi$ and $\mathfrak{D}(\Phi) := \chi$. We will sometimes interpret QCNFs as sets of clauses and cubes. If Φ is a QCNF or AQBF, we define $\mathrm{var}(\Phi) := \bigcup_{C \in \Phi} \mathrm{var}(C)$.

We restrict a QCNF $\Phi = \mathcal{Q} \cdot \phi$ by an assignment σ as $\Phi|_\sigma := \mathcal{Q}|_\sigma \cdot \phi|_\sigma$, where $\mathcal{Q}|_\sigma$ is obtained by deleting all variables from $\mathcal{Q}$ that appear in σ. Analogously, we restrict an AQBF $\Phi = \mathcal{Q} \cdot (\psi \vee \chi)$ as $\Phi|_\sigma := \mathcal{Q}|_\sigma \cdot (\psi|_\sigma \vee \chi|_\sigma)$.

If L is a set of literals (e.g., an assignment), we can get the negation of L, which we define as $\neg L := \overline{L} := \{\bar{\ell} |\, \ell \in L\}$.

(Long-distance) Q-resolution. Let C_1 and C_2 be two clauses from a QCNF or AQBF Φ. Let ℓ be an existential literal with $\mathrm{var}(\ell) \notin \mathrm{var}(C_1) \cup \mathrm{var}(C_2)$. The *resolvent* of $C_1 \vee \ell$ and $C_2 \vee \bar{\ell}$ over ℓ is defined as

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

Let $C := \ell_1 \vee \ldots \vee \ell_m$ be a clause from a QCNF or AQBF Φ such that $\ell_i \leqslant_\Phi \ell_j$ for all $i < j$, while $i, j \in [m]$. Let k be minimal such that $\ell_k, \ldots, \ell_m$ are universal. Then we can perform a *universal reduction* step and obtain

$$\mathrm{red}^\forall_\Phi(C) := \ell_1 \vee \ldots \vee \ell_{k-1}.$$

If it is clear that C is a clause, we can just write $\mathrm{red}_\Phi(C)$ or even $\mathrm{red}(C)$, if the QBF Φ is also obvious. We will write $\mathrm{red}(\Phi) = \mathrm{red}_\Phi(\Phi)$, if we reduce all clauses of Φ according to its prefix.

We can also perform *partial universal reduction*. Let K is a non-tautological set of literals and let $C := \ell_1 \vee \ldots \vee \ell_m$ be a clause from a QCNF Φ such that

$$\{\ell_k, \ldots, \ell_m\} = \{\ell \in C |\, \ell \in K,\ \ell \text{ is universal and } x <_\Phi \ell \text{ for all existential } x \in C\}.$$

Then we can partially reduce C by K and obtain

$$\mathrm{red}^\forall_{\Phi,K}(C) := \ell_1 \vee \ldots \vee \ell_{k-1}.$$

Intuitively, we will reduce all reducible literals that are also contained in K.

As before, we simply write red_K instead of $\mathrm{red}^\forall_{\Phi,K}$ if the context is clear.

As defined by Kleine Büning et al. [22], a Q-resolution proof π from a QCNF or AQBF Φ of a clause C is a sequence of clauses $\pi = (C_i)_{i=1}^m$, such that $C_m = C$ and for each C_i one of the following holds:

- *Axiom:* $C_i \in \mathfrak{C}(\Phi)$;
- *Resolution:* $C_i = C_j \overset{x}{\bowtie}_\Phi C_k$ with x existential, $j, k < i$, and C_i non-tautological;
- *Reduction:* $C_i = \mathrm{red}^\forall_\Phi(C_j)$ for some $j < i$.

[3] introduced an extension of Q-resolution proofs to long-distance Q-resolution proofs by replacing the resolution rule by

- *Resolution (long-distance):* $C_i = C_j \overset{x}{\bowtie} C_k$ with x existential and $j, k < i$. The resolvent C_i is allowed to contain tautologies such as $u \vee \bar{u}$, if u is universal. If there is such a universal $u \in \mathrm{var}(C_j) \cap \mathrm{var}(C_k)$, then we require $x <_\Phi u$.

The work [28] presented a further extension for Q-resolution, called QU-resolution, where we can also resolve over universal literals. Formally, it replaces the resolution rule by

- *Resolution (QU-Res)*: $C_i = C_j \overset{x}{\bowtie}_\Phi C_k$ with x existential *or* universal, $j, k < i$, and C_i non-tautological.

In [4], long-distance Q-resolution and QU-resolution were combined into a new proof system: long-distance QU$^+$-resolution. The resolution rule is as follows:

- *Resolution (long-distance QU$^+$-Res):* $C_i = C_j \overset{x}{\bowtie} C_k$ with x existential *or* universal and $j, k < i$. The resolvent C_i is allowed to contain tautologies such as $u \vee \bar{u}$, if u is universal. If there is a such a universal $u \in \text{var}(C_j) \cap \text{var}(C_k)$, then we require $\text{index}(x) < \text{index}(u)$, where index( ) is a fixed total order on the variables of Φ such that $\text{index}(a_1) < \text{index}(a_2)$ whenever $a_1 <_\Phi a_2$ for variables a_1, a_2 of Φ.

A Q-resolution (resp. long-distance Q-resolution, QU-resolution or long-distance QU$^+$-resolution) proof from Φ of the empty clause ($\bot$) is called a *refutation* of Φ. In that case, Φ is called *false*. We will sometimes interpret π as a set of clauses.

For the sake of completeness, we note that the above described proof systems are refutational proof systems that cannot be used to prove the truth of a QBF. For that, we would need analogously defined proof systems that work on cubes instead of clauses. For these proof systems, it is common to use the notion *consensus* instead of resolution, as well as *verification* instead of *refutation*. However, as we will purely concentrate on false formulas in this paper, we omit defining these aspects in more detail.

A proof system P *p-simulates* a system Q, if every Q proof can be transformed in polynomial time into a P proof of the same formula. P and Q are *p-equivalent* (denoted $P \equiv_p Q$) if they p-simulate each other.

3 Our QCDCL models

First, we need to formalise QCDCL procedures as proof systems in order to analyse their complexity. We follow the approach initiated in [10, 16–18].

We store all relevant information of a QCDCL run in *trails*. Since QCDCL uses several runs and potentially also restarts, a QCDCL proof will typically consist of many trails.

▶ **Definition 1** (trails). *A* trail $\mathcal{T}$ *for a QCNF or AQBF Φ is a (finite) sequence of pairwise distinct literals from Φ, including the empty literals $\bot$ and $\top$. Each two literals in $\mathcal{T}$ have to correspond to pairwise distinct variables from Φ. In general, a trail has the form*

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

where the d_i are decision literals *and $p_{(i,j)}$ are* propagated literals*. Decision literals are written in **boldface**. We use a semicolon before each decision to mark the end of a decision level. If one of the empty literals $\bot$ or $\top$ is contained in $\mathcal{T}$, then it has to be the last literal $p_{(r,g_r)}$. In this case, we say that $\mathcal{T}$ has* run into a conflict*.*

Trails can be interpreted as non-tautological sets of literals, and therefore as (partial) assignments. We write $x <_\mathcal{T} y$ if $x, y \in \mathcal{T}$ and x is left of y in $\mathcal{T}$. Furthermore, we write $x \leqslant_\mathcal{T} y$ if $x <_\mathcal{T} y$ or $x = y$.

As trails are produced gradually from left to right in an algorithm, we define $\mathcal{T}[i, j]$ for $i \in \{0, \ldots, r\}$ and $j \in \{0, \ldots, g_i\}$ as the subtrail that contains all literals from $\mathcal{T}$ up to (and excluding) $p_{(i,j)}$ (resp. d_i, if $j = 0$) in the same order. Intuitively, $\mathcal{T}[i, j]$ is the state of the trail before we assigned the literal at the point $[i, j]$ (which is $p_{(i,j)}$ or d_i).

For each point $[i, j]$ in the trail there must exist a set of literals $K_{(i,j)}$ which we call the reductive set at point *$[i, j]$. Intuitively, $K_{(i,j)}$ contains all literals that are reduced directly before the point $[i, j]$. The sets $K_{(i,j)}$ depend on the QCDCL variant (i.e., the reduction policy). Note that these sets are non-empty only if reduction is enabled.*

For each propagated literal $p_{(i,j)} \in \mathcal{T}$ *there has to be be a clause (or cube)* $ante_{\mathcal{T}}(p_{(i,j)})$ *such that* $red_{K_{(i,j)}}(ante_{\mathcal{T}}(p_{(i,j)})|_{\mathcal{T}[i,j]}) = (p_{(i,j)})$ *(or* $[p_{(i,j)}]$*). We call such a clause (cube) the* antecedent clause (cube) *of* $p_{(i,j)}$.

▶ Remark 2. In classic QCDCL, all $K_{(i,j)}$ are set to $\text{var}(\Phi) \cup \overline{\text{var}(\Phi)}$.

We state some general facts about trails and antecedent clauses/cubes.

▶ Remark 3. Let $\mathcal{T}$ be a trail, $\ell \in \mathcal{T}$ a propagated literal and $A := \text{ante}_{\mathcal{T}}(\ell)$.

- If ℓ is existential, then $\ell \in A$ and for each existential literal $x \in A$ with $x \neq \ell$ we need $\bar{x} <_{\mathcal{T}} \ell$.
- If ℓ is universal, then $\bar{\ell} \in A$ and for each universal literal $u \in A$ with $u \neq \bar{\ell}$ we need $u <_{\mathcal{T}} \ell$.

▶ **Definition 4** (natural trails)**.** *We call a trail* $\mathcal{T}$ natural *for formula* Φ*, if for each* $i \in \{0, \dots, r\}$ *the formula* $red_{K_{(i,0)}}(\Phi|_{\mathcal{T}[i,0]})$*, contains unit or empty constraints. Furthermore, the formula* $red_{K_{(i,j)}}\Phi|_{\mathcal{T}[i,j]}$ *must not contain empty constraints for each* $i \in [r]$*,* $j \in [g_i]$*, except* $[i, j] = [r, g_r]$*. Intuitively, this means that decisions are only made if there are no more propagations on the same decision level possible. Also, conflicts must be immediately taken care of.*

▶ Remark 5. Although it is allowed to define all sets $K_{(i,j)}$ differently, it might make sense from a practical perspective to weaken these possibilities. We point out three nuances of partial reduction in QCDCL that are interesting to consider:

(i) We change the reductive set after each propagation or decision step. That means that all sets $K_{(i,j)}$ might be different. This is the strongest possible version of partial reduction.

(ii) We only update the reductive set after backtracking. That means the sets $K_{(i,j)}$ are constant for each trail. It will turn out that this version is enough for our characterisation of mLD-Q-Res (cf. Theorem 25). Consequently, this version is as strong as version (i).

(iii) We never change the reductive set. That means that the sets $K_{(i,j)}$ remain constant throughout the whole QCDCL proof. This version is enough for the separation between systems with and systems without partial reduction (cf. Theorem 16).

▶ **Definition 6** (learnable constraints)**.** *Let* $\mathcal{T}$ *be a trail for* Φ *of the form* (1) *with* $p_{(r,g_r)} \in \{\bot, \top\}$*. Starting with* $ante_{\mathcal{T}}(\bot)$ *(resp.* $ante_{\mathcal{T}}(\top)$*) we reversely resolve with the antecedent clauses (cubes) that were used to propagate the existential (universal) variables, until we stop at some point. Literals that were propagated via cubes (clauses) will be interpreted as decisions. If a resolution step cannot be performed at some point due to a missing pivot, we simply skip that antecedent. The clause (cube) we so derive is a* learnable constraint*. We denote the sequence of learnable constraints by* $\mathfrak{L}(\mathcal{T})$*.*

We can also learn cubes from trails that did not run into conflict. If $\mathcal{T}$ *is a total assignment of the variables from* Φ*, then we define the set of learnable constraints as the set of cubes* $\mathfrak{L}(\mathcal{T}) := \{red^{\exists}_{\Phi}(D) | \; D \subseteq \mathcal{T}$ *and* D *satisfies* $\mathfrak{C}(\Phi)\}$*.*

Generally, we allow to learn an arbitrary constraint. However, for the characterisations, it suffices to concentrate on clause learning. Additionally, most of the time we will simply learn the clause which we obtain after propagation over every available literal in the trail. This clause can only consist of negated decision literals, and literals that were reduced during unit propagation. Since this is the last clause we can derive during clause learning in a trail $\mathcal{T}$, we will refer to that clause as the *rightmost clause in* $\mathcal{L}(\mathcal{T})$.

▶ **Definition 7** (QCDCL proof systems). *Let* $\mathsf{D} \in \{\textsf{LEV-ORD}, \textsf{ANY-ORD}\}$ *a decision policy,* $\mathsf{R} \in \{\textsf{ALL-RED}, \textsf{NO-RED}, \textsf{ANY-RED}\}$ *a reduction policy and* $\mathsf{P} \in \{\textsf{EXI-PROP}, \textsf{ALL-PROP}\}$ *a propagation policy (all defined below). A* $\mathsf{QCDCL}^{\mathsf{D}}_{\mathsf{R,P}}$ *proof* ι *from a QCNF* $\Phi = \mathcal{Q} \cdot \phi$ *of a clause or cube* C *is a (finite) sequence of triples*

$$\iota := [(\mathcal{T}_i, C_i, \pi_i)]_{i=1}^{m},$$

where $C_m = C$*, each* $\mathcal{T}_i$ *is a trail for* Φ_i *that follows the policies* D*,* R *and* P*, each* $C_i \in \mathfrak{L}(\mathcal{T}_i)$ *is one of the constraints we can learn from each trail and* π_i *is the proof from* Φ_i *of* C_i *we obtain by performing the steps described in Definition 6, where* Φ_i *are AQBFs that are defined recursively by setting* $\Phi_1 := \mathcal{Q} \cdot (\mathfrak{C}(\Phi) \vee \varnothing)$ *and*

$$\Phi_{j+1} := \begin{cases} \mathcal{Q} \cdot ((\mathfrak{C}(\Phi_j) \wedge C_j) \vee \mathfrak{D}(\Phi_j)) & \text{if } C_j \text{ is a clause,} \\ \mathcal{Q} \cdot (\mathfrak{C}(\Phi_j) \vee (\mathfrak{D}(\Phi_j) \vee C_j)) & \text{if } C_j \text{ is a cube,} \end{cases}$$

for $j = 1, \dots, m-1$*. If necessary, we set* $\pi_i := \varnothing$*.*

We now explain the three types of policies:

- *Decision policies:*
  - LEV-ORD*: For each decision* d_i *we have that* $lv_{\Psi|_{\mathcal{T}[i,0]}}(d_i) = 1$*. I.e., decisions are level-ordered.*
  - ANY-ORD*: Decisions can be made arbitrarily in any order.*
- *Reduction policies:*
  - ALL-RED*: All* $K_{(i,j)}$ *are set to* $var(\Phi) \cup \overline{var(\Phi)}$*. This is the classic setting – we have to reduce all reducible literals during unit propagation.*
  - NO-RED*: All* $K_{(i,j)}$ *are set to* $\varnothing$*. We are not allowed to reduce during unit propagation at all. There is one exception: Combined with* ALL-PROP*, we are allowed (but not forced) to reduce universal unit clauses (existential unit cubes) and immediately obtain a conflict. This is due to reasons of completeness which will be explained later.*
  - ANY-RED*: The sets* $K_{(i,j)}$ *can be set arbitrarily. Hence, we can choose after each propagation or decisions step which literals are to be reduced next.*
- *Propagation policies:*
  - EXI-PROP*: Unit clauses (cubes) can only propagate existential (universal) literals. Universal (existential) unit clauses (cubes) will be reduced to the empty clause (cube) if allowed by the reduction policy.*
  - ALL-PROP*: Universal (existential) unit clauses (cubes) will lead to the propagation of the universal (existential) unit literal. This policy is nullified if combined with* ALL-RED*. If combined with* NO-RED*, we are allowed to reduce universal (existential) unit clauses (cubes) instead of doing a unit propagation. This is due to reasons of completeness.*

Having defined all policies, we can now denote trails that follow the policies D*,* R *and* P *as* $\mathsf{QCDCL}^{\mathsf{D}}_{\mathsf{R,P}}$ trails*.*

We require that $\mathcal{T}_1$ *is a natural* $\mathsf{QCDCL}^{\mathsf{D}}_{\mathsf{R,P}}$ *trail and for each* $2 \leqslant i \leqslant m$ *there is a point* $[a_i, b_i]$ *such that* $\mathcal{T}_i[a_i, b_i] = \mathcal{T}_{i-1}[a_i, b_i]$ *and* $\mathcal{T}_i \backslash \mathcal{T}_i[a_i, b_i]$ *has to be a natural* $\mathsf{QCDCL}^{\mathsf{D}}_{\mathsf{R,P}}$ *trail for* $\Phi_i|_{\mathcal{T}_i[a_i,b_i]}$*. This process is called* backtracking*. If* $\mathcal{T}_{i-1}[a_i, b_i] = \varnothing$*, then this is also called a* restart*.*

If $C = C_m = (\bot)$*, then* ι *is called a* $\mathsf{QCDCL}^{\mathsf{D}}_{\mathsf{R,P}}$ refutation *of* Φ*. If* $C = C_m = [\top]$*, then* ι *is called a* $\mathsf{QCDCL}^{\mathsf{D}}_{\mathsf{R,P}}$ verification *of* Φ*. The proof ends once we have learned* $(\bot)$ *or* $[\top]$*.*

If C *is a clause, we can stick together the* long-distance Q-resolution *derivations from* $\{\pi_1, \dots, \pi_m\}$ *and obtain a* long-distance Q-resolution *proof from* Φ *of* C*, which we call* $\mathfrak{R}(\iota)$*.*

The size *of* ι *is defined as* $|\iota| := \sum_{i=1}^{m} |\mathcal{T}_i|$*. Obviously, we have* $|\mathfrak{R}(\iota)| \in \mathcal{O}(|\iota|)$*.*

We can show that all combinations of the above policies lead to sound and complete proof systems (and algorithms).

▶ **Proposition 8.** *All defined QCDCL variants are sound and complete.*

Proof. It suffices to show completeness for the weakest combinations. Hence, we can use LEV-ORD and choose between ALL-RED and NO-RED, as both are subsumed by ANY-RED. For EXI-PROP, completeness was already shown in [10]. For ALL-PROP, we distinguish two cases:

(i) ALL-RED: Then we will never propagate universal (existential) literals via clauses (cubes), as they will always be directly reduced to the empty clause (cube). Hence, this system is the same as if we would have chosen EXI-PROP.

(ii) NO-RED: We are not forced to do universal (existential) propagations via clauses (cubes). Therefore, the version with EXI-PROP is already simulated by this combination system.

The soundness follows from the soundness of long-distance QU^+-resolution (long-distance QU^+-consensus) proofs, which can be extracted from all QCDCL variants defined here. ◀

4 The simulation order of QCDCL proof systems

While the policies ALL-RED and NO-RED were already introduced in work (cf. [10]), in which an incomparability between these two models was shown, it is natural to analyse their relation to our new policy ANY-RED. Obviously, ANY-RED covers (hence: simulates) both ALL-RED an NO-RED, as we can simply choose to reduce everything or nothing. We want to prove now that both ALL-RED and NO-RED are exponentially worse than ANY-RED on some family of QBFs. I.e., we want to show that there exist formulas where we need to reduce *some* but not *all* literals during unit propagation.

These formulas will be hand-crafted, consisting of two already well-known QCNFs, named $\texttt{MirrorCR}_n$, which is a modified version of the Completion Principle [21], and $\texttt{QParity}_n$ [12].

▶ **Definition 9** ([17]). *The QCNF* $\texttt{MirrorCR}_n$ *consists of the prefix* $\exists T \forall u \exists T$*, where* $X := \{x_{(1,1)}, \dots, x_{(n,n)}\}$ *and* $T := \{a_1, \dots, a_n, b_1, \dots, b_n\}$*, and the matrix*

$$\begin{array}{lllll} x_{(i,j)} \vee u \vee a_i & \bar{a}_1 \vee \dots \vee \bar{a}_n & x_{(i,j)} \vee \bar{u} \vee \bar{a}_i & a_1 \vee \dots \vee a_n & \\ \bar{x}_{(i,j)} \vee \bar{u} \vee b_j & \bar{b}_1 \vee \dots \vee \bar{b}_n & \bar{x}_{(i,j)} \vee u \vee \bar{b}_j & b_1 \vee \dots \vee b_n & \text{for } i,j \in [n]. \end{array}$$

The reason why we use $\texttt{MirrorCR}_n$ instead of $\texttt{CR}_n$ is because its matrix is unsatisfiable. That means that cube learning, which might have a positive effect on $\texttt{CR}_n$ (note that there are false QCNFs that become easy with cube learning [17]) is now completely unavailable. Additionally, we can now guarantee to always get a conflict once all variables from $\texttt{MirrorCR}_n$ got assigned.

▶ **Lemma 10** ([17]). *The matrix* $\mathfrak{C}(\texttt{MirrorCR}_n)$ *of* $\texttt{MirrorCR}_n$ *is unsatisfiable as a propositional formula.*

As $\texttt{MirrorCR}_n$ is simply an extension of the Completion Principle ($\texttt{CR}_n$), which is known to be easy for Q-resolution [21], we can simply reuse the exact same refutation from [21]. Note that we do not need all axiom clauses to refute the formula.

▶ **Proposition 11** ([17]). *The QBFs* $\texttt{MirrorCR}_n$ *have polynomial-size* Q-resolution *refutations.*

▶ **Definition 12** ([12]). *The QCNF* $\texttt{QParity}_n(Y, w, S)$ *consists of the prefix* $\exists Y \forall w \exists S$*, where* $Y := \{y_1, \dots, y_n\}$ *and* $S := \{s_2, \dots, s_n\}$*, and the matrix*

$$\begin{array}{lllll} y_1 \vee y_2 \vee \bar{s}_2 & y_1 \vee \bar{y}_2 \vee s_2 & \bar{y}_1 \vee y_2 \vee s_2 & \bar{y}_1 \vee \bar{y}_2 \vee \bar{s}_2 & \\ y_i \vee s_{i-1} \vee \bar{s}_i & y_i \vee \bar{s}_{i-1} \vee s_i & \bar{y}_i \vee s_{i-1} \vee s_i & \bar{y}_i \vee \bar{s}_{i-1} \vee \bar{s}_i & \text{for } i \in \{2, \dots, n\}, \\ s_n \vee w & \bar{s}_n \vee \bar{w}. & & & \end{array}$$

When introduced in [17], it was shown that $\texttt{MirrorCR}_n$ is hard for all QCDCL models with level-ordered decisions considered in [17]. We generalize this result and show that the lower bound for $\texttt{MirrorCR}_n$ indeed only depends on the decision policy used and also holds for our new models introduced here.

▶ **Proposition 13.** *The QBFs* $\texttt{MirrorCR}_n(X, u, T)$ *need exponential-sized refutations in all our QCDCL variants with the* LEV-ORD *policy.*

Proof. (Sketch) We recall the hardness results of $\texttt{MirrorCR}_n$ for classical QCDCL in [17], which were independent of the reduction policy. One can also show that it is impossible to propagate universal literals, therefore the propagation policies do not matter, either. ◀

With the QBFs $\texttt{QParity}_n$ one obtains one direction of the incomparability between classical QCDCL (here called $\mathsf{QCDCL}^{\textsc{Lev-Ord}}_{\textsc{All-Red,Exi-Prop}}$) and Q-resolution, being easy for the former and hard for the latter system.

▶ **Theorem 14** ([10, 13])**.** *The QBFs* $\texttt{QParity}_n$ *need exponential-sized* Q-resolution *and* QU-resolution *refutations, but admit polynomial-sized* $\mathsf{QCDCL}^{\textsc{Lev-Ord}}_{\textsc{All-Red,Exi-Prop}}$ *refutations.*

We combine the $\texttt{MirrorCR}$ and $\texttt{QParity}$ formulas into a new one, using auxiliary variables.

▶ **Definition 15.** *The QBF* $\texttt{MiPa}_n$ *consists of the prefix* $\forall z \exists X \forall u \exists T \forall p \exists Y \forall w \exists S \forall v \exists r$ *such that* X, u, T *are the variables for* $\texttt{MirrorCR}_n(X, u, T)$, *and* Y, w, S *are the variables for* $\texttt{QParity}_n(Y, w, S)$. *The matrix of* $\texttt{MiPa}_n$ *contains the clauses*

$$z \vee \bar{r}, \quad \bar{z} \vee \bar{r}$$

$$\left.\begin{array}{l} C \vee p \vee v \vee r \\ C \vee p \vee \bar{v} \vee r \\ C \vee \bar{p} \vee v \vee r \\ C \vee \bar{p} \vee \bar{v} \vee r \end{array}\right\} \text{for } C \in \mathfrak{C}(\texttt{MirrorCR}_n(X, U, T)),$$

$$\left.\begin{array}{l} p \vee D \\ \bar{p} \vee D \end{array}\right\} \text{for } D \in \mathfrak{C}(\texttt{QParity}_n(Y, w, S)).$$

We show next that $\texttt{MiPa}_n$ needs indeed ANY-RED in order to admit polynomial-size refutations in QCDCL. The idea is that ALL-RED will always lead to refutations of $\texttt{MirrorCR}_n$, and NO-RED will alternatively lead to Q-resolution refutations of $\texttt{QParity}_n$, which are both of exponential size.

▶ **Theorem 16.** *The QBFs* $\texttt{MiPa}_n$

(i) *need exponential-size* $\mathsf{QCDCL}^{\textsc{Lev-Ord}}_{\textsc{All-Red,Exi-Prop}}$ *refutations,*

(ii) *need exponential-size* $\mathsf{QCDCL}^{\textsc{Lev-Ord}}_{\textsc{No-Red,Exi-Prop}}$ *refutations,*

(iii) *but have polynomial-size* $\mathsf{QCDCL}^{\textsc{Lev-Ord}}_{\textsc{Any-Red,Exi-Prop}}$ *refutations.*

Proof. For (i), since the formula has no unit clauses, we have to start by deciding the variable z. Because z occurs symmetrically in $\texttt{MiPa}_n$, we can assume that we set z to true. This always triggers the unit propagation of $\bar{r}$ via the clause $\bar{z} \vee \bar{r}$. After that, we are forced to assign the variables from X, $U := \{u\}$ and T along the quantification order. Since the matrix of $\texttt{MirrorCR}_n$ is unsatisfiable, and we need to reduce all literals if possible, we will detect a conflict at the same time as we would get the conflict in $\texttt{MirrorCR}_n$ itself. The proof we can extract from the trails is essentially a $\mathsf{QCDCL}^{\textsc{Lev-Ord}}_{\textsc{All-Red,Exi-Prop}}$ refutation of $\texttt{MirrorCR}_n$, except that it additionally contains the variables z, p, v and r in some polarities. However, this does not change the fact that we can still not resolve two clauses that contain X-, U-, and T-variables over any X-variable. Therefore, if we shorten the proof by assigning r to false and z to true, we get a refutation of $\texttt{MirrorCR}_n$, in which we never resolve two clauses that contain X-, U-, and T-variables over an X-variable. This property is called *primitive* (cf. [16]). Also in [16], it was shown that primitive Q-resolution refutations of $\texttt{MirrorCR}_n$ need exponential size.

For (ii), we start in the same way as in (i), but we do not get a conflict once we assigned all variables of $\texttt{MirrorCR}_n$. Next, we need to decide p in some polarity, but nothing will happen for the moment. We then start assigning the variables of $\texttt{QParity}_n$ along the quantification order. Now we have to distinguish two cases:

Case 1: We get a conflict in $\texttt{QParity}_n$. But then, because of NO-RED, we can only extract Q-resolution derivations of learned clauses. And if we get enough conflicts in $\texttt{QParity}_n$, we can essentially extract a Q-resolution refutation of $\texttt{QParity}_n$, which has exponential size.

Case 2: We do not get a conflict in $\texttt{QPartity}_n$. This might happen when the universal player assigns the variable w the "wrong" way. Then the only unassigned variable is v. After deciding it in any polarity, we will always get a conflict in $\texttt{MirrorCR}_n$. If we find enough conflicts in $\texttt{MirrorCR}_n$, we can essentially extract an exponential-size fully reduced primitive Q-resolution refutation of $\texttt{MirrorCR}_n$ as in (i).

Note that it is possible to get both kind of conflicts. However, it is only important with what kind of conflicts we were able to derive the empty clause.

Finally, for (iii), we can construct a polynomial-size $\mathsf{QCDCL}^{\textsc{Lev-Ord}}_{\textsc{Any-Red},\textsc{Exi-Prop}}$ proof by only reducing the literals w and $\bar{w}$. After deciding z, propagating $\bar{r}$, assigning all variables from X, u and T and deciding p arbitrarily, we can simply copy the polynomial-size $\mathsf{QCDCL}^{\textsc{Lev-Ord}}_{\textsc{All-Red},\textsc{Exi-Prop}}$ proof of $\texttt{QParity}_n$ (note that ALL-RED only applies to w and $\bar{w}$). At some point, we will derive the clause (p) or $(\bar{p})$, which can be reduced to the empty clause. ◀

One of the initial motivations of this paper was to find a way to p-simulate long-distance Q-resolution refutations of QCNFs by certain variants of QCDCL. However, it appears that not all resolution steps that are allowed in long-distance Q-resolution can be recreated with QCDCL proofs. In long-distance Q-resolution proofs that are extracted from QCDCL, one can easily observe that for each resolution step $C_1 \overset{\ell}{\bowtie} C_2$, at least one parent clause C_i has to be an antecedent clause for ℓ or $\bar{\ell}$ in the corresponding trail. In particular, there must be a partial assignment τ and a set of literals K such that $\mathrm{red}_K(C_i|_\tau)$ becomes unit, i.e. $\mathrm{red}_K(C_i|_\tau) = (\ell)$ (resp. $(\bar{\ell})$). This is not possible if there are tautologies left of ℓ in C_i that cannot be reduced.

Motivated by this observation, we introduce a new proof system similar to long-distance Q-resolution, but with the restriction that such a situation as described above is not allowed.

▶ **Definition 17.** *A* long-distance Q-resolution *proof is called a* mLD-Q-Res proof*, if it does not contain a resolution step between two clauses D and E, such that $C = D \overset{x}{\bowtie} E$ for an existential variable x and there are universal variables u, w such that $u^* \in D$, $w^* \in E$ and $lv_\Phi(u), lv_\Phi(w) < lv_\Phi(x)$.*

With this definition in place, we can show that mLD-Q-Res proofs can be extracted from runs of most variants of QCDCL that we defined. Further, for some QCDCL paradigms, stricter simulations hold.

▶ **Proposition 18.** *The following holds on false QCNFs:*
(i) Q-resolution *p-simulates* $\mathsf{QCDCL}^{\textsc{Any-Ord}}_{\textsc{No-Red},\textsc{Exi-Prop}}$.
(ii) QU-resolution *p-simulates* $\mathsf{QCDCL}^{\textsc{Any-Ord}}_{\textsc{No-Red},\textsc{All-Prop}}$.
(iii) mLD-Q-Res *p-simulates* $\mathsf{QCDCL}^{\textsc{Any-Ord}}_{\textsc{Any-Red},\textsc{Exi-Prop}}$.

Proof. Item (i) was already shown in [10].

For (ii), because of ALL-PROP, we might propagate (and resolve) over universal literals, which can be handled by QU-resolution. It remains to show that NO-RED prevents the derivation of tautological clauses. This holds because we only use antecedent clauses for clause learning. Let us assume we learn a tautological clause C from a $\mathsf{QCDCL}^{\text{ANY-ORD}}_{\text{NO-RED,ALL-PROP}}$ trail $\mathcal{T}$. Then there would be two antecedent clauses $D := \text{ante}_{\mathcal{T}}(\ell_1)$ and $E := \text{ante}_{\mathcal{T}}(\ell_2)$ such that there exists a universal literal u with $u \neq \ell_1$, $\bar{u} \neq \ell_2$, $u \in D$ and $\bar{u} \in E$. We need $\bar{u} \in \mathcal{T}$ for D to become unit and at the same time we need $u \in \mathcal{T}$ for E to become unit, which is not possible. Therefore, we will never derive tautological clauses.

Let us now prove (iii). By definition, we can extract long-distance Q-resolution proof from $\mathsf{QCDCL}^{\text{ANY-ORD}}_{\text{ANY-RED,EXI-PROP}}$ trails (note that we only propagate existential literals, hence we also only resolve over existential variables during clause learning). It remains to show that the kind of resolution step that is forbidden in mLD-Q-Res (but allowed in long-distance Q-resolution) will never occur during clause learning.

Assume it does. Then we have derived a clause C by resolving two clauses D and E over some literal x (hence $C = D \overset{x}{\bowtie} E$), such that there exists universal tautologies $u^* \in D$ and $w^* \in E$ with $u^* \neq w^*$ and $\text{lv}(u^*), \text{lv}(w^*) < \text{lv}(x)$. Then at least one of these parent clauses needs to be an antecedent clause for a trail $\mathcal{T}$, say $D = \text{ante}_{\mathcal{T}}(x)$. But then D can never become the unit clause (x), because we cannot reduce u^* since it is blocked by x, and we cannot falsify it by the previous trail assignment since it is a tautology. This is a contradiction that shows that all resolution and reduction steps are allowed in mLD-Q-Res. ◀

We could formulate analogous results on true QCNFs using the notation of consensus proofs. However, we will omit this as all separations and characterisations will be performed on false QCNFs and resolution proofs.

One can easily show that the separation between Q-resolution and long-distance Q-resolution transfers to a separation between Q-resolution and mLD-Q-Res.

▶ **Corollary 19.** mLD-Q-Res *p-simulates and is exponentially stronger than* Q-resolution*.*

Proof. The simulation follows by definition. The separation follows by Theorem 14 and Proposition 18 (iii). ◀

In fact, all currently known upper bounds for long-distance Q-resolution can be easily transformed into mLD-Q-Res upper bounds. However, we leave open the question whether long-distance Q-resolution is stronger than or equivalent to mLD-Q-Res.

5 Characterisations of QU-resolution and mLD-Q-Res

In this section, show that all the simulations in Proposition 18 can be tightened to equivalences. For this we will characterise both mLD-Q-Res and QU-resolution by the specific variants of QCDCL mentioned in Proposition 18. Characterising Q-resolution by $\mathsf{QCDCL}^{\text{ANY-ORD}}_{\text{NO-RED,EXI-PROP}}$ was already undertaken in [10]. However, we leave open, whether we can extend these characterisations to long-distance Q-resolution. This will depend on whether it is possible to polynomially transform the “forbidden” resolution steps that can occur in long-distance Q-resolution, but cannot be created by QCDCL, into mLD-Q-Res steps.

The characterisations follow the same idea as in [10], in which Q-resolution was characterised. One crucial difference is that we now want to use the ANY-RED policy, i.e., in each step we have to decide what literals to reduce.

As already mentioned in Remark 5, it suffices to update the reductive sets only after a conflict. That means that for characterising $\mathsf{mLD\text{-}Q\text{-}Res}$, it is enough to fix the literals that are going to be reduced throughout the whole trail. Thus, we introduce the notion of *L-reductive trails.*

▶ **Definition 20** (L-reductive trails)**.** *Let L be a set of literals. A trail $\mathcal{T}$ is called L-*reductive*, if for each propagation step in $\mathcal{T}$ the literals that were selected to be reduced are exactly the literals in L. Formally, this means that for each $p_{(i,j)}$ there is an antecedent clause (resp. cube) $ante_{\mathcal{T}}(p_{(i,j)})$ such that $red_L(ante_{\mathcal{T}}(p_{(i,j)})|_{\mathcal{T}[i,j]}) = (p_{(i,j)})$ (resp. $[\bar{p}_{(i,j)}]$).*

Before starting with a new L-reductive trail, we always need to consider the choice of the reductive set L. As we know from [10] and Proposition 18, tautologies can only be created when the corresponding literal got reduced somewhere in the trail. In fact, since $\mathsf{QCDCL}^{\textsc{Any-Ord}}_{\textsc{No-Red,Exi-Prop}}$ already characterises $\mathsf{Q\text{-}resolution}$ [10], we can conclude that in some sense the only purpose of reductions during unit propagation is to create tautological clauses. Therefore we will distinguish between the tautological and the non-tautological part of a clause.

▶ **Definition 21.** *Let C be a clause. Let $G(C) := \{u \in C : u$ is universal and $\bar{u} \in C\}$. This set is the* tautological part *of C. The* non-tautological part *$H(C)$ of C is defined as $H(C) := C \backslash G(C)$.*

For each $\mathsf{QU\text{-}resolution}$ proof π and $C \in \pi$ we have $G(C) = \varnothing$.

Our next notion is similar to the concepts of *unreliable* [10] and *1-empowering* [26].

▶ **Definition 22** (Blockades)**.** *Let $\mathsf{S} \in \{\mathsf{QCDCL}^{\textsc{Any-Ord}}_{\textsc{Any-Red,Exi-Prop}}, \mathsf{QCDCL}^{\textsc{Any-Ord}}_{\textsc{No-Red,All-Prop}}\}$ and C be a clause. A tuple $(\mathcal{U}, \alpha, \ell, K)$, where $\mathcal{U}$ is a trail, ℓ is a literal, α is a non-tautological set of literals and K is a set of universal literals, is called a* blockade of C with respect to S *for a QCNF $\Phi = \mathcal{Q} \cdot \phi$, if $\mathcal{U}$ is a K-reductive S trail with decisions α, such that $\ell \in C$, $\alpha \subseteq \overline{C} \backslash \{\bar{\ell}\}$, $K \subseteq G(C)$ and $\alpha \cap K = \varnothing$.*

For $\mathsf{S} = \mathsf{QCDCL}^{\textsc{Any-Ord}}_{\textsc{Any-Red,Exi-Prop}}$, we additionally require that ℓ is an existential literal and α consists of only existential literals.

▶ **Example 23.** Blockades occur when we are not able to choose all decisions from a pre-defined non-tautological set α. For example, consider the QCNF

$$\exists x, y \forall u, v \exists z \ (\bar{y} \vee \bar{z}) \wedge (\bar{x} \vee \bar{u} \vee z) \wedge (x \vee y \vee v \vee z) \wedge (y \vee \bar{v} \vee z).$$

Assume that we use $\mathsf{QCDCL}^{\textsc{Any-Ord}}_{\textsc{Any-Red,All-Prop}}$. Then the clause $C := \bar{x} \vee \bar{y} \vee u \vee \bar{u} \vee z$ has a blockade $(\mathcal{U}, \alpha, \ell, K)$ with $\mathcal{U} := (\mathbf{y}, \bar{z}, \bar{x})$, where $\text{ante}_{\mathcal{U}}(\bar{z}) = \bar{y} \vee \bar{z}$, $\text{ante}_{\mathcal{U}}(\bar{x}) = \bar{x} \vee \bar{u} \vee z$, as well as $\ell := \bar{x} \in C$, $\alpha := \{y\} \subseteq \overline{C} \backslash \{\bar{\ell}\}$ and $K := \{\bar{u}\}$.

Intuitively, this means that although the clause C is not directly contained in the formula, we are still able to detect the implication $(\alpha \wedge \overline{K} \to \ell) = (y \wedge u) \to \bar{x}$ (which is equivalent to $\bar{y} \vee \bar{u} \vee \bar{x} \subseteq C$) as a composition of decisions and unit propagations. It turns out that, instead of learning C directly, it is enough to detect a blockade in order to make use of C for unit propagations in later trails.

The next lemma shows, that we can recall trails (and blockades in particular), that were detected and stored at an earlier point, and restore all propagations they contained. This will be important for the characterisations, as we will go through the given proof, find blockades or conflicts for all clauses in that proof, and recall the corresponding trails (by using this Lemma) an all their containing propagations whenever the clauses are needed for another resolution step. In that way, we can virtually store previous trails and recall them later again as natural trails.

▶ **Lemma 24.** *Let $\Phi = \mathcal{Q} \cdot \phi$ and $\Psi = \mathcal{Q} \cdot \psi$ be QCNFs such that $\psi \subseteq \phi$.*

Let $\mathcal{U}$ be a K-reductive trail (for NO-RED *we set $K = \emptyset$) for the QCNF Ψ with decisions β. Let $\mathcal{T}$ be a natural L-reductive trail ($L = \emptyset$ for* NO-RED*) with decisions α for the QCNF Φ such that $K \subseteq L$, $\beta \subseteq \mathcal{T}$ and $\alpha \cap L = \emptyset$. If $\mathcal{T}$ does not run into a clause conflict, then all propagated literals from $\mathcal{U}$ are also contained in $\mathcal{T}$.*

Proof. Assume that $\mathcal{T}$ does not run into a clause conflict, but there are some propagated literals from $\mathcal{U}$ that are not contained in $\mathcal{T}$. Let $p_{(a,b)}$ be the literal that is leftmost in $\mathcal{U}$ with this property and define $A := \mathrm{ante}_{\mathcal{U}}(p_{(a,b)})$. Since there are no cubes present, we conclude that A must be a clause, regardless of whether $p_{(a,b)}$ is existential or universal.

Because $p_{(a,b)}$ is leftmost, all other propagated literals before $p_{(a,b)}$ in $\mathcal{U}$ are already contained in $\mathcal{T}$. Since $\mathcal{U}$ was K-reductive, we know that $\mathrm{red}_K(A|_{\mathcal{U}[a,b]}) = (p_{(a,b)})$. Because of $K \subseteq L$ and $\mathcal{U}[a,b] \subseteq \mathcal{T}$ we have either $\mathrm{red}_L(A|_{\mathcal{T}}) \in \{(p_{(a,b)}), (\bot)\}$, or $A|_{\mathcal{T}}$ becomes true. Note that we can set $K := L := \emptyset$ for the rest of our argumentation in the case where $p_{(a,b)}$ is universal.

The first case would contradict our assumption (since $\mathcal{T}$ is natural), therefore we have to assume that $A|_{\mathcal{T}}$ becomes true. This means that we can find a literal $p_{(a,b)} \neq u \in A \cap \mathcal{T}$. If u was existential, then we would need $\bar{u} \in \mathcal{U}[a,b]$. But this would also imply $\bar{u} \in \mathcal{T}$ which contradicts the fact that $u \in \mathcal{T}$. Hence u must be universal.

If u was a decision in $\mathcal{T}$, then we would have $u \in \alpha$. Because of $\alpha \cap L = \emptyset$ we conclude $u \notin L$ and also $u \notin K$. In order to make u vanish in $\mathrm{red}_K(A|_{\mathcal{U}[a,b]})$, we need $\bar{u} \in \mathcal{U}[a,b]$, hence also $\bar{u} \in \mathcal{T}$. However, this is a contradiction because we already assumed $u \in \mathcal{T}$.

Therefore, u must have been propagated by an antecedent clause $\mathrm{ante}_{\mathcal{T}}(u)$. But then we have $K = \emptyset$, hence $u \notin K$ and $\bar{u} \in \mathcal{U}[a,b] \subseteq \mathcal{T}$, which is a contradiction again because of $u \in \mathcal{T}$. ◀

▶ **Theorem 25.** *The following holds:*
- $\mathsf{QCDCL}^{\textit{Any-Ord}}_{\textit{Any-Red,Exi-Prop}}$ *p-simulates* mLD-Q-Res*.*
- $\mathsf{QCDCL}^{\textit{Any-Ord}}_{\textit{No-Red,All-Prop}}$ *p-simulates* QU-resolution*.*

In detail: Let $\Phi = \mathcal{Q} \cdot \phi$ be a QCNF in n variables and $\pi = D_1, \ldots, D_m$ be a mLD-Q-Res *(*QU-resolution*) refutation of Φ. Then we can construct a $\mathsf{QCDCL}^{\textit{Any-Ord}}_{\textit{Any-Red,Exi-Prop}}$ ($\mathsf{QCDCL}^{\textit{Any-Ord}}_{\textit{No-Red,All-Prop}}$) refutation ι of Φ with $|\iota| \in \mathcal{O}(n \cdot |\pi|)$.*

Proof. We only sketch the proof here.

Going through a given mLD-Q-Res (QU-resolution) refutation π, starting at the axioms, for each $C \in \pi$ we create specific natural trails (where some of them will later be part of the $\mathsf{QCDCL}^{\text{Any-Ord}}_{\text{Any-Red,Exi-Prop}}$ or $\mathsf{QCDCL}^{\text{Any-Ord}}_{\text{No-Red,All-Prop}}$ proof) in which all decisions are negated literals from C, until one of the following events occur:
- We get a conflict and learn a subclause of C.
- We obtain a blockade of C.

When this happens, we either assign the label "subclause" or the label "blockade" to C. When a clause was derived via a resolution or reduction step in π, we simply recall the blockades of its parent clauses by applying Lemma 24 to create a blockade for the resolvent or a conflict. If a parent clause does not have a blockade, the clause itself (or a subclause) must have been learned directly and can therefore be used as an antecedent clause for the trail that either becomes a blockade for the resolvent, or that runs into a conflict from which we can learn a subclause of the resolvent.

Since a clause $C \in \pi$ can be derived via resolution (say $C = D \bowtie E$) or reduction (say $C = \mathrm{red}(D)$), we have to consider all possible cases:

- resolution, both D and E are labelled "blockade"
- resolution, D is labelled "blockade", E is labelled "subclause", or vice versa
- resolution, both D and E are labelled "subclause"
- reduction, D is labelled "blockade"
- reduction, D is labelled "subclause"

At the end, each clause in π is either labelled "subclause" or "blockade". In particular, this holds for the empty clause. Because, by definition, there cannot be a blockade of the empty clause (we need at least one literal), the empty clause must be labelled "subclause", which means we have learned the empty clause. ◀

Proposition 18 and Theorem 25 yield the following characterisations:

▶ **Corollary 26.** $\mathsf{QCDCL}^{\textsc{Any-Ord}}_{\textsc{Any-Red},\textsc{Exi-Prop}} \equiv_p$ mLD-Q-Res *and* $\mathsf{QCDCL}^{\textsc{Any-Ord}}_{\textsc{No-Red},\textsc{All-Prop}} \equiv_p$ QU-Res.

▶ Remark 27. Note that our simulations require a particular learning scheme, in which we almost always restart after each conflict. This is also the reason why we get an improved simulation complexity of $\mathcal{O}(n \cdot |\pi|)$ compared to $\mathcal{O}(n^3 \cdot |\pi|)$ from [10], in which arbitrary (asserting) learning schemes were allowed (where we do not necessarily restart every time).

Performing our simulation under arbitrary asserting learning schemes might require some additional analysis on asserting clauses under the ANY-ORD and ANY-RED rules, as a clause learned from a K_1-reductive trail might not be asserting in K_2-reductive trails anymore. However, if it was clear how to guarantee asserting clauses in our systems, we would be able to obtain similar results as in [10], that is:

- For each clause C in the given mLD-Q-Res (QU-resolution) refutation and an arbitrary asserting learning scheme, we need $\mathcal{O}(n^2)$ trails and backtracking steps until we either learn a subclause of C, or we receive a blockade for C.
- Under any arbitrary asserting learning scheme, we can perform the simulation in time $\mathcal{O}(n^3 \cdot |\pi|)$. In particular, we do not need to restart after each conflict.

6 Conclusion

Proving theoretical characterisations of QCDCL variants successfully used in practice is an important and compelling endeavour. While we contributed to this line research, a number of open questions remain, both theoretically and practically. In particular, in light of Figure 1, it seems worthwhile to explore whether some of the QCDCL models shown to be theoretically better than standard QCDCL can be used for practical solving.

In our quest to modify QCDCL to match the strength of its underlying system long-distance Q-resolution, we introduced the new proof system mLD-Q-Res, which not only characterises a strong version of QCDCL, but also simulates all related variants. This allows to use proof-theoretic results for mLD-Q-Res whenever considering the strength of QCDCL solvers. Yet, we leave open whether mLD-Q-Res is strictly weaker than or equivalent to long-distance Q-resolution. Both possible outcomes would be interesting, as either long-distance Q-resolution does not characterise QCDCL, or there are modifications of QCDCL that unleash the full strength of long-distance Q-resolution.

Additionally, we exhibited a QCDCL version characterising QU-resolution. One could try to combine these two characterisations to obtain an even stronger family of QCDCL variants in the spirit of LDQU$^+$-resolution. Further, cube learning, which can hugely impact the running time even on false formulas [17], was not considered here. Hence, verifying true formulas as well as the proof-theoretic characterisation of modifications to QCDCL such as dependency learning [25] are further topics for future research.

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 Albert Atserias and Moritz Müller. Automating resolution is NP-hard. In *IEEE Annual Symposium on Foundations of Computer Science (FOCS)*, pages 498–509. IEEE Computer Society, 2019.

3 Valeriy Balabanov and Jie-Hong R. Jiang. Unified QBF certification and its applications. *Form. Methods Syst. Des.*, 41(1):45–65, 2012.

4 Valeriy Balabanov, Magdalena Widl, and Jie-Hong R. Jiang. QBF resolution systems and their proof complexities. In *Proc. Theory and Applications of Satisfiability Testing (SAT)*, pages 154–169, 2014.

5 Paul Beame, Henry A. Kautz, and Ashish Sabharwal. Towards understanding and harnessing the potential of clause learning. *J. Artif. Intell. Res. (JAIR)*, 22:319–351, 2004. `doi:10.1613/jair.1410`.

6 Olaf Beyersdorff. Proof complexity of quantified Boolean logic – a survey. In Marco Benini, Olaf Beyersdorff, Michael Rathjen, and Peter Schuster, editors, *Mathematics for Computation (M4C)*, pages 353–391. World Scientific, 2022.

7 Olaf Beyersdorff, Joshua Blinkhorn, and Luke Hinde. Size, cost, and capacity: A semantic technique for hard random QBFs. In *Proc. Conference on Innovations in Theoretical Computer Science (ITCS'18)*, pages 9:1–9:18, 2018.

8 Olaf Beyersdorff, Joshua Blinkhorn, and Meena Mahajan. Building strategies into QBF proofs. *J. Autom. Reason.*, 65(1):125–154, 2021. `doi:10.1007/s10817-020-09560-1`.

9 Olaf Beyersdorff, Joshua Blinkhorn, Meena Mahajan, and Tomás Peitl. Hardness characterisations and size-width lower bounds for QBF resolution. *ACM Transactions on Computational Logic*, 2022.

10 Olaf Beyersdorff and Benjamin Böhm. Understanding the Relative Strength of QBF CDCL Solvers and QBF Resolution. In *12th Innovations in Theoretical Computer Science Conference (ITCS 2021)*, volume 185 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 12:1–12:20, 2021.

11 Olaf Beyersdorff, Ilario Bonacina, Leroy Chew, and Jan Pich. Frege systems for quantified Boolean logic. *J. ACM*, 67(2):9:1–9:36, 2020.

12 Olaf Beyersdorff, Leroy Chew, and Mikolás Janota. New resolution-based QBF calculi and their proof complexity. *ACM Transactions on Computation Theory*, 11(4):26:1–26:42, 2019.

13 Olaf Beyersdorff, Leroy Chew, and Mikoláš Janota. Proof complexity of resolution-based QBF calculi. In *Proc. Symposium on Theoretical Aspects of Computer Science (STACS'15)*, pages 76–89. LIPIcs, 2015.

14 Olaf Beyersdorff, Mikolás Janota, Florian Lonsing, and Martina Seidl. Quantified Boolean formulas. In Armin Biere, Marijn Heule, Hans van Maaren, and Toby Walsh, editors, *Handbook of Satisfiability*, Frontiers in Artificial Intelligence and Applications, pages 1177–1221. IOS Press, 2021.

15 Armin Biere, Marijn Heule, Hans van Maaren, and Toby Walsh, editors. *Handbook of Satisfiability*, Frontiers in Artificial Intelligence and Applications. IOS Press, 2021.

16 Benjamin Böhm and Olaf Beyersdorff. Lower bounds for QCDCL via formula gauge. In Chu-Min Li and Felip Manyà, editors, *Theory and Applications of Satisfiability Testing (SAT)*, pages 47–63, Cham, 2021. Springer International Publishing.

17 Benjamin Böhm, Tomás Peitl, and Olaf Beyersdorff. QCDCL with cube learning or pure literal elimination – What is best? In Luc De Raedt, editor, *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence (IJCAI)*, pages 1781–1787. ijcai.org, 2022.

18 Benjamin Böhm, Tomás Peitl, and Olaf Beyersdorff. Should decisions in QCDCL follow prefix order? In Kuldeep S. Meel and Ofer Strichman, editors, *25th International Conference on Theory and Applications of Satisfiability Testing (SAT)*, volume 236 of *LIPIcs*, pages 11:1–11:19. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2022.

19 Sam Buss and Jakob Nordström. Proof complexity and SAT solving. In Armin Biere, Marijn Heule, Hans van Maaren, and Toby Walsh, editors, *Handbook of Satisfiability*, Frontiers in Artificial Intelligence and Applications, pages 233–350. IOS Press, 2021.

20 Mikolás Janota. On Q-Resolution and CDCL QBF solving. In *Proc. International Conference on Theory and Applications of Satisfiability Testing (SAT)*, pages 402–418, 2016.

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

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

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

24 João P. Marques Silva, Inês Lynce, and Sharad Malik. Conflict-driven clause learning SAT solvers. In Armin Biere, Marijn Heule, Hans van Maaren, and Toby Walsh, editors, *Handbook of Satisfiability*, Frontiers in Artificial Intelligence and Applications. IOS Press, 2021.

25 Tomás Peitl, Friedrich Slivovsky, and Stefan Szeider. Dependency learning for QBF. *J. Artif. Intell. Res.*, 65:180–208, 2019.

26 Knot Pipatsrisawat and Adnan Darwiche. On the power of clause-learning SAT solvers as resolution engines. *Artif. Intell.*, 175(2):512–525, 2011. `doi:10.1016/j.artint.2010.10.002`.

27 Friedrich Slivovsky. Quantified CDCL with universal resolution. In Kuldeep S. Meel and Ofer Strichman, editors, *25th International Conference on Theory and Applications of Satisfiability Testing (SAT)*, volume 236 of *LIPIcs*, pages 24:1–24:16. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2022.

28 Allen Van Gelder. Contributions to the theory of practical quantified Boolean formula solving. In *Proc. Principles and Practice of Constraint Programming (CP)*, pages 647–663, 2012.

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

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

31 Lintao Zhang and Sharad Malik. Conflict driven learning in a quantified Boolean satisfiability solver. In *Proc. IEEE/ACM International Conference on Computer-aided Design (ICCAD)*, pages 442–449, 2002.